This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSIPN.2016.2632041, IEEE Transactions on Signal and Information Processing over Networks

Communication Efficient Decentralized Sparse Bayesian Learning of Joint Sparse Signals

Saurabh Khanna, *Student Member, IEEE,* and Chandra R. Murthy, *Senior Member, IEEE*

***Abstract*—We consider the problem of decentralized estimation of multiple joint sparse vectors by a network of nodes from locally acquired noisy and underdetermined linear measurements, when the cost of communication between the nodes is at a premium. We propose an iterative, decentralized Bayesian algorithm called FB-DSBL (Fusion Based Distributed Sparse Bayesian Learning) in which the nodes collaborate by exchanging highly compressed messages to learn a common joint sparsity inducing signal prior. The learnt signal prior is subsequently used by each node to compute the maximum a posteriori probability (MAP) estimate of its respective sparse vector. Since the inter-node communication cost is expensive, the size of the messages exchanged between nodes is reduced substantially by exchanging only those local signal prior parameters which are associated with the nonzero support detected via multiple composite log-likelihood ratio tests. The average message size is empirically shown to be proportional to the information rate of the unknown vectors. The proposed Sparse Bayesian Learning (SBL) based distributed algorithm allows nodes to exploit the underlying joint sparsity of the signals. In turn, this enables the nodes to recover sparse vectors with significantly lower number of measurements compared to the standalone SBL algorithm. The proposed algorithm is interpreted as a degenerate case of a distributed consensus based stochastic approximation algorithm for finding a fixed point of a function, and its generalized version with Robbins-Monro type iterations is also developed. Using Monte Carlo simulations, we demonstrate that the proposed FB-DSBL has superior MSE and support recovery performance compared to the existing decentralized algorithms with similar or higher communication complexity.**

***Index Terms*—Compressed sensing, distributed estimation, joint sparsity, sparse Bayesian learning, sensor networks.**

I. Introduction

A vector $\mathbf{x} \in \mathbb{R}^n$ is said to be *s-sparse* if at most $s(\ll n)$ of its coefficients are nonzero. Further, two or more vectors are said to be *jointly sparse* if they share a common nonzero support, i.e., their nonzero coefficients belong to the same index set. In this work, we consider the problem of in-network estimation of multiple joint sparse vectors by a network of nodes, where each node is interested in estimating only its own local sparse vector from its noisy, underdetermined, linear measurements. Since the local sparse vectors at the individual nodes share a common support, they can be jointly estimated from significantly fewer measurements compared to their independent reconstruction by their respective nodes. This motivates us to explore efficient mechanisms for collaboration between the network nodes which can exploit the network-wide joint sparsity of the local sparse vectors. A typical example of such a distributed setup involves multiple connected agents forming a large network, trying to learn a local sparse parameterized model of an unknown but common physical phenomenon. Since the same underlying phenomenon is observed by each agent, their respective sparse model parameters/representations tend to exhibit joint sparsity. The joint sparse vectors considered in this work belong to the Type-2 Joint Sparsity Model (JSM-2) [1], in which the nonzero coefficients are assumed to be uncorrelated within and across the vectors. The JSM-2 signal model has been successfully applied to several interesting problems such as MIMO channel estimation [2]–[4], cooperative spectrum sensing [5], [6], decentralized event detection [7] and acoustic source localization [8].

In the literature, distributed estimation of JSM-2 signals has been addressed in two ways - centralized and decentralized. In a centralized approach, each node communicates its local measurements to a fusion center which then recovers the unknown joint sparse vectors and transmits the reconstructed signals back to the respective nodes [1], [9]–[15]. In contrast, in this paper, we consider decentralized recovery of joint sparse vectors. This approach is not only inherently robust to node failures, but is also energy efficient when implemented over large networks. We are interested in decentralized schemes that yield the same solution as the centralized algorithm, while processing the observations locally at each node and exchanging messages between one-hop neighbors in the network.

Decentralized algorithms, however, might incur a high communication cost due to the exchange of messages. Most existing decentralized algorithms in literature catering to joint sparse signal recovery involve multiple exchanges of $\mathcal{O}(n)$ sized messages between the network nodes, where n is the dimension of the sparse vectors. Since n is typically large, the $\mathcal{O}(n)$ scaling of message size poses a severe bottleneck in terms of the communication overhead for in-network signal recovery. In practice, each network node needs to operate within a limited budget for time and power resources dedicated towards inter-node communications. Thus, any sizable reduction in the amount of information exchanged between the nodes is highly desirable as it has a direct positive impact on the operating life time of the network. Herein lies the challenge of devising schemes for reconstruction of sparse signals locally at each of the nodes, while exploiting their joint sparsity to improve the reconstruction accuracy, by

The authors are with the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore 560012, India (e-mail: sakhanna@ece.iisc.ernet.in; cmurthy@ece.iisc.ernet.in).

This work has been financially supported in part by a research grant from the Ministry of Electronics and Information Technology (MeitY), Govt. of India.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSIPN.2016.2632041, IEEE Transactions on Signal and Information Processing over Networks

using minimal communication between them. Further, in many security related applications, it is also important to ensure that the messages exchanged between the nodes cannot be misused by a malicious node or an eavesdropper to reconstruct the signal vector of another node in the network. Thus, it is preferable to consider decentralized schemes which do not directly exchange signal coefficients or local measurements between the network nodes.

A. *Prior Work*

We start with a review of the decentralized algorithms that have been proposed in the literature for joint sparse signal recovery. DCOMP [16] and DCSP [17] are decentralized extensions of the greedy algorithms SOMP [1] and SSP [17] respectively. Both these algorithms have very low computational complexity but perform poorly compared to convex relaxation and Bayesian recovery methods. Moreover, both DCOMP and DCSP require knowledge of s, the sparsity level, which is generally not available in practice. In [18], three distributed greedy pursuit algorithms: DiOMP, DiSP and DiFROGS have been proposed, and an extensive simulation based performance comparison study has been carried out for a wide range of network connection densities. Among these three algorithms, DiOMP turns out to be the best all-round performer. However, DCOMP is more accurate than DiOMP in terms of support detection, as demonstrated in [16]. DR-LASSO [19] is an iterative decentralized algorithm which uses alternating minimization to optimize a convex regularized objective with an ℓ_1-ℓ_2 mixed norm based joint sparsity inducing penalty. In [6], the same ℓ_1-ℓ_2 norm penalty is used in a cooperative spectrum sensing setup to estimate the joint sparse spectral occupancy patterns perceived by multiple cognitive radios via independent channels. The decentralized re-weighted ℓ_1 norm minimization algorithm or DRL-1 [7] employs a sum-log-sum penalty which is better at promoting sparsity than the penalty function used in DR-LASSO. In DRL-1, the non convex objective is minimized by replacing it with a surrogate convex function made up of weighted ℓ_1 norm terms, and the weights are updated in each iteration. In both DR-LASSO and DRL-1, the sparsity of the solution is controlled by a regularization parameter λ which biases the joint sparsity inducing penalty term in the objective. The correct amount of regularization or the optimal value of λ is typically chosen via cross-validation, which is not practical unless additional training data is available.

In a Bayesian approach, the amount of regularization is tuned automatically by the procedure of selecting an appropriate member prior from a parameterized family of joint sparsity inducing priors. The selected prior is the one with maximal Bayesian evidence. Subsequently, the learned signal prior is used to obtain a maximum a posteriori probability (MAP) estimate of the sparse vectors. DCS-AMP [20] is one such Bayesian algorithm which uses approximate message passing to efficiently learn the parameters of a joint sparsity inducing Bernoulli-Gaussian family of priors. DCS-FBMP [21] also uses a Bernoulli-Gaussian signal prior, and constructs the sparse support incrementally by using a greedy approach to maximize the log-posterior probability of the support parameters. However, this algorithm is designed to work only with star and ring topology networks. The recently proposed CB-DSBL algorithm [22] employs a Gaussian signal prior whose parameters are learned via Type-2 maximum likelihood (ML) estimation. The ML cost is maximized by solving a series of convex optimization problems in a decentralized manner using the alternating directions method of multipliers (ADMM). To reduce the amount of inter-node communication and the associated overheads, a special internode communication scheme is employed, in which the nodes exchange messages with only a small set of bridge nodes. In [23], a similar ADMM based decentralized algorithm is proposed for recovery of JSM-1 signals.

Each of these algorithms involve exchange of $\mathcal{O}(n)$ sized messages per iteration between the nodes, with the exception of DCSP with $\mathcal{O}(s \log n)$ message size, where s is the size of the nonzero support. In this work, we aim to fill this gap in the existing literature by proposing a decentralized algorithm which is well endowed in terms of both signal recovery performance as well as low communication complexity.

B. *Our Contributions*

Our main contributions in this paper are as follows:

1) We propose a novel, decentralized, iterative algorithm for in-network estimation of joint sparse vectors. We call our algorithm *Fusion based Decentralized Sparse Bayesian Learning* (FB-DSBL) to emphasize that the algorithm fuses the support estimates across the nodes, is decentralized, and is based on the sparse Bayesian learning algorithm. In the proposed algorithm, each node computes the MAP estimate of its local sparse vector, using a network-wide joint parameterized prior. The parameterized prior is itself learned on-the-fly using highly compressed messages exchanged over the network. The combined effect of exchanging compressed messages and using a joint parameterized prior is accurate reconstruction of the sparse vectors at the individual nodes using far fewer measurements compared to the independent reconstruction of the sparse vectors at each node.
2) In order to reduce the communication complexity, we propose a scheme to reduce the size of the messages exchanged between the nodes. Each node shares with its single-hop neighbors, only those components of the joint parameterized prior which are associated with the active locations in the local instantaneous estimate of the true support. These active locations are identified via multiple log-likelihood ratio tests. We also propose a scheme for refinement of the local estimates of the joint prior by using the compressed messages received from the single-hop neighbors. We show empirically that FB-DSBL requires the exchange of only $\mathcal{O}(s \log n)$ sized messages between the nodes, where s is the size of the nonzero support.
3) We show that FB-DSBL can be analyzed under the stochastic approximation framework by interpreting its iterations as degenerate distributed Robbins-Monro iterations for finding a fixed point of certain function.

> With this interpretation, we also propose FB-DSBL†, a stochastic approximation inspired generalization of the FB-DSBL algorithm. Through simulations, we show that its performance is marginally better, but its convergence is slower and the communication cost is higher, compared to the FB-DSBL algorithm.

Unlike DCSP and DCOMP, FB-DSBL does not require s, the true sparsity level, to be known apriori, nor does it require the exchange of the raw measurements or sparse vector estimates between the nodes. Thus, it is suitable for applications where data privacy needs to be preserved.

