Delft University of Technology

Consensus Based Distributed Sparse Bayesian Learning By Fast Marginal Likelihood Maximization

Manss, Christoph; Shutin, Dmitriy; Leus, Geert

DOI
10.1109/LSP.2020.3039481

Publication date
2020

Document Version
Final published version

Published in
IEEE Signal Processing Letters

Citation (APA)
Manss, C., Shutin, D., & Leus, G. (2020). Consensus Based Distributed Sparse Bayesian Learning By Fast Marginal Likelihood Maximization. *IEEE Signal Processing Letters*, *27*, 2119-2123. Article 9264682. https://doi.org/10.1109/LSP.2020.3039481

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

https://www.openaccess.nl/en/you-share-we-take-care

Otherwise as indicated in the copyright section: the publisher
is the copyright holder of this work and the author uses the
Dutch legislation to make this work public.

Consensus Based Distributed Sparse Bayesian Learning by Fast Marginal Likelihood Maximization

Christoph Manss, *Member, IEEE*, Dmitriy Shutin, *Senior Member, IEEE*, and Geert Leus, *Fellow, IEEE*

***Abstract*—For swarm systems, distributed processing is of paramount importance and Bayesian methods are preferred for their robustness. Existing distributed sparse Bayesian learning (SBL) methods rely on the automatic relevance determination (ARD), which involves a computationally complex reweighted l1-norm optimization, or they use loopy belief propagation, which is not guaranteed to converge. Hence, this paper looks into the fast marginal likelihood maximization (FMLM) method to develop a faster distributed SBL version. The proposed method has a low communication overhead, and can be distributed by simple consensus methods. The performed simulations indicate a better performance compared with the distributed ARD version, yet the same performance as the FMLM.**

***Index Terms*—Consensus algorithms, distributed optimization, sparse bayesian learning.**

I. Introduction

THIS work focuses on a distributed scheme for cooperative data analysis in swarm systems for exploration. Applications of such cooperative systems can be found in environmental monitoring [1], robotic exploration [2], or disaster relief [3].

To be more specific, consider a network of K robotic agents, each equipped with a sensor, that are able to sense, communicate, and process data. The data collected by the network of agents is assumed to agree with the following general linear model

$$\boldsymbol{y}_k = \boldsymbol{\Phi}_k \boldsymbol{w} + \boldsymbol{\xi}_k, \quad k = 1, \ldots, K. \tag{1}$$

Model (1) assumes that every sensor has access only to its "private" design matrix $\boldsymbol{\Phi}_k \in \mathbb{R}^{M_k \times N}$ and measurement data $\boldsymbol{y}_k \in \mathbb{R}^{M_k}$, which is perturbed by an additive noise $\boldsymbol{\xi}_k \in \mathbb{R}^{M_k}$ that we assume to be a white zero-mean normally distributed random process with precision parameter $\lambda \in \mathbb{R}_+$. However, all agents share a common parameter vector $\boldsymbol{w} \in \mathbb{R}^N$. The estimation of $\boldsymbol{w}$ from individual measurements $\boldsymbol{y}_k$ requires cooperation between agents or in-network estimation strategies. This work is concerned with a specific type of in-network algorithms that results in a sparse estimate of $\boldsymbol{w}$.

Classical approaches to in-network sparse estimation of $\boldsymbol{w}$ are often related to a distributed solution of a so-called (LASSO) problem [4]–[6] that through enforcing a network consensus leads to a common sparse estimate at all agents. Alternatively, the requirement of a global consensus can be relaxed, as in e.g. [7], [8]. Yet these methods are essentially concerned with finding parameter values that optimize a certain cost, such as the mean squared error. The statistics (or confidence) of an estimate of $\boldsymbol{w}$ is less relevant for these methods. Moreover, the computation of the confidence of the estimated parameters can be challenging for LASSO problems due to the non-smoothness of the objective function. There are, however, a class of applications of distributed sparse estimation where the uncertainty about the parameters of interest is important and needs to be available. For instance, if agents are supposed to explore [9], they can utilize this uncertainty for an active information gathering (see e.g., [10], [11]).

The approach proposed in this work cooperatively computes a sparse estimate of the parameter vector $\boldsymbol{w}$ and, at the same time, a parameter covariance matrix. Specifically, our focus lies on Bayesian approaches to distributed estimation of $\boldsymbol{w}$, and particularly SBL methods [12], [13]. SBL is a family of empirical Bayes techniques that find a sparse estimate of $\boldsymbol{w}$ by modeling the weights using a hierarchical gamma-Gaussian prior [14] parametrized by hyper-parameters that model the variance of the weight vector elements. This formulation has been shown to encourage the weight posterior probability mass to concentrate around the axes in the $\boldsymbol{w}$-parameter space, which leads to a sparse weight vector estimate. Then, the estimated hyper-parameters parametrize the estimated covariance matrix. Although the latter feature is not utilized in the current paper, the proposed method provides a basis for active information gathering as in [10], [11], while requiring a network-wide consensus over the hyper-parameters.