The rest of the paper is organized as follows. In Section II, we formalize the problem considered in this work and briefly present the required background on Sparse Bayesian Learning (SBL) [17]. In Section III, we present the proposed FB-DSBL algorithm for in-network decentralized recovery of multiple joint sparse signals. An interesting interpretation of FB-DSBL algorithm as a special case of the Robbins-Monro type distributed stochastic approximation algorithm is discussed in Section IV. Based on this interpretation, we also suggest a pseudo-stochastic variant of the proposed algorithm. A detailed comparison of the performance of the proposed algorithms relative to existing solutions is presented in Section V. Concluding remarks are offered in Section VI.

Notation: Throughout this paper, lowercase (x) and bold lowercase ($\mathbf{x}$) alphabets are used to represent scalars and vectors, respectively. Bold uppercase alphabets ($\mathbf{X}$) are used to represent matrices. The i^{th} entry of vector $\mathbf{x}$ at node j is denoted by $\mathbf{x}_j(i)$. The vector $supp(\mathbf{x})$ represents the binary nonzero support of $\mathbf{x}$. The operator $\circ$ denotes the element-wise multiplication operation. The vector comprising elementwise squared entries of the vector $\mathbf{x}$ is denoted by $\mathbf{x}^2$. The i^{th} column of the matrix $\mathbf{\Phi}_j$ is denoted by $\mathbf{\Phi}_{j,i}$. The index set $\{1, 2, \ldots, n\}$ is denoted by $[n]$. The cardinality of a set $\mathcal{A}$ is given by $|\mathcal{A}|$. The probability of an event and a probability density function are represented by $\mathbb{P}(\cdot)$ and $p(\cdot)$, respectively. The expectation of $f(\mathbf{x})$ with respect to the conditional probability distribution $p(\mathbf{x}|\mathbf{y})$ is denoted by $\mathbb{E}_{\mathbf{x}|\mathbf{y}}[f(\mathbf{x})]$. A multivariate Gaussian probability density with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Sigma}$ is denoted by $\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$.

II. System Model and Background

A. Decentralized Joint Sparse Signal Recovery Problem

We consider a network consisting of L computing nodes enumerated by the index set $\mathcal{J} = \{1, 2, \ldots, L\}$. The network connectivity is represented by an undirected graph $\mathcal{G}$ whose vertices have a one-to-one correspondence with the network nodes. An edge between the i^{th} and j^{th} vertices of $\mathcal{G}$ represents an error free communication link between node i and node j. Each node $j \in \mathcal{J}$ communicates only with its single-hop neighboring nodes, belonging to an index set denoted by $\mathcal{N}_j$.

Each sensor node j is interested in estimating an unknown s-sparse vector $\mathbf{x}_j \in \mathbb{R}^n$ from its $m(\ll n)$ noisy linear measurements $\mathbf{y}_j \in \mathbb{R}^m$, generated according to the local measurement model:

$$\mathbf{y}_j = \mathbf{\Phi}_j \mathbf{x}_j + \mathbf{w}_j, \quad j \in \mathcal{J}. \tag{1}$$

Here, $\mathbf{\Phi}_j$ is an $m \times n$ measurement matrix which is assumed to be known locally at node j. The measurement noise vector w_j is assumed to be zero mean Gaussian with known covariance matrix $\sigma_j^2 \mathbf{I}_m$. The sparse vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L$ are assumed to be *joint sparse* i.e., their nonzero coefficients belong to the same index set $\mathcal{S} \subset [n]$, with $|\mathcal{S}| = s$. In accordance with JSM-2 [1], the nonzero coefficients are uncorrelated within and across the vectors.

The goal here is to estimate the local sparse vector $\mathbf{x}_j$ at node j, for each $j \in \mathcal{J}$, in a decentralized fashion. Decentralized processing here implies that each node is capable of processing its local measurements and is allowed to exchange messages with only its one-hop neighbors. In particular, we seek to devise algorithms that (a) exploit the joint sparsity structure of the vectors to be recovered; and (b) require minimal communication overhead between the nodes.

B. Sparse Bayesian Learning

We start by briefly introducing the Sparse Bayesian Learning (SBL) framework [24] which lies at the core of our proposed algorithm. The SBL algorithm is suitable when each node j estimates its local unknown sparse vector $\mathbf{x}_j$ in a standalone fashion, using only local measurements $\mathbf{y}_j$ generated according to (1). In SBL, a MAP estimate of the unknown sparse vector $\mathbf{x}_j$ is sought by imposing a fictitious parameterized signal prior on $\mathbf{x}_j$ as defined below.

$$p(\mathbf{x}_j; \boldsymbol{\gamma}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\boldsymbol{\gamma}(i)}} \exp\left(-\frac{|\mathbf{x}_j(i)|^2}{2\boldsymbol{\gamma}(i)}\right). \tag{2}$$

The hyperparameter $\boldsymbol{\gamma}(i)$ models the variance of $\mathbf{x}_j(i)$. The unknown hyperparameter vector $\boldsymbol{\gamma} \in \mathbb{R}_+^n$ is chosen by maximizing the Bayesian evidence $p(\mathbf{y}_j; \boldsymbol{\gamma})$, which is equivalent to *maximum likelihood* (ML) estimation of $\boldsymbol{\gamma}$. The ML estimate of $\boldsymbol{\gamma}$ is obtained by an iterative Expectation-Maximization (EM) procedure, comprising the following two steps:

$$\begin{aligned} &\text{E-step:} \quad && Q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^k) = \mathbb{E}_{\mathbf{x}_j|\mathbf{y}_j,\boldsymbol{\gamma}^k}[\log p(\mathbf{y}_j, \mathbf{x}_j; \boldsymbol{\gamma})] \\ &\text{M-step:} \quad && \boldsymbol{\gamma}^{k+1} = \underset{\boldsymbol{\gamma} \in \mathbb{R}_+^n}{\arg\max}\ Q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^k), \end{aligned} \tag{3}$$

where k denotes the EM iteration index. From LMMSE theory [25], the a posteriori probability density $p(\mathbf{x}_j|\mathbf{y}_j, \boldsymbol{\gamma}^k)$ used to evaluate the expectation in the E-step is $\mathcal{N}(\boldsymbol{\mu}_j^k, \mathbf{\Sigma}_j^k)$, with mean $\boldsymbol{\mu}_j^k$ and covariance $\mathbf{\Sigma}_j^k$ given by

$$\begin{aligned} \mathbf{\Sigma}_j^k &= \left(\left(\mathbf{\Gamma}^k\right)^{-1} + \frac{\mathbf{\Phi}_j^T \mathbf{\Phi}_j}{\sigma_j^2}\right)^{-1} \\ \boldsymbol{\mu}_j^k &= \sigma_j^{-2} \mathbf{\Sigma}_j^k \mathbf{\Phi}_j^T \mathbf{y}_j \end{aligned} \tag{4}$$

where $\mathbf{\Gamma} = \operatorname{diag}(\boldsymbol{\gamma})$. The M-step optimization yields the following update rule for $\boldsymbol{\gamma}$:

$$\boldsymbol{\gamma}^{k+1} = \left(\boldsymbol{\mu}_j^k\right)^2 + \operatorname{diag}\left(\mathbf{\Sigma}_j^k\right). \tag{5}$$

By repeatedly iterating between (4) and (5), the EM algorithm converges to a local maximum ($\hat{\gamma}_{\text{ML}}$) of the ML cost $\log p(\mathbf{y}_j; \boldsymbol{\gamma})$. Once $\hat{\gamma}_{\text{ML}}$ is found, the MAP estimate of the local sparse vector $\mathbf{x}_j$ is given by the posterior mean $\boldsymbol{\mu}_j$ in

(4) evaluated at $\hat{\gamma}_{\text{ML}}$. In [24], it has been shown that all local maxima of $\log p(\mathbf{y}_j;\boldsymbol{\gamma})$ are at most m-sparse and hence, the SBL algorithm promotes a sparse MAP estimate of $\mathbf{x}_j$. In practice, to obtain a sparse estimate, estimated coefficients with variance at least one order of magnitude lower than noise variance are forced to zero and excluded from the active support.

SBL can be easily extended to handle the estimation of multiple joint sparse vectors, resulting in its multiple measurement vector or MMV variant called M-SBL [10]. M-SBL is a centralized algorithm which enforces joint sparsity of $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L$ by assuming each vector $\mathbf{x}_j$ to be drawn independently from a common $\mathcal{N}(0, \boldsymbol{\Gamma})$ signal prior, where $\boldsymbol{\Gamma} = \text{diag}(\boldsymbol{\gamma})$. Thus, the joint distribution of $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L]$ is given by

$$\begin{aligned} p(\mathbf{X};\boldsymbol{\gamma}) &= \prod_{j=1}^{L} p(\mathbf{x}_j;\boldsymbol{\gamma}) \\ &= \prod_{j=1}^{L} \frac{1}{(2\pi)^{\frac{m}{2}} |\boldsymbol{\Gamma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\mathbf{x}_j^T \boldsymbol{\Gamma}^{-1}\mathbf{x}_j\right). \quad (6) \end{aligned}$$

Once again, the ML estimate of $\boldsymbol{\gamma}$ is sought by maximizing the joint log-likelihood $\log p(\mathbf{Y};\boldsymbol{\gamma}) = \sum_{j=1}^{L} \log p(\mathbf{y}_j;\boldsymbol{\gamma})$, with $p(\mathbf{y}_j;\boldsymbol{\gamma}) = \mathcal{N}(0, \sigma_j^2 \mathbf{I}_m + \boldsymbol{\Phi}_j \boldsymbol{\Gamma} \boldsymbol{\Phi}_j^T)$. As shown in [10], an iterative EM procedure can be derived to obtain the ML estimate of $\boldsymbol{\gamma}$ in the MMV setup using the following M-step update rule:

$$\boldsymbol{\gamma}^{k+1} = \frac{1}{L}\sum_{j=1}^{L}\left(\left(\mu_j^k\right)^2 + \text{diag}\big(\boldsymbol{\Sigma}_j^k\big)\right). \quad (7)$$

Here, μ_j^k and $\boldsymbol{\Sigma}_j^k$ are evaluated according to (4). A naïve centralized implementation of M-SBL would be a fusion center (FC) based approach, in which each node j communicates its $\mathcal{O}(m)$ sized local measurement vector $\mathbf{y}_j$ to the FC. The FC recovers all the joint sparse vectors and transmits the recovered vectors back to their respective nodes. In the case of distinct measurement matrices at the nodes, each node incurs an additional communication cost of sending its local $m \times n$ sized measurement matrix to the FC. The communication cost associated with the exchange of measurement matrices between the nodes and FC can become prohibitively high, especially if the measurement matrices vary with time. To keep the communication costs low, we seek a decentralized implementation of the M-SBL algorithm. In our previous work, we proposed a decentralized algorithm called CB-DSBL [22], for in-network estimation of the common hyperparameters $\boldsymbol{\gamma}$. Although closely matching M-SBL in performance, each iteration of CB-DSBL entails exchange of $n \times 1$ sized vectors between the single-hop neighboring nodes. In the following section, we propose a new decentralized extension of the SBL algorithm called FB-DSBL, which can estimate the common $\boldsymbol{\gamma}$ with significantly reduced communication complexity.

III. Proposed FB-DSBL Algorithm

In this section, we present the proposed FB-DSBL algorithm, in which each network node learns the same joint sparsity inducing signal prior as the centralized M-SBL algorithm, but in a decentralized fashion. Each node $j (\in \mathcal{J})$ assumes a separate SBL type signal prior $p(\mathbf{x}_j) \sim \mathcal{N}(0, \text{diag}(\boldsymbol{\gamma}_j))$ on its local sparse vector $\mathbf{x}_j$ of interest. Here, $\boldsymbol{\gamma}_j$ denotes the $n \times 1$ hyperparameters of the local SBL prior at node j. We also define two terms: *hard support estimate* and *soft support estimate*, which will be used in the sequel. The *hard support estimate* $\mathbf{b}_j \in \{0,1\}^n$, is an estimate of true binary support $\mathcal{S}$ of the sparse vector at node j. Its computation is discussed in Section III-A. For a given $\mathbf{b}_j$ at node j, we define the *soft support estimate* $\mathbf{g}_j \triangleq \boldsymbol{\gamma}_j \circ \mathbf{b}_j$, where $\circ$ is the element-wise multiplication operator. To take advantage of the common support of the unknown signal vectors, each node j shares its soft support estimate $\mathbf{g}_j$ with its single-hop neighbors after every local SBL iteration. Unlike $\boldsymbol{\gamma}_j$, its censored copy $\mathbf{g}_j$ is highly sparse and hence, node j can share it with its one-hop neighbors at a substantially lower communication cost. The soft support estimates gathered from the neighboring nodes are subsequently used by node j to refine its local estimate of $\boldsymbol{\gamma}_j$ obtained from a standalone EM step.

We now outline the main steps involved in a single iteration of the proposed algorithm:

Step-1 Each node j updates its local hyperparameters $\boldsymbol{\gamma}_j$ according to the local SBL update rule given by (4) and (5).

Step-2 Each node j generates a local hard support estimate $\mathbf{b}_j$. Node j broadcasts the soft support estimate $\mathbf{g}_j = \boldsymbol{\gamma}_j \circ \mathbf{b}_j$ to its single-hop neighbors in $\mathcal{N}_j$.

Step-3 Upon receiving $\mathbf{g}_l$ from all of its neighboring nodes l in $\mathcal{N}_j$, each node j fuses the hard support estimates $\mathbf{b}_l = supp(\mathbf{g}_l)$ to generate an improved binary estimate of the true support $\mathcal{S}$ denoted by $\mathbf{b}_j^{\text{fused}}$.

Step-4 Each node j updates its local hyperparameters $\boldsymbol{\gamma}_j$ to assimilate the available extrinsic information $\{\mathbf{g}_l \mid l \in \mathcal{N}_j\}$ conditioned on $\mathbf{b}_j^{\text{fused}}$.

Step-5 Repeat steps 1-4 until convergence.

Step 1 is the standard EM (SBL) iteration executed locally by each node in a standalone fashion using its own observations and a previous estimate of the hyperparameters. The remaining steps update the hyperparameters at each node based on the coarse information about the common support received from the neighboring nodes, for use in the next iteration of SBL. Since the network is connected, the neighborhoods $\mathcal{N}_j, j \in \mathcal{J}$ are partially overlapping, which allows FB-DSBL to exploit the joint sparsity across the entire network. The details of the computations involved in steps 2, 3 and 4 are fleshed out in the subsections III-A, III-B and III-C, respectively.

A. Computation of Local Hard Support Estimates

Since, under the JSM-2 model, a common $\boldsymbol{\gamma}$ is sought across the nodes, it is desirable that, at the end of the iterations, the estimates of $\boldsymbol{\gamma}_j(i)'$s across the nodes are equal if $i \in \mathcal{S}$, and 0 otherwise. Hence, we essentially want to exchange the values of the entries of $\boldsymbol{\gamma}_j^k$ that are likely to correspond to the true common support $\mathcal{S}$, which is a real valued vector of length s, where s is the sparsity of the vectors to be recovered. If the set of indices that contain the true support $\mathcal{S}$ with high

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSIPN.2016.2632041, IEEE Transactions on Signal and Information Processing over Networks

probability, along with their corresponding hyperparameter values, is exchanged across the nodes, that would suffice for estimating/updating the local estimate of $\boldsymbol{\gamma}$. Hence, the first step is to estimate the true support $\mathcal{S}$, which we denote by $\mathbf{b}_j$, at node j. We do this by setting up a hypothesis test for each of the indices, as described in the sequel.

Recall that $\mathbf{b}_j$ is an n length binary vector representing an estimate of the support at node j, and is computed locally at the node based on its measurements. The i^{th} coefficient of $\mathbf{b}_j$ is detected through a composite hypothesis test, by testing hypothesis $\mathcal{H}_0$, that the coefficient is zero, against hypothesis $\mathcal{H}_1$, that the coefficient is nonzero. Such an approach was first used in [26], where the authors proposed index-wise log-likelihood ratio tests (LLRTs) to prune the hyperparameters corresponding to the inactive coefficients for faster convergence of the EM iterations in the SBL algorithm in a centralized setup. Unlike in [26], our goal here is to use the LLRTs to identify the set of hyperparameters corresponding to the most likely support which will be exchanged with other nodes. For an index $i \in [n]$, we setup the following hypothesis test:

$$\begin{aligned} \mathcal{H}_0: &\quad \mathbf{x}_j(i) = 0 \\ \mathcal{H}_1: &\quad \mathbf{x}_j(i) \neq 0 \end{aligned} \tag{8}$$

or equivalently,

$$\begin{aligned} \mathcal{H}_0: &\quad \boldsymbol{\gamma}_j(i) = 0 \\ \mathcal{H}_1: &\quad \boldsymbol{\gamma}_j(i) > 0. \end{aligned} \tag{9}$$

To keep the notation light, the specificity of the hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$ to node j and coefficient i is not highlighted, however, it is implicitly assumed. Unlike (8), the hypothesis test in (9) is a one sided test, and hence, a uniformly most powerful (UMP) test may exist. For the i^{th} index, we decide in favor of $\mathcal{H}_1$, if

$$\log \frac{p(\mathbf{y}_j; \mathcal{H}_1)}{p(\mathbf{y}_j; \mathcal{H}_0)} \geq \theta \tag{10}$$

where θ is the detection threshold. Note that, the likelihood $p(\mathbf{y}_j; \boldsymbol{\gamma})$ equals $\mathcal{N}(0, \sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_{j,i}^k \mathbf{\Phi}_j^T + \boldsymbol{\gamma}_j(i) \mathbf{\Phi}_{j,i} \mathbf{\Phi}_{j,i}^T)$ under $\mathcal{H}_1$ and $\mathcal{N}(0, \sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_{j,i}^k \mathbf{\Phi}_j^T)$ under $\mathcal{H}_0$. Here, k denotes the current iteration index and, $\tilde{\mathbf{\Gamma}}_{j,i}^k$ $= \text{diag}\big(\boldsymbol{\gamma}_j^k(1), \ldots, \boldsymbol{\gamma}_j^k(i-1), 0, \boldsymbol{\gamma}_j^k(i+1), \ldots, \boldsymbol{\gamma}_j^k(n)\big)$, with the $(i,i)^{\text{th}}$ diagonal entry set equal to zero. By substituting the likelihood functions and simplifying the LLR, it is shown in Appendix A that the above hypothesis test decides in favor of $\mathcal{H}_1$ for index i if

$$\mathcal{T}_j^i(\mathbf{y}_j) = \left\{ \mathbf{\Phi}_{j,i}^T \left(\sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_{j,i}^k \mathbf{\Phi}_j^T \right)^{-1} \mathbf{y}_j \right\}^2 \geq \theta^{'}, \tag{11}$$

and decides in favor of $\mathcal{H}_0$ otherwise. Here, $\theta^{'}$ denotes the detection threshold. Since the detection test metric $\mathcal{T}_j^i$ is independent of the parameter under test, i.e., $\boldsymbol{\gamma}_j(i)$, a UMP test for $\boldsymbol{\gamma}_j(i)$ exists. We normalize the detection test metric $\mathcal{T}_j^i$ to a standard chi-squared distributed random variable with a single degree of freedom under $\mathcal{H}_0$, resulting in the following Neyman-Pearson (NP) test:

Decide $\mathcal{H}_1$ for index i if,

$$\bar{\mathcal{T}}_j^i = \frac{\left\{ \mathbf{\Phi}_{j,i}^T \left(\sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_j^k \mathbf{\Phi}_j^T \right)^{-1} \mathbf{y}_j \right\}^2}{\mathbf{\Phi}_{j,i}^T \left(\sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_{j,i}^k \mathbf{\Phi}_j^T \right)^{-1} \mathbf{\Phi}_{j,i}} \geq \bar{\theta}. \tag{12}$$

For a desired probability of false alarm (PFA) $\alpha \in [0,1]$, the normalized threshold $\bar{\theta}$ is computed offline as

$$\bar{\theta} = \left(\mathcal{Q}^{-1} \left(\frac{\alpha}{2} \right) \right)^2, \tag{13}$$

where $\mathcal{Q}(\cdot)$ is the standard Q-function. In the proposed scheme, α is an algorithm parameter which is common across all the network nodes.

In the k^{th} iteration of the proposed algorithm, node j generates its hard support estimate $\mathbf{b}_j^k$ by performing the NP test (12) for each index $i = 1$ to n. Subsequently, node j also computes the corresponding soft support estimate $\mathbf{g}_j^k = \boldsymbol{\gamma}_j^k \circ \mathbf{b}_j^k$, and broadcasts it to its single-hop neighboring nodes in $\mathcal{N}_j$.