Various (centralized) implementations of SBL are proposed in the literature [12], [15], [16]. Decentralized variations of SBL are proposed in e.g. [17], where a distributed estimate is calculated by loopy belief propagation. In [18] another version of a decentralized SBL method is presented, where an ARD version of SBL is implemented. Its key feature is guaranteed convergence. The solution is obtained by optimizing a convex variational bound via a sequence of LASSO optimization problems. The latter is then solved distributively using the popular alternating direction method of multipliers (ADMM) algorithm [5], [6].

The ADMM-based solution to distributed SBL has, however, several shortcomings. First, this approach includes several optimization loops: an outer loop for inferring the hyper-parameters that parametrize the hierarchical weight prior and an inner loop

Manuscript received August 4, 2020; revised October 19, 2020; accepted November 11, 2020. Date of publication November 19, 2020; date of current version December 18, 2020. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ioannis D. Schizas. *(Corresponding author: Christoph Manss.)*

Christoph Manss and Dmitriy Shutin are with the German Aerospace Center, Institute of Communication and Navigation, 82234 Wessling, Germany (e-mail: christoph.manss@dlr.de; dmitriy.shutin@dlr.de).

Geert Leus is with the Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: g.j.t.leus@tudelft.nl).

Digital Object Identifier 10.1109/LSP.2020.3039481

1070-9908 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: TU Delft Library. Downloaded on August 26,2021 at 08:14:15 UTC from IEEE Xplore. Restrictions apply.

that involves an ADMM algorithm, which requires multiple consensus iterations. This slows down the convergence of the distributed SBL algorithm and increases the communication load in the network. Second, the ADMM algorithm requires to specify a regularization parameter for the augmented Lagrangian. This parameter impacts the convergence properties of the whole distributed solution, and has to be individually calibrated beforehand [19].

The current paper proposes a modification of the algorithm in [18], which instead uses an incremental optimization of the SBL objective function. In essence, we propose a distributed version of the FMLM algorithm [15], [20] for SBL. To the best of our knowledge, this is the first distributed implementation of FMLM. In contrast to [18], the proposed method does not require ADMM and is thus free of additional parameters regulating the convergence. Moreover, a network consensus is required only once, before the local computations commence. Consequently, the algorithm converges fast while having a fixed communication load on the network. As we will show, using simulations and real data studies, the proposed algorithm performs on par with the centralized versions in terms of normalized mean squared error (NMSE) and parameter sparsity, and outperforms the distributed SBL variant proposed in [18] for homogeneous learning.

II. Sparse Bayesian Learning and Fast Marginal Likelihood Maximization

In SBL the parameter weights $\boldsymbol{w}$ are modeled with a prior $p(\boldsymbol{w}|\boldsymbol{\gamma}) = \prod_{n=1}^{N} p(w_n|\gamma_n)$, where $p(w_n|\gamma_n) = \mathcal{N}(0, \gamma_n)$, $n = 1, \ldots, N$, and $\boldsymbol{\gamma} \in \mathbb{R}_+^N$ are the hyper-parameters. For a centralized estimator, having access to all $M = \sum_{k=1}^{K} M_k$ measurements, the hyper-parameters $\boldsymbol{\gamma}$ are estimated by maximizing a Type II likelihood function $p(\boldsymbol{y}|\boldsymbol{\gamma})$ [12], [14] computed as follows:

$$p(\boldsymbol{y}|\boldsymbol{\gamma}) = \int_{-\infty}^{\infty} p(\boldsymbol{y}|\boldsymbol{w})p(\boldsymbol{w}|\boldsymbol{\gamma})\mathrm{d}\boldsymbol{w} = \frac{e^{-\frac{1}{2}\boldsymbol{y}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{y}}}{(2\pi)^{M/2}\sqrt{|\boldsymbol{\Sigma}|}}, \quad (2)$$

where $\boldsymbol{\Sigma} = \boldsymbol{\Lambda}^{-1} + \boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T \in \mathbb{R}^{M\times M}$, $\boldsymbol{\Phi} = [\boldsymbol{\Phi}_1^T, \ldots, \boldsymbol{\Phi}_K^T]^T \in \mathbb{R}^{M\times N}$, $\boldsymbol{\Lambda} = \lambda \boldsymbol{I} \in \mathbb{R}^{M\times M}$, and $\boldsymbol{\Gamma} = \operatorname{diag}\{\boldsymbol{\gamma}\}$. By defining $\mathcal{L}(\boldsymbol{\gamma}) = -\log p(\boldsymbol{y}|\boldsymbol{\gamma})$ the optimal $\widehat{\boldsymbol{\gamma}}$ is found as a solution to