Remark: In [27], the active coefficients are identified by performing component-wise maximization of marginalized likelihood across individual hyperparameters. In this scheme, node j declares i^{th} index as active if the ratio $q_{j,i}^2 / s_{j,i}$ is greater than one, where $q_{j,i} = \mathbf{\Phi}_{j,i}^T \left(\sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_j^k \mathbf{\Phi}_j^T \right)^{-1} \mathbf{y}_j$ and $s_{j,i} = \mathbf{\Phi}_{j,i}^T \left(\sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_{j,i}^k \mathbf{\Phi}_j^T \right)^{-1} \mathbf{\Phi}_{j,i}$. The ratio $q_{j,i}^2 / s_{j,i}$ is shown to be a proxy for SNR of the i^{th} component, thereby suggesting that the active indices must have associated SNR greater than 0 dB. In [28], this rule is generalized to $q_{j,i}^2 / s_{j,i} > \eta$, where η is the predefined SNR of the i^{th} component. It is interesting to note that the ratio $q_{j,i}^2 / s_{j,i}$ is the same as the chi-squared LLRT statistic (12) derived for the proposed FB-DSBL algorithm. In fact, if η is set equal to the NP threshold $\bar{\theta}$, as defined in (13), we obtain the LLRT based criterion for active support detection. Hence, the selection of the PFA parameter α in FB-DSBL offers a principled mechanism to control the sparsity of the hard support estimates, and consequently, the size of the messages exchanged between the network nodes.

B. Fusion of Hard Support Estimates from Local Neighborhood

In this subsection, we discuss how node j combines the soft support estimates $\left\{ \mathbf{g}_l^k \mid l \in \mathcal{N}_j \right\}$ received from its single-hop neighbors to obtain a more accurate estimate of the true binary support $\mathcal{S}$. In each iteration of the proposed algorithm, node j computes a fused binary support estimate called $\mathbf{b}_j^{\text{fused}}$ by applying an element-wise *majority rule* to the locally available binary support estimates $\mathbf{b}_l = supp(\mathbf{g}_l)$, where $l \in \mathcal{N}_j \cup \{j\}$. For index $1 \leq i \leq n$,

$$\mathbf{b}_j^{\text{fused},k}(i) \triangleq \begin{cases} 1 & \text{if } |\mathcal{A}_i^{j,k}| > \left\lceil \dfrac{|\mathcal{N}_j|}{2} \right\rceil \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSIPN.2016.2632041, IEEE Transactions on Signal and Information Processing over Networks

where $\mathcal{A}_i^{j,k} \triangleq \left\{l \in \mathcal{N}_j \cup \{j\} : \mathbf{b}_l^k(i) = 1, \mathbf{b}_l^k = supp(\mathbf{g}_l^k)\right\}$. The above fused estimate of $\mathcal{S}$ is subsequently used in section III-C to further refine the local hyperparameter estimate $\boldsymbol{\gamma}_j^k$ obtained from the local SBL update.

In the proposed algorithm, element-wise majority rule has been used to fuse the binary support estimates from the neighboring nodes. The primary reason for this choice is the lack of knowledge of the probability of detection P_{D} associated with the binary support estimates of the one-hop neighbors. Since the direct exchange of local measurements and measurement matrices between network nodes is not allowed, the optimal "*K-out-of-N*" fusion rule is not implementable. Further motivation for choice of majority fusion rule comes from [29] where it has been shown that the optimal fusion rule for the binary decisions from non-identical sensors has a similar structure as the majority rule.

C. Updating Local Hyperparameters using Extrinsic Information

Now, we present a scheme to update the hyperparameters estimate $\boldsymbol{\gamma}_j^k$ at node j by assimilating the available extrinsic information. The extrinsic information at node j refers to $\left\{\mathbf{g}_l^k \mid l \in \mathcal{N}_j\right\}$, i.e., the collection of soft support estimates gathered from the neighboring nodes. From [10], [22], the optimal scheme to update the local hyperparameter estimate $\boldsymbol{\gamma}_j^k$ at node j is to replace it with the average of the current hyperparameter estimates from all network nodes. However, the optimal scheme cannot be implemented as each node has access to only censored copies of the current hyperparameter estimates of its single-hop neighbors.

The problem of parameter estimation from censored measurements in a distributed setup has been studied in the literature [30], [31]. One of the ways to circumvent the non-availability of hyperparameters associated with indices under the $\mathcal{H}_0$ hypothesis is to replace the missing hyperparameters with their respective ML estimates. We observe that the probability of detection P_{D} for a one sided LLRT (similar to (12)) for i^{th} index is functionally dependent on $\boldsymbol{\gamma}_j(i)$. Then, under the assumption that the expression for $P_{\text{D}}(\boldsymbol{\gamma}(i))$ is the same for all the neighboring nodes, the missing $\boldsymbol{\gamma}(i)$ can be chosen such that the associated P_{D} maximizes the likelihood of locally available binary decisions from the neighboring nodes regarding the i^{th} index. However, this scheme is not suitable for the current situation where P_{D} of a neighboring node also depends on its local measurement matrix and local measurement noise power, which are not available globally. Moreover, for a practical network topology, the number of available binary decisions can be insufficient for robust estimation of the missing parameters.

We now propose a suboptimal but pragmatic rule to update the current local hyperparameter estimate $\boldsymbol{\gamma}_j^k(i)$, $i \in [n]$. The proposed update rule is designed to approximate the M-SBL update rule (7). Conditioned on the majority vote $\mathbf{b}_j^{\text{fused},k}(i)$ computed according to (14), we propose separate update rules for the following two cases.

a) ***Case I:*** $\mathbf{b}_j^{\textit{\textbf{fused}},k}(i) = 0$*:* If the majority vote $\mathbf{b}_j^{\text{fused},k}(i)$ suggests $\mathcal{H}_0$ at the i^{th} location in the k^{th} iteration, $\boldsymbol{\gamma}_j^k(i)$ at node j is set equal to the average of the estimated hyperparameter of all nodes in $\mathcal{N}_j$, with censored hyperparameters replaced with zero. The updated local hyperparameter $\bar{\boldsymbol{\gamma}}_j^k(i)$ is given by:

$$\bar{\boldsymbol{\gamma}}_j^k(i) = \frac{\boldsymbol{\gamma}_j^k(i) + \sum_{l\in\mathcal{N}_j} \mathbf{g}_l^k(i)}{1+|\mathcal{N}_j|}. \tag{15}$$

For index $i \notin \mathcal{S}$, since $\boldsymbol{\gamma}_j^k(i)$ finally converges to zero for all $j \in \mathcal{J}$, replacing the missing hyperparameters with zero turns out be a good approximation.

b) ***Case II:*** $\mathbf{b}_j^{\textit{\textbf{fused}},k}(i) = 1$*:* If the majority vote $\mathbf{b}_j^{\text{fused},k}(i)$ suggests $\mathcal{H}_1$ at the i^{th} location in the k^{th} iteration, we propose to set the hyperparameter $\boldsymbol{\gamma}_j^k(i)$ at node j to equal the average of its own hyperparameter $\boldsymbol{\gamma}_j^k(i)$ and the hyperparameters received from only those neighboring nodes that are in agreement with the majority vote. The updated local hyperparameter $\bar{\boldsymbol{\gamma}}_j^k(i)$ is given by:

$$\bar{\boldsymbol{\gamma}}_j^k(i) = \frac{\boldsymbol{\gamma}_j^k(i) + \sum_{l\in\mathcal{N}_j} \mathbf{g}_l^k(i)}{1+\sum_{l\in\mathcal{N}_j} \mathbf{b}_l(i)}. \tag{16}$$

The selective averaging in (16) can be seen as an unbiased approximation of the hyperparameter update (7) used in the centralized M-SBL algorithm by allowing only the neighboring nodes that are in agreement with the majority vote, $\mathbf{b}_j^{\text{fused},k}(i)$, to contribute to the average. Note that, for a fully connected network, when α equals one, the proposed FB-DSBL algorithm is tantamount to executing the M-step update (7) via a decentralized local averaging algorithm. Finally, we summarize the steps involved in proposed FB-DSBL algorithm as Algorithm 1. In Table I, we provide a stepwise breakdown of the per iteration computational and communication complexity of the algorithm.

In the above exposition, a majority vote based selective averaging procedure has been proposed which combines both the available censored soft samples and the associated hard decisions to estimate the index wise local average of the unknown variance parameter $\boldsymbol{\gamma}(i)$. The proposed updates can be understood as a solution to a theoretical formalism which we will now discuss independently in a simpler setup.

Consider a toy problem in which the goal is to find the average of L scalar random variables x_i, generated according to the model $x_i = x_{\text{true}} + w_i, 1 \leq i \leq L$, where x_{true} is an unknown nonnegative parameter and w_i is a zero mean measurement noise with an unknown probability distribution. Without any knowledge about the statistics of the measurement noise, a reasonable estimator of x_{true} is the sample average $\bar{x}$ which can be computed by solving the optimization: $\bar{x} = \arg\min_{x} \frac{1}{L}\sum_{i=1}^{L}(x - x_i)^2$. Say, in addition to the soft samples x_i, we also have access to side information $\{b_i\}_{i\in[L]}$, where each b_i is an independently generated hard decision variable which takes values 0 or 1 to indicate weather x_{true} is zero or nonzero, respectively. The hard decisions b_i could be in error, and as motivated from many practical setups similar to ours, the associated type-1 and type-2 error probabilities are unknown. A natural formalism which incorporates this extra side information for improved estimation of x_{true} is to solve the regularized optimization problem:

$$\hat{x} = \arg\min_{x} \frac{1}{L}\sum_{i=1}^{L}(x - x_i)^2 + \lambda x^2 \tag{17}$$

where a quadratic penalty x^2 has been introduced to attract the solution towards zero, with the strength of attraction governed by a regularization parameter $\lambda \geq 0$. Clearly, the value of λ must reflect the available side information $\{b_i\}_{i\in[L]}$. Intuitively, λ should increase with the number of hard decisions indicating x_{true} to be zero.

It is easy to check that (17) has a closed form solution: $\hat{x} = \bar{x}\left(\frac{1}{1+\lambda}\right)$. The multiplicative factor $\frac{1}{1+\lambda}$ causes shrinkage of the original unregularized solution $\bar{x}$, thereby accounting for the side information which may be suggesting x_{true} to be zero. To ensure that the shrinkage is proportional to the number of zero hard decisions, we suggest choosing λ as the ratio: $\lambda = \frac{Z}{NZ}$, where Z and $NZ = (L - Z)$ denote the number of hard decisions b_i equal to zero and nonzero, respectively. Note that this choice of λ allows the penalty term to completely disappear when all hard decisions suggest x_{true} to be nonzero, resulting in $\hat{x} = \bar{x}$. Next, we complicate the toy problem further by censoring those soft samples x_i, for which the associated b_i is zero. Due to the censoring of the soft samples, the solution $\hat{x}$ of the regularized optimization (17) can no longer be computed, as $\hat{x}$ depends on $\bar{x}$ whose evaluation requires the uncensored values of all soft samples. To proceed, we evaluate $\hat{x} = \bar{x}\left(\frac{1}{1+\lambda}\right)$ with $\bar{x}$ replaced with its unbiased proxy, $\tilde{x} = \frac{1}{NZ}\sum_{i:b_i\neq 0} x_i$ computed only using uncensored soft samples. Thus, we obtain the regularized estmate $\hat{x} = \left(\frac{1}{NZ}\sum_{i:b_i\neq 0} x_i\right)\left(\frac{NZ}{Z+NZ}\right) = \frac{1}{L}\sum_{i:b_i\neq 0} x_i$, which interestingly is also the FB-DSBL update (15) with x_i's and b_i's representing the per index soft and hard support estimates, respectively.

To account for potential errors in the hard decisions b_i's, the regularized solution (17) is accepted only if $Z > NZ$, i.e., when the majority of the hard decisions indicate that x_{true} is zero, otherwise the unregularized but unbiased estimate $\tilde{x} = \frac{1}{NZ}\sum_{i:b_i\neq 0} x_i$ is accepted to be the solution. This also provides a theoretical underpinning for the other FB-DSBL update (16). In the absence of of knowledge of the type-1 and type-2 error probabilities, the majority rule based selection between the regularized and unregularized solutions turns out to be surprisingly effective in practice, as illustrated via simulations in Section V.

D. Inter-node Communication

As discussed in Section III-A, in every FB-DSBL iteration, each node j broadcasts its local soft support estimate $\mathbf{g}_j$ to its single-hop neighbors in $\mathcal{N}_j$. Although $\mathbf{g}_j$ is an n length vector, in practice, it is found to be a highly sparse vector i.e., most of its entries are equal to zero. From Fig. 1, it can be seen that for a fixed sparsity rate (s/n), the fraction of nonzero coefficients in the soft support estimate remains roughly constant with increasing signal dimension n, which is desirable. Further, from Fig. 2, it can be seen that the sparsity of the soft support estimates grows linearly with the

Algorithm 1: FB-DSBL: Fusion Based Decentralized Sparse Bayesian Learning

```
Input:
$\{\mathbf{y}_j, \mathbf{\Phi}_j, \sigma_j\}_{j=1}^{L}$ , and $\alpha$

Initializations:
$k \leftarrow 1$
$\boldsymbol{\gamma}_j^0 \leftarrow 10^{-3} \cdot \mathbf{1}_{n\times 1} \quad \forall\, j \in \mathcal{J}$
$\Delta = 2\epsilon$

while $(k < k_{max})$ and $(\Delta > \epsilon)$ do
    1a. Local E step: Each node $j$ updates its posterior mean $\boldsymbol{\mu}_j^{k-1}$ and variance $\mathbf{\Sigma}_j^{k-1}$ according to (4).
    1b. Local M step: Each node $j$ updates it local hyperparameter vector: $\boldsymbol{\gamma}_j^k = \text{diag}(\mathbf{\Sigma}_j^{k-1}) + (\boldsymbol{\mu}_j^{k-1})^2$ .
    2a. Each node $j$ generates hard support estimate $\mathbf{b}_j$ by performing index-wise LLRTs as shown in (12).
    2b. Each node $j$ computes the soft support estimate $\mathbf{g}_j^k = \boldsymbol{\gamma}_j^k \circ \mathbf{b}_j^k$ and broadcasts it to all the nodes in $\mathcal{N}_j$.
    3. Each node $j$ computes $\mathbf{b}_j^{\text{fused}}$ by fusing $\left\{\mathbf{b}_l \triangleq supp(\mathbf{g}_l)\right\}_{l\in\mathcal{N}_j}$ using the majority rule (14).
    4. Each node $j$ assimilates available extrinsic information by updating $\boldsymbol{\gamma}_j^k$ to $\bar{\boldsymbol{\gamma}}_j^k$ according to:
    For $1 \leq i \leq n$,
        If $\mathbf{b}_j^{\text{fused}}(i) = 0$: $\bar{\boldsymbol{\gamma}}_j^k(i) = \frac{\boldsymbol{\gamma}_j^k(i)+\sum_{l\in\mathcal{N}_j}\mathbf{g}_l^k(i)}{1+|\mathcal{N}_j|}$
        If $\mathbf{b}_j^{\text{fused}}(i) = 1$: $\bar{\boldsymbol{\gamma}}_j^k(i) = \frac{\boldsymbol{\gamma}_j^k(i)+\sum_{l\in\mathcal{N}_j}\mathbf{g}_l^k(i)}{1+\sum_{l\in\mathcal{N}_j}\mathbf{b}_l(i)}$
    5. $\boldsymbol{\gamma}_j^k \leftarrow \bar{\boldsymbol{\gamma}}_j^k$, $\Delta \leftarrow \min_{j\in\mathcal{J}} \frac{||\boldsymbol{\gamma}_j^k - \boldsymbol{\gamma}_j^{k-1}||_2}{||\boldsymbol{\gamma}_j^{k-1}||_2}$ and $k \leftarrow k+1$.
end

Output:
For $1 \leq j \leq L$,   $\hat{\mathbf{x}}_{j,\text{MAP}} \leftarrow \mu_j^k$
```

TABLE I
COMPUTATIONAL & COMMUNICATION COMPLEXITY ANALYSIS OF A SINGLE ITERATION OF FB-DSBL

Steps in FB-DSBL iteration	Computational complexity per node	Communication complexity per node
Local E-step (4)	$\mathcal{O}(n^2m + m^3)$	0
Local M-step (5)	$\mathcal{O}(n)$	0
Compute $\mathbf{b}_j, \mathbf{g}_j$ (12)	$\mathcal{O}(nm^3)$	0
Compute $\mathbf{b}_j^{\text{fused}}$ (14)	$\mathcal{O}(nL)$	$\mathcal{O}(sL\log n)$
Compute $\bar{\boldsymbol{\gamma}}_j$ (15, 16)	$\mathcal{O}(nL)$	0

TABLE II
COMPARISON IN TERMS OF COMMUNICATION COST

Decentralized algorithm	Total number of message exchanges per iteration
FB-DSBL (proposed)	$\mathcal{O}(sL^2\log n)$
DCSP [17]	$\mathcal{O}(sL^2\log n)$
DCOMP [16]	$\mathcal{O}(nL^2)$
CB-DSBL [22]	$\mathcal{O}(nL^2)$
DRL-1 [7]	$\mathcal{O}(nL^2)$

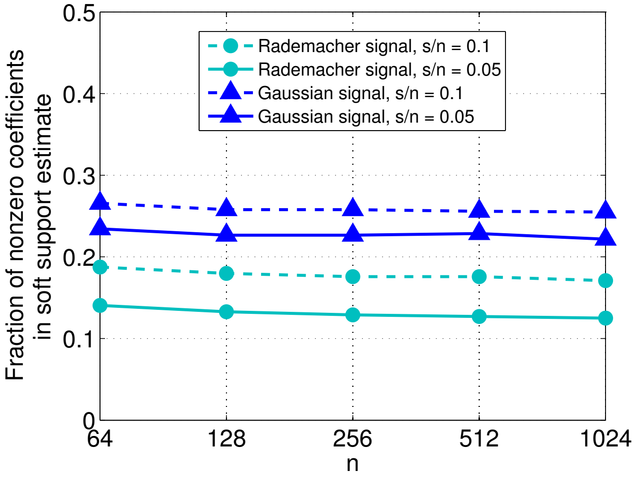


Fig. 1. Fraction of nonzero entries in the soft support estimate $\mathbf{g}_j$ plotted against increasing value of n. The fraction of nonzero entries in $\mathbf{g}_j$ displayed here is averaged across nodes and iterations. The sparse soft support estimates are exchanged between the nodes in each FB-DSBL iteration. The flat curves indicate that the sparsity of the soft support estimates does not change with increasing signal dimension n. Here, SNR = 20 dB, the network size $L = 10$ nodes and the number of measurements at each node is $m = s \log n/s$.

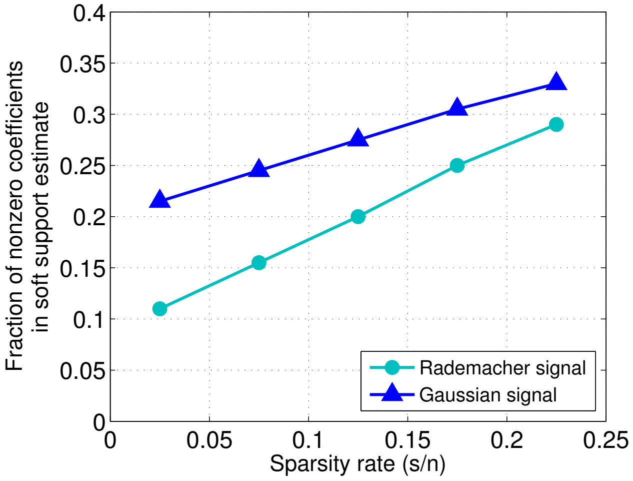


Fig. 2. Fraction of nonzero entries in the soft support estimate $\mathbf{g}_j$ plotted against increasing value of sparsity rate (s/n) for fixed n equal to 200. The fraction of nonzero entries in $\mathbf{g}_j$ displayed here is averaged across nodes and iterations. Here, SNR = 20 dB, the network size $L = 10$ nodes and the number of measurements at each node is $m = s \log n/s$.

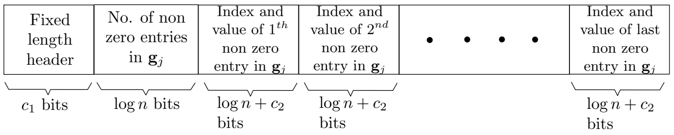
sparsity rate (s/n) of the unknown vectors for fixed signal dimension n. Figures 1 and 2 together imply that by encoding the locations and magnitudes of only the nonzero entries of $\mathbf{g}_j$, the size of the messages exchanged amongst the network nodes can be restricted to $\mathcal{O}(s \log n)$. The additional $\log n$ bits are needed to encode the locations of the individual nonzero coefficients in the soft support estimates. Fig. 3 shows one such example of a frame structure for encoding $\mathbf{g}_j$ using $\mathcal{O}(s \log n)$ information bits. Compared to $\mathcal{O}(n)$ sized messages exchanged in existing decentralized joint sparse signal recovery algorithms [7], [16], [22], the proposed algorithm requires significantly lower communication bandwidth for the recovery of the jointly sparse vectors at the individual nodes.