$$\widehat{\boldsymbol{\gamma}} = \underset{\boldsymbol{\gamma}\in\mathbb{R}_+}{\arg\min}\, \mathcal{L}(\boldsymbol{\gamma}) = \underset{\boldsymbol{\gamma}\in\mathbb{R}_+}{\arg\min}\, \log|\boldsymbol{\Sigma}| + \boldsymbol{y}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{y}, \quad (3)$$

where we ignored the constant terms. Given $\widehat{\boldsymbol{\gamma}}$, the weights can be estimated from the posterior $p(\boldsymbol{w}|\boldsymbol{y}, \widehat{\boldsymbol{\gamma}}) \propto \mathcal{N}(\widehat{\boldsymbol{w}}, \boldsymbol{\Sigma}_{\boldsymbol{w}})$ where

$$\widehat{\boldsymbol{w}} = \boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{\Phi}^T\boldsymbol{\Lambda}\boldsymbol{y}, \qquad \boldsymbol{\Sigma}_{\boldsymbol{w}} = \left(\boldsymbol{\Phi}^T\boldsymbol{\Lambda}\boldsymbol{\Phi} + \boldsymbol{\Gamma}^{-1}\right)^{-1}. \quad (4)$$

In FMLM, (3) is optimized component-wise. To this end, the contribution of the n-th component γ_n to $\mathcal{L}(\boldsymbol{\gamma})$ is studied as follows (see also [15]). Define $\boldsymbol{\Sigma}_{\overline{n}} = \boldsymbol{\Lambda}^{-1} + \sum_{i\neq n}\gamma_i\boldsymbol{\phi}_i\boldsymbol{\phi}_i^T \in \mathbb{R}^{M\times M}$ with the n-th column $\boldsymbol{\phi}_n \in \mathbb{R}^M$ of $\boldsymbol{\Phi}$ removed. Then, $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_{\overline{n}} + \gamma_n\boldsymbol{\phi}_n\boldsymbol{\phi}_n^T$ and its inverse can be computed using the matrix inversion lemma [21] as

$$\boldsymbol{\Sigma}^{-1} = \boldsymbol{\Sigma}_{\overline{n}}^{-1} - \frac{\boldsymbol{\Sigma}_{\overline{n}}^{-1}\boldsymbol{\phi}_n\boldsymbol{\phi}_n^T\boldsymbol{\Sigma}_{\overline{n}}^{-1}}{\gamma_n^{-1} + \boldsymbol{\phi}_n^T\boldsymbol{\Sigma}_{\overline{n}}^{-1}\boldsymbol{\phi}_n}. \quad (5)$$

Algorithm 1: FMLM.

```
Initialize $\boldsymbol{\gamma}$:
$\forall n, \gamma_n \leftarrow (\|\boldsymbol{\phi}_n^T\boldsymbol{y}\|/\|\boldsymbol{\phi}_n\|^2 - \lambda^{-1})/\|\boldsymbol{\phi}_n\|^2$
while not converged do
    for $n \in \{1, \ldots, N\}$ do
        $S_n \leftarrow (9)$, $Q_n \leftarrow (10)$, $s_n, q_n \leftarrow (8)$
        if $q_n^2 > s_n$ and $\gamma_n \neq 0$ then                ▷Update
            $\gamma_n = \frac{q_n^2 - s_n}{s_n^2}$
        else if $q_n^2 > s_n$ and $\gamma_n = 0$ then              ▷Add
            $\gamma_n = \frac{q_n^2 - s_n}{s_n^2}$, and add $\phi_n$ to the model
        else if $q_n^2 \leq s_n$ and $\gamma_n \neq 0$ then         ▷Remove
            $\gamma_n = 0$, and remove $n$-th basis function
    $\boldsymbol{\Sigma}_{\boldsymbol{w}} \leftarrow (\boldsymbol{\Phi}^T\boldsymbol{\Lambda}\boldsymbol{\Phi} + \boldsymbol{\Gamma}^{-1})^{-1}$
return $\widehat{\boldsymbol{w}} \leftarrow \boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{\Phi}^T\boldsymbol{\Lambda}\boldsymbol{y}$
```

Inserting (5) into $\mathcal{L}(\boldsymbol{\gamma})$ and simplifying the result leads to

$$\mathcal{L}(\boldsymbol{\gamma}_{\overline{n}}, \gamma_n) = -\log|\boldsymbol{\Sigma}_{\overline{n}}| - \boldsymbol{y}^T\boldsymbol{\Sigma}_{\overline{n}}^{-1}\boldsymbol{y} + l(\gamma_n), \quad (6)$$

where $\boldsymbol{\gamma}_{\overline{n}} \in \mathbb{R}_+^{N-1}$ is a vector of hyper-parameters $\boldsymbol{\gamma}$ without the n-th component, and

$$l(\gamma_n) = -\log\left\{\gamma_n(\gamma_n^{-1} + s_n)\right\} + \frac{q_n^2}{\gamma_n^{-1} + s_n}. \quad (7)$$