IV. A Stochastic Approximation View of FB-DSBL

We now present an interesting interpretation of the proposed FB-DSBL algorithm as a degenerate case of a stochastic approximation based distributed algorithm for maximum likelihood estimation of the unknown model parameters $\boldsymbol{\gamma}$. We show that the FB-DSBL updates are a special case of a distributed Robbins-Monro type stochastic approximation updates [32], [33], when the PFA parameter α is set to one. For $\alpha = 1$, the threshold $\bar{\theta}$ used in the index wise LLRT (12) is zero. This in turn causes the hard support estimates $\mathbf{b}_j$'s to be evaluated as $\mathbf{1}_n$, by each node. Thus, the nodes exchange their local hyperparameters estimates as is, without any censoring. In this case, the k^{th} iteration of FB-DSBL at node j comprises the following two steps:

$$\text{Combined EM step:} \qquad \tilde{\boldsymbol{\gamma}}_j^k = G_j(\boldsymbol{\gamma}_j^{k-1}, \mathbf{y}_j), \tag{18}$$

$$\text{Consensus step:} \qquad \boldsymbol{\gamma}_j^k = \frac{1}{|\mathcal{N}_j|} \sum_{l \in \mathcal{N}_j} \tilde{\boldsymbol{\gamma}}_l^k. \tag{19}$$



Fixed length header	No. of non zero entries in $\mathbf{g}_j$	Index and value of 1^{th} non zero entry in $\mathbf{g}_j$	Index and value of 2^{nd} non zero entry in $\mathbf{g}_j$	• • • •	Index and value of last non zero entry in $\mathbf{g}_j$
c_1 bits	$\log n$ bits	$\log n + c_2$ bits	$\log n + c_2$ bits		$\log n + c_2$ bits

Fig. 3. $\mathcal{O}(s \log n)$ sized example data packet format encoding the local soft support estimate $\mathbf{g}_j$, which is broadcast by node j. The scalar constant c_2 controls the quantization noise of nonzero entries of $\mathbf{g}_j$.

The function $G_j : \mathbb{R}_+^n \times \mathbb{R}^m \rightarrow \mathbb{R}_+^n$ replaces the combination of the local E and M-steps given by (4) and (5), respectively, and is evaluated as

$$G_j(\boldsymbol{\gamma}, \mathbf{y}) = \text{diag}\left[\left(\boldsymbol{\Gamma}^{-1} + \frac{\boldsymbol{\Phi}_j^T\boldsymbol{\Phi}_j}{\sigma_j^2}\right)^{-1} + \frac{1}{\sigma^4}\left(\boldsymbol{\Gamma}^{-1} + \frac{\boldsymbol{\Phi}_j^T\boldsymbol{\Phi}_j}{\sigma^2}\right)^{-1} \boldsymbol{\Phi}_j^T\mathbf{y}\mathbf{y}^T\boldsymbol{\Phi}_j\left(\boldsymbol{\Gamma}^{-1} + \frac{\boldsymbol{\Phi}_j^T\boldsymbol{\Phi}_j}{\sigma_j^2}\right)^{-1}\right] \tag{20}$$

where $\boldsymbol{\Gamma} = \text{diag}(\boldsymbol{\gamma})$. Let us define the function $G : \mathbb{R}_+^{nL} \times \mathbb{R}^{mL} \rightarrow \mathbb{R}_+^{nL}$ as

$$G(\boldsymbol{\gamma}_\ominus, \mathbf{y}_\ominus) = \left[G_1(\boldsymbol{\gamma}_1, \mathbf{y}_1), G_2(\boldsymbol{\gamma}_2, \mathbf{y}_2), \ldots, G_L(\boldsymbol{\gamma}_L, \mathbf{y}_L)\right]^T, \tag{21}$$

where $\boldsymbol{\gamma}_\ominus = \left(\boldsymbol{\gamma}_1^T, \boldsymbol{\gamma}_2^T, \ldots, \boldsymbol{\gamma}_L^T\right)^T$ and $\mathbf{y}_\ominus = \left(\mathbf{y}_1^T, \mathbf{y}_2^T, \ldots, \mathbf{y}_L^T\right)^T$ are $nL \times 1$ and $mL \times 1$ sized concatenated vectors representing the local hyperparameter estimates and the local observations, respectively. With these new definitions, we can rewrite the network wide FB-DSBL iterations in a compact vector form,

$$\boldsymbol{\gamma}_\ominus^k = (\mathbf{W} \otimes \mathbf{I}_n) G\left(\boldsymbol{\gamma}_\ominus^{k-1}, \mathbf{y}_\ominus\right), \tag{22}$$

where $\otimes$ denotes the Kronecker product, and $\mathbf{W}$ is an $L \times L$ matrix with entry $\mathbf{W}_{jl} = \frac{1}{|\mathcal{N}_j|+1}$ when $l \in \mathcal{N}_j \cup \{j\}$ and 0 otherwise. The weight matrix $\mathbf{W}$ defined in this manner encodes the local averaging based consensus step described in (19). Other consensus strategies can be realized in this framework by appropriately choosing the entries of $\mathbf{W}$. For example, by modeling $\mathbf{W}$ as an appropriate random matrix, the vector representation in (22) can support the $\alpha \neq 1$ case. In the ensuing arguments, we elucidate the structural similarities between the FB-DSBL vector iterations in (22) and the distributed Robbins-Monro stochastic approximation algorithm proposed in [32].

We begin by observing that the EM iterations (4), (7) in centralized MSBL can be rewritten as a fixed point iteration:

$$\begin{aligned} \boldsymbol{\gamma}^{k+1} &= \sum_{j=1}^{L} G_j(\boldsymbol{\gamma}^k, \mathbf{y}_j) \\ &= \langle G(\mathbf{1}_L \otimes \boldsymbol{\gamma}^k, \mathbf{y}_\ominus)\rangle \end{aligned} \tag{23}$$

where $\langle \mathbf{x} \rangle \triangleq (\mathbf{1}_L^T \otimes \mathbf{I}_n)\mathbf{x}$ is a vector in $\mathbb{R}^n$ for any vector $\mathbf{x}$ in $\mathbb{R}^{nL}$. From the convergence property of EM [34], the above fixed point iteration (23) converges to one of the stationary points of the log-likelihood $\log p(\mathbf{Y}, \boldsymbol{\gamma})$.

We now setup a distributed stochastic approximation algorithm to find the fixed point of $\langle G(\mathbf{1}_L \otimes \boldsymbol{\gamma}^k, \mathbf{y}_\ominus)\rangle$. As proposed in [32], let us consider a distributed algorithm in which each node j implements a "local step" followed by a "gossip step" as described below:

$$\text{Local step:} \quad \tilde{\boldsymbol{\gamma}}_j^k = \boldsymbol{\gamma}_j^{k-1} + a_k \left(G_j(\boldsymbol{\gamma}_j^{k-1}, \mathbf{u}_j^k) - \boldsymbol{\gamma}_j^{k-1} \right) \tag{24}$$

$$\text{Gossip step:} \quad \boldsymbol{\gamma}^k = \frac{1}{|\mathcal{N}_j|} \sum_{j \in \mathcal{N}_j} \tilde{\boldsymbol{\gamma}}_l^k \tag{25}$$

where a_k is an iteration dependent positive step size satisfying $\sum_{k=1}^{\infty} a_k = \infty$ and $\sum_{k=1}^{\infty} a_k^2 < \infty$. In (24), the observation vectors $\mathbf{u}_j^k$s are drawn independently by node j according to some distribution $p_j(\mathbf{u})$ in every iteration. The distribution $p_j(\mathbf{u})$ depends on the measurement model and measurement noise distribution at node j. Once again, we can rewrite the update equations (24) and (25) together in a compact vector form as

$$\boldsymbol{\gamma}_\ominus^k = (\mathbf{W} \otimes \mathbf{I}_n) \left[\boldsymbol{\gamma}_\ominus^{k-1} + a_k \left(G(\boldsymbol{\gamma}_\ominus^{k-1}, \mathbf{u}_\ominus^k) - \boldsymbol{\gamma}_\ominus^{k-1} \right) \right] \tag{26}$$

where $\mathbf{u}_\ominus = \left(\mathbf{u}_1^T, \mathbf{u}_2^T, \ldots, \mathbf{u}_L^T \right)^T$ is $mL \times 1$ sized concatenated vector representing the combined observations across the nodes. For the above distributed Robbins-Monro type update [35], we introduce the associated *mean field function* $h(\boldsymbol{\gamma}) : \mathbb{R}_+^n \to \mathbb{R}_+^n$ as

$$h(\boldsymbol{\gamma}) = \mathbb{E}_{\mathbf{1}_n \otimes \boldsymbol{\gamma}} \left[\frac{1}{L} \left(\left(\langle G(\mathbf{1}_L \otimes \boldsymbol{\gamma}, \mathbf{u}_\ominus) \rangle - \langle \mathbf{1}_L \otimes \boldsymbol{\gamma} \rangle \right) \right] . \tag{27}$$

where the expectation is evaluated as a conditional mean given the past observations [33]. In [32], the authors have shown that under certain assumptions related to the step size a_k, the weight matrix $\mathbf{W}$ and the function G, the distributed update in (26) converges to one of the zeros/roots of the mean field function h. Further, there is network wide consensus between the nodes with respect to their local hyperparameters $\boldsymbol{\gamma}_j$ upon convergence [32], [33].

In our distributed setup, the nodes have access to only a single observation, and hence are forced to use the same observation repeatedly in all iterations of the stochastic update (26). This repeated use of the observations is modeled as each node j drawing its local observations $\mathbf{u}_j^k$ independently according to the degenerate Dirac-delta distribution, i.e., $p_j(\mathbf{u}) = \delta(\mathbf{u} - \mathbf{y}_j)$. Under this modeling assumption, the stochastic update in (26) follows a deterministic trajectory and converges to the zero of the mean field function $\frac{1}{L}\langle G(\mathbf{1}_L \otimes \boldsymbol{\gamma}, \mathbf{y}_\ominus)\rangle - \langle \mathbf{1}_L \otimes \boldsymbol{\gamma}\rangle$, or equivalently, to one of the fixed points of $\langle G(\mathbf{1}_L \otimes \boldsymbol{\gamma}, \mathbf{y}_\ominus)\rangle$. From (23), we recall that every fixed point of $\langle G(\mathbf{1}_L \otimes \boldsymbol{\gamma}, \mathbf{y}_\ominus)\rangle$ is also a stationary point of the log-likelihood cost considered in MSBL. Hence, the pseudo-stochastic[1] updates proposed in (26) converge to the centralized MSBL solution.