The parameters s_n and q_n in (7) are defined as [15]

$$s_n = \frac{\gamma_n^{-1}S_n}{\gamma_n^{-1} - S_n}, \qquad q_n = \frac{\gamma_n^{-1}Q_n}{\gamma_n^{-1} - S_n}, \quad (8)$$

where

$$S_n = \boldsymbol{\phi}_n^T\boldsymbol{\Lambda}\boldsymbol{\phi}_n - \boldsymbol{\phi}_n^T\boldsymbol{\Lambda}\boldsymbol{\Phi}\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{\Phi}^T\boldsymbol{\Lambda}\boldsymbol{\phi}_n, \quad (9)$$

$$Q_n = \boldsymbol{\phi}_n^T\boldsymbol{\Lambda}\boldsymbol{y} - \boldsymbol{\phi}_n^T\boldsymbol{\Lambda}\boldsymbol{\Phi}\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{\Phi}^T\boldsymbol{\Lambda}\boldsymbol{y}. \quad (10)$$

The solution to (3) with respect to γ_n can then be found in a closed form as

$$\widehat{\gamma}_n = \begin{cases} \frac{q_n^2 - s_n}{s_n^2}, & \text{if } q_n^2 > s_n, \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

The FMLM algorithm computes (11) for all N components iteratively to solve (3) using a coordinate-wise descent. Moreover, it can be shown [22] that such coordinate-wise optimization of (3) is equivalent to a coordinate-wise minimization of a convex upper bound on $\mathcal{L}(\boldsymbol{\gamma})$, which ensures convergence to the minimizer.

Furthermore, if $\widehat{\gamma}_n = 0$ as shown in (11), the corresponding n-th element $\widehat{w}_n$ in (4) will become zero as well, and, thus, yielding a sparse estimate. It is also worth noting that the FMLM algorithm permits adding or removing basis functions by evaluating (11) (see also [15], [23]). Algorithm 1 summarizes the key FMLM algorithmic steps.

III. Distributed Fast Marginal Likelihood Maximization

The original FMLM algorithm is centralized. To implement it in a distributed fashion, let us first assume that K agents

Authorized licensed use limited to: TU Delft Library. Downloaded on August 26,2021 at 08:14:15 UTC from IEEE Xplore. Restrictions apply.

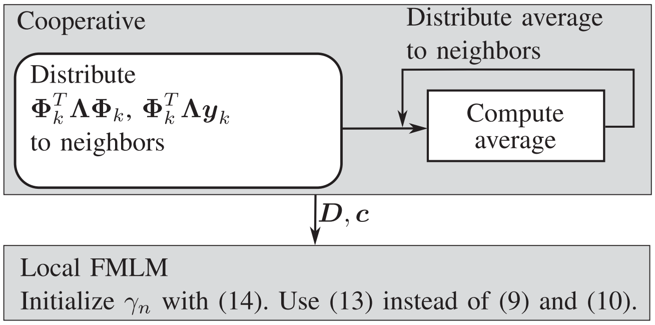
Cooperative
Distribute average to neighbors
Distribute $\boldsymbol{\Phi}_k^T\boldsymbol{\Lambda}\boldsymbol{\Phi}_k$, $\boldsymbol{\Phi}_k^T\boldsymbol{\Lambda}\boldsymbol{y}_k$ to neighbors
Compute average
$\boldsymbol{D}, \boldsymbol{c}$
Local FMLM
Initialize γ_n with (14). Use (13) instead of (9) and (10).

Fig. 1. Flowchart of the proposed algorithm – DFMLM.

form a strongly connected network [24, Sec. 6]. Inspecting now Algorithm 1, we note that the FMLM computations are centered around the quantities S_n and Q_n in (9) and (10), respectively. These can be computed using averaged consensus [25]. It is known that for strongly connected networks average consensus converges to an averaged value of the quantity of interest after a certain number of consensus iterations I_{cons} that depend on the network topology. Thus, we define

$$\boldsymbol{D} \triangleq \boldsymbol{\Phi}^T\boldsymbol{\Lambda}\boldsymbol{\Phi} = \sum_{k=1}^{K}\boldsymbol{\Phi}_k^T\boldsymbol{\Lambda}\boldsymbol{\Phi}_k,\ \boldsymbol{c} \triangleq \boldsymbol{\Phi}^T\boldsymbol{\Lambda}\boldsymbol{y} = \sum_{k=1}^{K}\boldsymbol{\Phi}_k^T\boldsymbol{\Lambda}\boldsymbol{y}_k, \tag{12}$$

which can be computed using average consensus [25], if K is known to all agents. Now, we define a selection vector $\boldsymbol{e}_n \in \mathbb{R}^N$, which is zero everywhere except at the n-th position where it equals 1. Once the quantities in (12) have converged, each agent can evaluate (9) and (10) locally as

$$S_n = \boldsymbol{e}_n^T\left(\boldsymbol{D} - \boldsymbol{D}\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{D}\right)\boldsymbol{e}_n,\ Q_n = \boldsymbol{e}_n^T\left(\boldsymbol{c} - \boldsymbol{D}\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{c}\right), \tag{13}$$

with $\boldsymbol{\Sigma}_{\boldsymbol{w}} = (\boldsymbol{D} + \boldsymbol{\Gamma}^{-1})^{-1}$. Thus, using (13), S_n and Q_n become known to the network, and (8) can be used to estimate $\widehat{\gamma}_n$, $\forall n = 1, \ldots, N$ as in (11). Moreover, each agent has then an estimate of the weight covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{w}}$. By assuming a network wide consensus over $\boldsymbol{D}$ and $\boldsymbol{c}$, we thus ensure that both $\widehat{\boldsymbol{\gamma}}$ as well as $\boldsymbol{\Sigma}_{\boldsymbol{w}}$ are the same for all agents. As such, no more communication is required. The flowchart in Fig. 1 summarizes the distributed FMLM (DFMLM). DFMLM is an iterative optimization technique that requires appropriate initialization. In a distributed setting, we utilize the initialization proposed in [15] using $\boldsymbol{D}$ and $\boldsymbol{c}$ computed with average consensus. Specifically, we compute

$$\begin{aligned}\gamma_n^{[\text{init}]} &= \frac{\|\boldsymbol{\phi}_n^T\boldsymbol{y}\|/\|\boldsymbol{\phi}_n\|^2 - \lambda^{-1}}{\|\boldsymbol{\phi}_n\|^2} = \lambda\frac{(\lambda\boldsymbol{\phi}_n^T\boldsymbol{y})/(\lambda\boldsymbol{\phi}_n^T\boldsymbol{\phi}_n) - \lambda^{-1}}{\lambda\boldsymbol{\phi}_n^T\boldsymbol{\phi}_n}\\ &= \begin{cases}\lambda\frac{[\boldsymbol{c}]_n/[\boldsymbol{D}]_{n,n} - \lambda^{-1}}{[\boldsymbol{D}]_{n,n}}, & \text{if } [\boldsymbol{c}]_n/[\boldsymbol{D}]_{n,n} > \lambda^{-1},\\ 0, & \text{otherwise.}\end{cases}\end{aligned} \tag{14}$$

where $[\cdot]_k$ and $[\cdot]_{k,l}$ denote the k-th element of a vector or a (k,l)-th element of a matrix, respectively. Each agent evaluates (14) for all N components. Then, the component with the highest value of $\gamma_n^{[\text{init}]}$ is set as an initial value for γ_n. The remaining hyper-parameters $\boldsymbol{\gamma}_{\overline{n}}$ are set to zero. The corresponding component can then be added to the model during the FMLM iterations (see Algorithm 1).

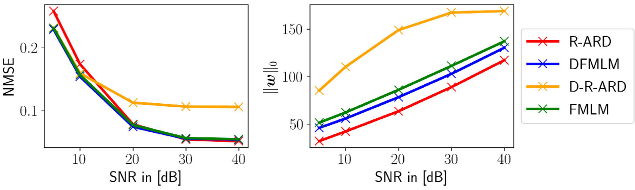

Fig. 2. (left) NMSE of all algorithms for different signal-to-noise ratio (SNR). (right) the number of nonzero coefficients of all algorithms for different SNR.

IV. Simulations

Here, we test the algorithm by using a two-dimensional scalar field $f(\boldsymbol{x})$, with $\boldsymbol{x} \in \mathbb{R}^2$. The design matrices $\boldsymbol{\Phi}_k, k = 1, \ldots, K$, are constructed using N Gaussian basis functions of the form $\phi_n(\boldsymbol{x}) = \exp(-(\boldsymbol{x} - \boldsymbol{\mu}_n)/(2\sigma^2))$ centered at predefined locations $\boldsymbol{\mu}_n \in \mathbb{R}^2$ and having a width of $\sigma^2 > 0$. By constructing a vector $\boldsymbol{\phi}(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}), \ldots, \phi_N(\boldsymbol{x})]^T \in \mathbb{R}^N$, we form $\boldsymbol{\Phi}_k = [\boldsymbol{\phi}(\boldsymbol{x}_1), \ldots, \boldsymbol{\phi}(\boldsymbol{x}_{M_k})]^T$, $k = 1, \ldots, K$. We assume that all $\boldsymbol{\mu}_n$ are regularly sampled on a grid, and that the measurement positions of each agent k are sampled uniformly at random. We will also assume that a network with K agents is strongly connected, and a broadcast communication protocol is used during consensus; we also assume no data loss during the transmission.