It is interesting to note that for a constant step size $a_k = 1$ and $p_j(\mathbf{u}) = \delta(\mathbf{u} - \mathbf{y}_j)$, the distributed Robbins-Monro stochastic updates in (24)-(25) revert to the FB-DSBL updates in (18)-(19) when the censoring threshold is set to zero (i.e., $\alpha = 1$). In fact, in Section V, we demonstrate through simulations that the proposed FB-DSBL updates with a constant step size ($a_k = 1$) converge significantly faster compared to its stochastic approximation based variant which uses (24) instead of (4)-(5). In the rest of the paper, we refer to the FB-DSBL variant with the local EM step (4)-(5) replaced with the Robbins-Monro update (24) as FB-DSBL†. A detailed performance comparison of FB-DSBL and FB-DSBL† is presented via simulations in the next section.

V. Simulation Results

We now present simulation results to illustrate the efficacy of the proposed FB-DSBL algorithm and compare its performance against the following decentralized joint sparse signal recovery algorithms.

1) DRL-1 - Decentralized Re-weighted ℓ_1 Norm Minimization algorithm proposed in [7].
2) DCOMP - Distributed and Collaborative Orthogonal Matching Pursuit proposed in [16].
3) DCSP - Decentralized and Collaborative Subspace Pursuit proposed in [17].
4) CB-DSBL - Consensus Based Distributed Sparse Bayesian Learning algorithm proposed in [22].
5) FB-DSBL† - The FB-DSBL variant with the local M-step (5) replaced with the stochastic approximation inspired update given by (24).

For each trial, the node connectivity in the network is dictated by a randomly generated Erdős-Renyi graph with a connection probability of $p = 0.8$. The joint sparse vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L$ to be estimated are assumed to be of length $n = 50$ and sharing a common nonzero support which is obtained by randomly selecting $s = 5$ distinct indices out of the set $[n]$. Unless specified otherwise, the nonzero coefficients of the joint sparse vectors are drawn independently from the Rademacher distribution.

Among the algorithms compared here, DCOMP and DCSP require prior knowledge of s, i.e., the size of nonzero support. In the final step of CB-DSBL and FB-DSBL algorithms, the active support is identified by element-wise thresholding the local hyperparameter vector $\boldsymbol{\gamma}_j$ at node j using the thresholds

[1] The prefix pseudo here emphasizes the fact that the stochastic updates are driven by repeated use of the same observations at every node.

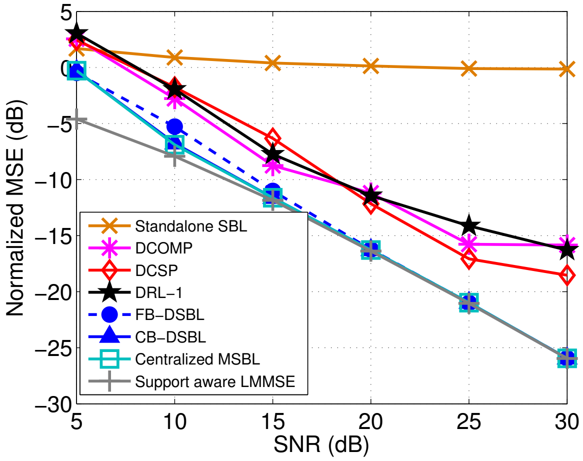


Fig. 4. Normalized mean squared error of the signals (nonzero coefficients from the Rademacher distribution) reconstructed by different algorithms versus the measurement SNR. Other simulation parameters: $n = 50$, $m = 10$, $s = 5$ and $L = 10$ nodes, and 200 trials.

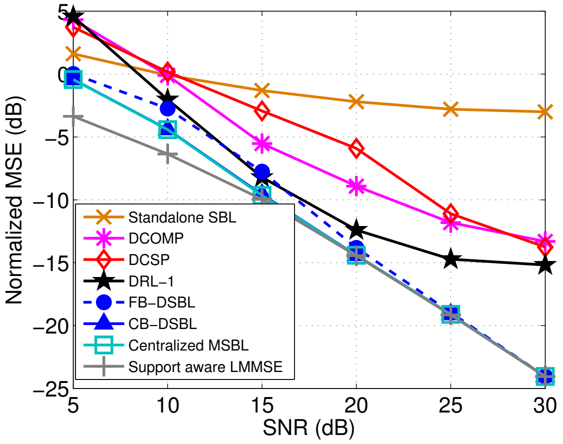


Fig. 5. Normalized mean squared error of the signals (nonzero coefficients from the standard Gaussian distribution) reconstructed by different algorithms versus the measurement SNR. Other simulation parameters: $n = 50$, $m = 10$, $s = 5$, $L = 10$ nodes, and 100 trials.

$4\sigma_j^2$ and $0.25\sigma_j^2$, respectively, where σ_j^2 denotes the local measurement noise variance. For FB-DSBL, the per index probability of false detection, α, is set to 10^{-4} for all the nodes. In FB-DSBL†, the step size a_k is set to $k^{-0.51}$, where k is the iteration index.

A. Performance vs. SNR

In the first set of experiments, we compare the average normalized mean squared error (NMSE) of the signals reconstructed by different algorithms over a wide range of SNRs. The performance benchmark is set by the support-aware linear minimum mean squared error (SA-LMMSE) estimator which assumes knowledge of the true support $\mathcal{S}$. We define the NMSE as

$$\text{NMSE} = \frac{1}{L}\sum_{j=1}^{L}\frac{||\mathbf{x}_j - \hat{\mathbf{x}}_j||_2^2}{||\mathbf{x}_j||_2^2}. \tag{28}$$

Here, the size of the network is fixed to $L = 10$ nodes. Fig. 4 compares the NMSE achieved by the different algorithms, averaged over 200 trials. Both CB-DSBL and the proposed FB-DSBL closely match the benchmark performance of SA-LMMSE at moderate to high SNRs. It is interesting to note that despite exchanging only $\mathcal{O}(s \log n)$ sized messages between the nodes, FB-DSBL is able to outperform DCSP, DRL-1 and DCOMP, which are of similar or higher communication complexity. Fig. 5 shows a similar trend in the relative performances of the algorithms when the nonzero coefficients of the unknown sparse vectors are drawn from the standard Gaussian distribution. The plots also highlight that even when the number of available measurements is not sufficient for independent signal reconstruction as depicted by the complete breakdown of the standalone SBL algorithm, the decentralized algorithms are able to recover the signals by exploiting their joint sparsity. Although not shown in the plots to avoid clutter, FB-DSBL† matches the performance of FB-DSBL in case of both the source distributions.

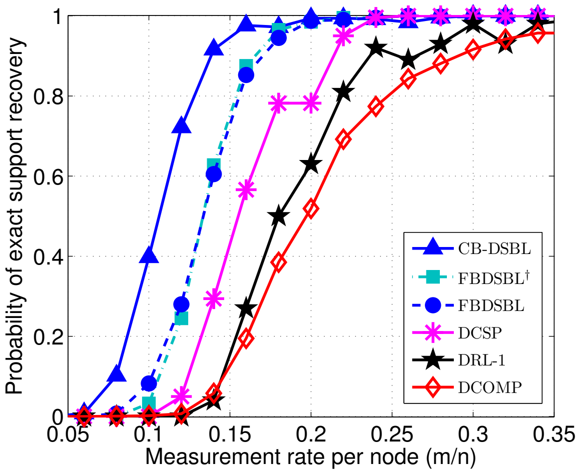


Fig. 6. Probability of exact support recovery versus the measurement rate (m/n) for different decentralized algorithms. Other simulation parameters: $n = 50$, $s = 5$, $L = 10$ nodes, SNR $= 15$ dB, $\alpha = 10^{-4}$ and number of trials = 400.

B. Support Recovery Performance

In the second set of experiments, we compare the support recovery performance of the decentralized algorithms considered here. From Fig. 6, it is evident that FB-DSBL is able to recover the correct support of the unknown sparse vectors using fewer number of measurements per node compared to DRL-1, DCOMP and DCSP. Its stochastic approximation inspired variant FB-DSBL† performs equally well. CB-DSBL has the best support recovery performance among all the decentralized algorithms discussed here, but also has a much higher communication cost compared to FB-DSBL (see Table II).

C. Phase Transition Characteristics

Here, we compare the phase transition characteristics [36] of the different algorithms under MSE and support recovery

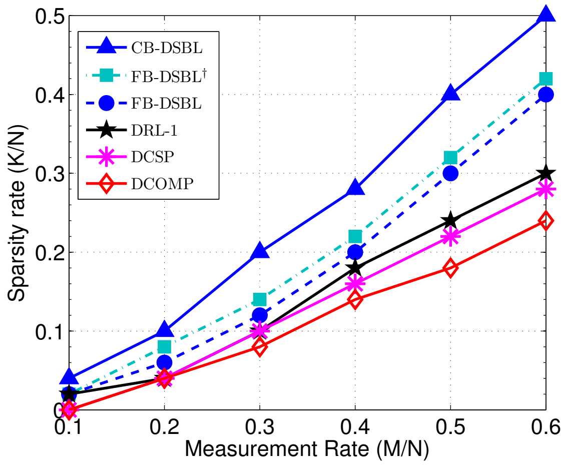


Fig. 7. MSE phase transition for the different algorithms. For a given measurement rate, the phase transition curves represent the maximum sparsity rate of the unknown sparse vectors that can be recovered with at most 1% reconstruction error. Other simulation parameters: $n = 50$, $L = 5$ nodes, SNR $= 30$ dB, $\alpha = 10^{-4}$ and number of trials = 200.