As benchmark algorithms we use the centralized FMLM algorithm [15] and the centralized reformulated automatic relevance detection (R-ARD) algorithm [16]. Also, we will compare the performance of the proposed algorithm to a distributed version of R-ARD – the distributed R-ARD (D-R-ARD) [18] applied to the model (1). As mentioned earlier, the D-R-ARD algorithm uses an ADMM algorithm to estimate the hyper-parameters via a sequence of several LASSO problems, which are solved over the network. The benchmark version, presented here, uses the distributed LASSO (DLASSO) algorithm from [6] for this purpose. The required regularization parameter $\rho \in \mathbb{R}_+$ for the augmented Lagrangian in the DLASSO algorithm is set to $\rho = 2.5$. The algorithms are evaluated by means of the NMSE defined as $\epsilon = \frac{\|\boldsymbol{f} - \boldsymbol{\Phi}_V\widehat{\boldsymbol{w}}\|}{\|\boldsymbol{f}\|}$, where $\boldsymbol{f} = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_V)]^T \in \mathbb{R}^V$ with $V \in \mathbb{N}$ validation points and $\boldsymbol{\Phi}_V = [\boldsymbol{\phi}(\boldsymbol{x}_1), \ldots, \boldsymbol{\phi}(\boldsymbol{x}_V)]^T \in \mathbb{R}^{V\times N}$. The validation points are drawn uniformly at random from the support of $f(\boldsymbol{x})$. All simulations are averaged over 100 Monte Carlo runs.

A. Artificial Data Simulations

Similar to [12], we used $f(\boldsymbol{x}) = \text{sinc}(x_0) + 0.1x_1$ for analyzing the algorithms presented in the current paper. The free parameters were set as $V = 225$, $N = 225$, $M_k = 9$, $K = 10$, and $\sigma = 1.5$. The left of Fig. 2 shows the performance of the algorithms as a function of SNR. All algorithms perform almost equally with except D-R-ARD, which has a higher NMSE for a higher SNR. This behavior can be explained by the impact of the fixed parameter ρ in ADMM on changing SNR values, as it is here fit for a lower SNR. Similar behavior can be seen on the

Authorized licensed use limited to: TU Delft Library. Downloaded on August 26,2021 at 08:14:15 UTC from IEEE Xplore. Restrictions apply.

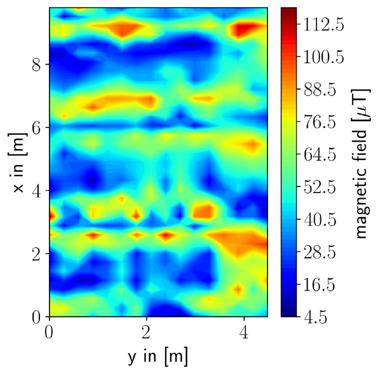

Fig. 3. The magnetic field strength in μT in our laboratory.

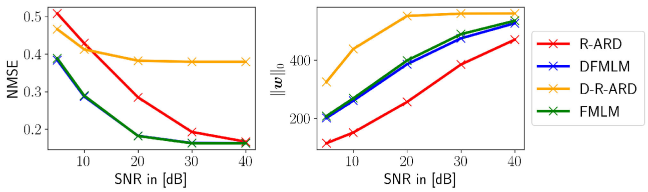

Fig. 4. (left) NMSE vs. SNR of all algorithms. (right) number all algorithm's nonzero coefficients for different SNR.

right of Fig. 2, where the D-R-ARD yields the lowest sparsity. FMLM and the proposed DFMLM have the same performance, as expected.

B. Real Data Simulations

Fig. 3 shows the magnetic field strength in our laboratory, which is used to test our algorithm on real data. Here we set $V = 560$, $N = 560$, $M_k = 24$, $K = 10$, and $\sigma = 0.25$. The performance with respect to SNR is shown in Fig. 4 on the left. Again, the performance of D-R-ARD is worse compared to the other algorithms.

The sparsity of the estimated signals is shown in the right panel of Fig. 4. D-R-ARD again performs poorly due to a fixed choice of the ADMM regularization parameter. Yet, this parameter is chosen to achieve a lower NMSE instead of a higher sparsity for low SNR. Additionally, we show in Fig. 5 the estimates of all algorithms for a particular run at 10 dB SNR, where a black cross represents the mean of a basis function with non-zero weight. Comparing FMLM and DFMLM, we see only a slight difference in the location of estimated components; yet in general both algorithms perform equivalently. In the case of D-R-ARD, the ADMM parameter again has a strong influence on the sparsity of the estimated representation.

C. Communication Load for DFMLM

As can be observed in Fig. 1, the DFMLM algorithm has only a single consensus step at the beginning for the computation of the covariance $\sum_{k=1}^{K} \boldsymbol{\Phi}_k^T \boldsymbol{\Lambda} \boldsymbol{\Phi}_k$ and the cross-correlation $\sum_{k=1}^{K} \boldsymbol{\Phi}_k^T \boldsymbol{\Lambda} \boldsymbol{y}_k$, which are used to compute $\boldsymbol{D}$ and $\boldsymbol{c}$ in (12). Exploiting the symmetry of $\boldsymbol{D}$, the consensus requires to exchange $N(N+3)/2$ values per consensus iteration; thus, in total $I_{\text{cons}} N(N+3)/2$ values are exchanged. The D-R-ARD [18] that uses the ADMM algorithm to estimate the hyper-parameters also requires the computation of the matrix $\boldsymbol{D}$ at the initialization stage. The algorithm itself includes I_{ARD} iterations, each of which requires $N \times I_{\text{cons}} I_{\text{ADMM}}$ exchanges to solve the LASSO problem distributively (eq. (11) in [16]), where I_{cons} and I_{ADMM} are the number of consensus and ADMM iterations, respectively. A simple analysis shows that for, e.g., a fully connected network (i.e., when $I_{\text{cons}} = 1$) the D-R-ARD requires $(I_{\text{ARD}} I_{\text{ADMM}} - 1)N$ more values to be exchanged compared with the DFMLM algorithm.

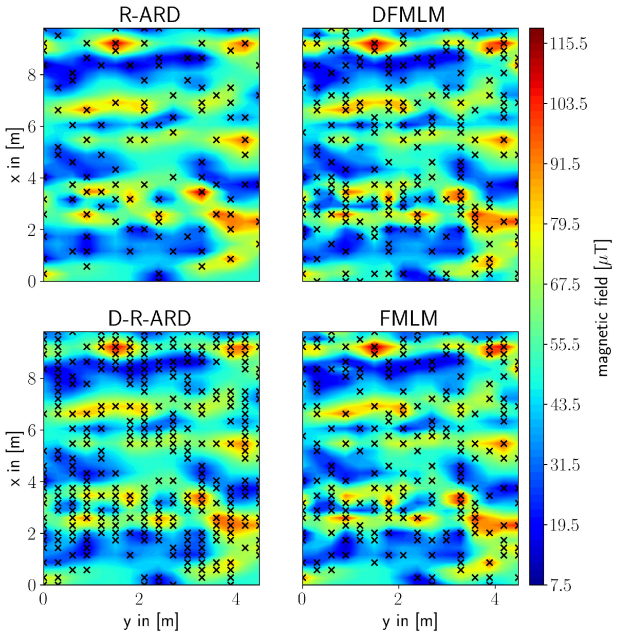

Fig. 5. Estimated magnetic field strength. A black cross represents the mean of a Gaussian basis function with non-zero weight.

Note that for SBL the amount of exchanged data scales as $O(N^2)$ since covariance information is used to estimate the sparsity pattern. In cases when only a sparse estimate of w without its covariance is of interest, the communication load typically scales with $O(N)$. Comparisons in this case are more involved and vary depending on the particular implementation of the algorithm: cases can be found when slow convergence leads to multiple $O(N)$ exchanges that will eventually dominate over the $I_{\text{cons}} N(N+3)/2$ exchanges for the DFMLM.

V. Conclusion

In this paper, we presented a distributed version of the FMLM – the DFMLM algorithm – that estimates a sparse parameter vector along with its covariance information in a distributed fashion. The proposed algorithm has only a single consensus step with fixed communication overhead. Simulations with artificial and real data indicate that DFMLM is superior in terms of sparsity and accuracy compared to another distributed implementation of the SBL algorithm namely D-R-ARD which depends on an additional free parameter that impacts the convergence.

Authorized licensed use limited to: TU Delft Library. Downloaded on August 26,2021 at 08:14:15 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] A. Viseras, T. Wiedemann, C. Manss, V. Karolj, D. Shutin, and J. Marchal, “Beehive-inspired information gathering with a swarm of autonomous drones,” *Sensors*, vol. 19, no. 19, pp. 43–49, Jan. 2019.

[2] R. Ouyang, K. H. Low, J. Chen, and P. Jaillet, “Multi-robot active sensing of non-stationary Gaussian process-based environmental phenomena,” in *Proc. Int. Conf. Auton. Agents Multi-Agent Syst.*, May 2014, pp. 573–580.

[3] M. Frassl, M. Lichtenstern, M. Angermann, and G. Gullotta, “Micro aerial vehicles in disaster assessment operations – The example of Cyprus 2011,” in *Future Security*, N.Aschenbruck, P. Martini, M. Meier, and J. Tölle, Eds. Berlin, Germany: Springer, 2012, pp. 475–479.

[4] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *J. Roy. Stat. Soc. Ser. B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[5] S. Boyd, N. Parikh, E. Chu, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.

[6] G. Mateos, J. A. Bazerque, and G. B. Giannakis, “Distributed sparse linear regression,” *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, Oct. 2010.

[7] N. Tran, H. Ambos, and A. Jung, “Classifying partially labeled networked data VIA logistic network lasso,” in *Process. ICASSP. Acoust., Speech Signal Proc.*, May 2020, pp. 3832–3836.

[8] A. Jung and N. Tran, “Localized linear regression in networked data,” *IEEE Signal Process. Lett.*, vol. 26, no. 7, pp. 1090–1094, Jul. 2019.

[9] C. Manss, D. Shutin, and G. Leus, “Coordination methods for entropy-based multi-agent exploration under sparsity constraints,” in *Proc. IEEE 8th Int. Workshop Comput. Adv. Multi-Sensor Adaptive Process.*, Le Gosier, Dec. 2019, pp. 490–494.

[10] D. J. C. MacKay, “Information-based objective functions for active data selection,” *Neural Comput.*, vol. 4, no. 4, pp. 590–604, Jul. 1992.

[11] P. Whaite and F. P. Ferrie, “Autonomous exploration: Driven by uncertainty,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 3, pp. 193–205, Mar. 1997.

[12] M. E. Tipping, “Sparse Bayesian learning and the relevance vector machine,” *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, 2001.

[13] D. P. Wipf and B. D. Rao, “Sparse Bayesian learning for basis selection,” *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2153–2164, Aug. 2004.

[14] R. Giri and B. Rao, “Type I and Type II Bayesian methods for sparse signal recovery using scale mixtures,” *IEEE Trans. Signal Process.*, vol. 64, no. 13, pp. 3418–3428, Jul. 2016.

[15] M. E. Tipping and A. C. Faul, “Fast marginal likelihood maximisation for sparse Bayesian models,” in *Proc. Ninth Int. Workshop Artif. Intell. Statist.*, 2003. [Online]. Available: https://web.archive.org/web/20071222191445/http://research.microsoft.com/conferences/AIStats2003/proceedings/papers.htm

[16] D. P. Wipf and S. S. Nagarajan, “A new view of automatic relevance determination,” in *Advances in Neural Information Processing Systems 20*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds. Curran Associates, Vancouver, BC, Canada, 2008, pp. 1625–1632.

[17] T. Buchgraber, D. Shutin, and H. V. Poor, “A sliding-window online fast variational sparse Bayesian learning algorithm,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2011, pp. 2128–2131.

[18] C. Manss, D. Shutin, and G. Leus, “Distributed splitting-over-features sparse Bayesian learning with alternating direction method of multipliers,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Apr. 2018, pp. 3654–3658.

[19] P. Giselsson and S. Boyd, “Linear convergence and metric selection for Douglas-Rachford splitting and ADMM,” *IEEE Trans. Autom. Control*, vol. 62, no. 2, pp. 532–544, Feb. 2017.

[20] D. Shutin, S. R. Kulkarni, and H. V. Poor, “ Stationary point variational Bayesian attribute-distributed sparse learning with l1 sparsity constraints,” in *Proc. 4th IEEE Int. Workshop. Multi-Sensor Adaptive Proc. (CAMSAP)*, Dec. 2011, pp. 277–280.

[21] D. J. Tylavsky and G. R. Sohie, “Generalization of the matrix inversion lemma,” *Proc. IEEE*, vol. 74, no. 7, pp. 1050–1052, Jul. 1986.

[22] D. Shutin, S. R. Kulkarni, and H. V. Poor, “Incremental reformulated automatic relevance determination,” *IEEE Trans. Signal Process.*, vol. 60, no. 9, pp. 4977–4981, Sep. 2012.

[23] D. Shutin, T. Buchgraber, S. R. Kulkarni, and H. V. Poor, “Fast variational sparse Bayesian learning with automatic relevance determination for superimposed signals,” *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6257–6261, Dec. 2011.

[24] A. H. Sayed, “Adaptation, learning, and optimization over networks,” *Found. Trends Mach. Learn.*, vol. 7, nos. 4/5, pp. 311–801, 2014.

[25] A. Nedic, A. Ozdaglar, and P. A. Parrilo, “Constrained consensus and optimization in multi-agent networks,” *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 922–938, Apr. 2010.

Authorized licensed use limited to: TU Delft Library. Downloaded on August 26,2021 at 08:14:15 UTC from IEEE Xplore. Restrictions apply.