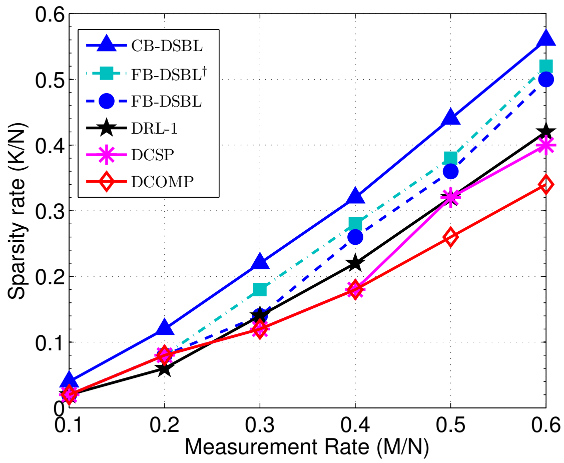


Fig. 8. Support recovery phase transition for the different algorithms. For a given measurement rate, the phase transition curves represent the maximum sparsity rate of the unknown sparse vectors whose nonzero support can be recovered with at least 90% accuracy. Other simulation parameters: $n = 50$, $L = 5$ nodes, SNR $= 30$ dB, $\alpha = 10^{-4}$ and number of trials = 200.

based pass/fail criteria. Fig. 7 plots the MSE phase transition of different algorithms. Any point below the phase transition curve represents a sparsity rate (k/n) and measurement rate (m/n) tuple which results in less than 1% signal reconstruction error. Similarly, in Fig. 8, points below the support recovery phase transition curve represent $(k/n, m/n)$ tuples which result in more than 90% accurate nonzero support reconstruction across all the nodes. From their phase transition behaviors, we conclude that the proposed FB-DSBL is able to recover the support and nonzero signal coefficients from fewer measurements per node compared to DRL-1, DCSP and DCOMP. As before, the CB-DSBL algorithm has the best phase transition characteristics, at the cost of the $\mathcal{O}(nL^2)$ communication complexity per iteration. An interesting observation is that FB-DSBL† has slightly better NMSE and support recovery phase transition characteristics compared to FB-DSBL. This is not surprising, as the filtered updates used in FB-DSBL† are more robust in the presence of measurement noise.

D. Communication Complexity

We also compare the communication overhead of the aforementioned decentralized algorithms. From Fig. 9, the overall communication complexity of FB-DSBL is lower than CB-DSBL, DRL-1, DCOMP and FB-DSBL†, while DCSP still remains the most communication efficient algorithm amongst all the algorithms compared here. As pointed out in Section III-D, in the proposed FB-DSBL algorithm, the nonzero coefficients of the censored vector $\mathbf{g}_j$ broadcast by node j can be represented using a finite number of bits. We study the impact of quantization of the nonzero coefficients of $\mathbf{g}_j$ on the performance of the FB-DSBL algorithm. Here, we have assumed uniform quantization of $\mathbf{g}_j$ in the logarithmic domain in the range $10^{-10} - 10^{5}$. From the MSE phase transition plot in Fig. 10, we observe that there is negligible drop in signal reconstruction performance when 4 or more bits are used to represent the nonzero coefficients of the censored vector.

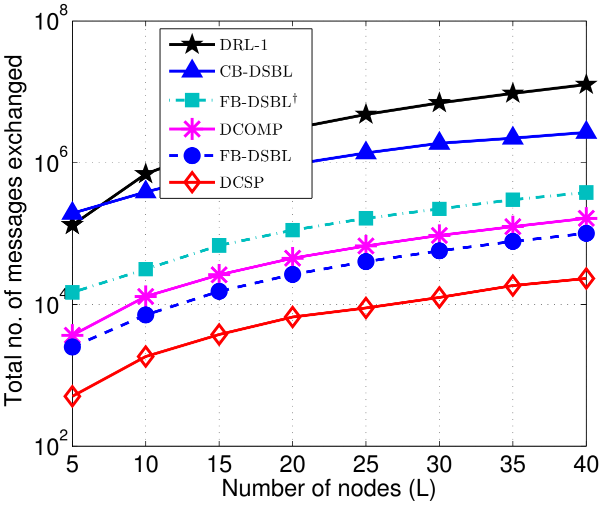


Fig. 9. Average number of messages exchanged between the nodes versus the size of the network. Simulation parameters: $n = 50$, $m = 10$, $s = 5$, SNR $= 20$ dB.

E. Comparison of Convergence Rates

For a decentralized algorithm, the total cost of inter node communication also depends on the number of iterations required for convergence. The centralized M-SBL on which the proposed FB-DSBL is based inherits the convergence guarantees of the underlying EM iterations, which always converge to a local minimum of the log-likelihood objective. Analyzing the convergence of FB-DSBL is non-trivial due to use of censored hyperparameter estimates in the decentralized implementation of (7). However, in practice, it converges within 10-30 iterations, as shown in Fig. 11. FB-DSBL†, on the other hand, converges slower than FB-DSBL, but is faster

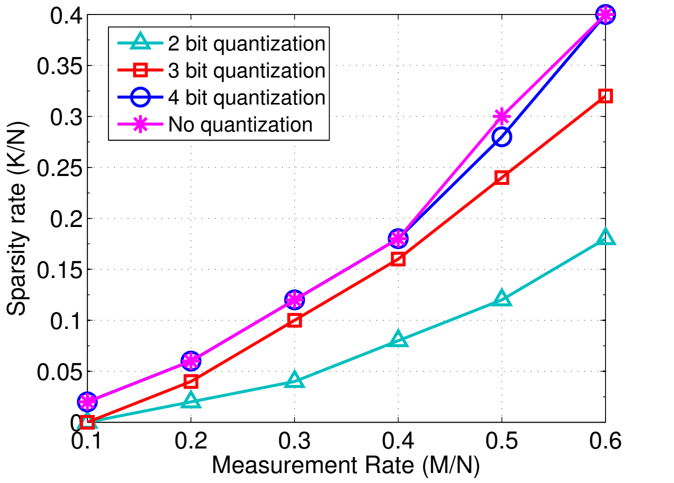


Fig. 10. MSE phase transition for FB-DSBL variants using $2, 3, 4$ bit quantization and analog transmission to encode the nonzero coefficients of soft support estimates exchanged between the nodes. Other simulation parameters: $n = 50$, $L = 5$ nodes, SNR $= 30$ dB, $\alpha = 10^{-4}$ and number of trials $= 1000$.

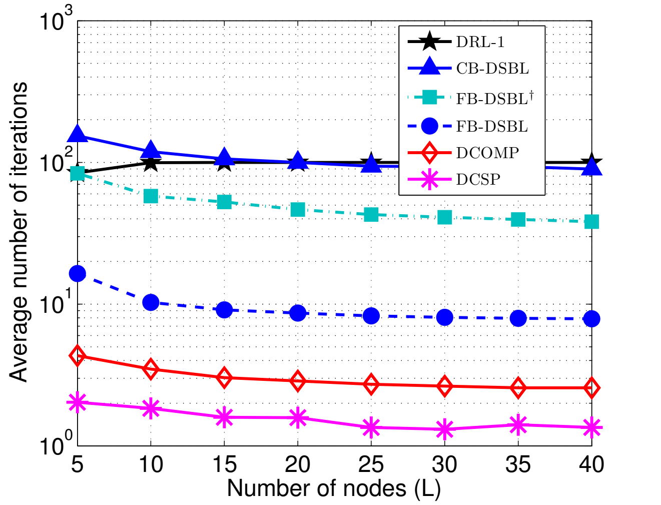


Fig. 11. Average number of iterations versus network size for different decentralized algorithms. Simulation parameters: $n = 50$, $m = 10$, $s = 5$ and SNR= 20 dB.

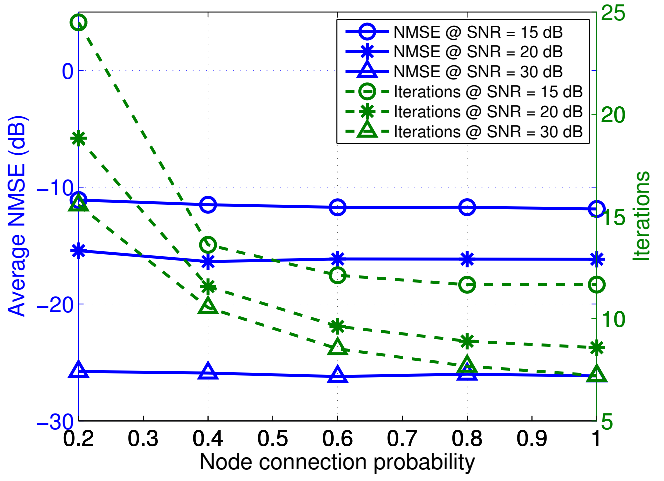


Fig. 12. Illustration of the robustness of FB-DSBL's performance and convergence speed with varying node connection density. Connection probability of one corresponds to a fully connected network. Simulation parameters: $n = 100$, $m = 10$, $s = 5$, $L = 20$ nodes, trials = 100.

than CB-DSBL and DRL-1. In comparison to CB-DSBL, the variance hyperparameters belonging to the non-active support set converge to zero significantly faster in the FB-DSBL, resulting in its faster overall convergence. As expected, due to their greedy approach towards support estimation, both DCSP and DCOMP require the least number of iterations to converge.

From Fig. 12, we observe a small improvement in FB-DSBL's convergence speed as the network becomes increasingly densely connected, with fastest convergence observed in the case of a fully connected network. More importantly, the reconstruction error (measured in NMSE) remains constant independent of the density of node connections in the network, which is a highly desirable attribute for a distributed algorithm. From these experiments, we conclude that FB-DSBL exhibits stable performance and converges under wide variations in network size and node connection density, provided the network remains connected.

F. Selection of parameter α

The FB-DSBL parameter α has a direct impact on the size of messages exchanged between the nodes. As described in section III-A, α represents the probability of false alarm used in the index-wise LLRTs for generation of hard support estimates at each node. Choosing a lower value for α biases the index-wise LLRTs towards the $\mathcal{H}_0$ hypothesis, leading to sparser hard/soft support estimates, and consequently, smaller message size. Likewise, higher values of α lead to larger message size. In fact, for $\alpha = 1$, there is no censoring of the locally estimated hyperparameters γ exchanged between the nodes. In this case, the FB-DSBL algorithm reverts to a local averaging based distributed implementation of the M-SBL update rule (7). Fig. 13 illustrates the effect of α on the MSE performance and the communication cost of the FB-DSBL algorithm. As seen in the plots, there exists a stable range of α between 10^{-2} to 10^{-4}, where one can obtain the twin benefits of low communication complexity and good reconstruction performance. A good rule-of-thumb value of α is $0.01/n$, which corresponds to approximately 1% chance of observing a false alarm in one of the n hypothesis tests performed at each node.

VI. Concluding Remarks

Most of the existing decentralized algorithms for joint sparse signal recovery entail inter node exchange of messages whose size is proportional to n, the ambient signal dimension, which is typically very large. This requires the nodes to expend a significant share of their limited time/energy resources for inter node communication. In this work, we address this issue, by proposing a highly communication efficient, decentralized joint sparse signal recovery algorithm

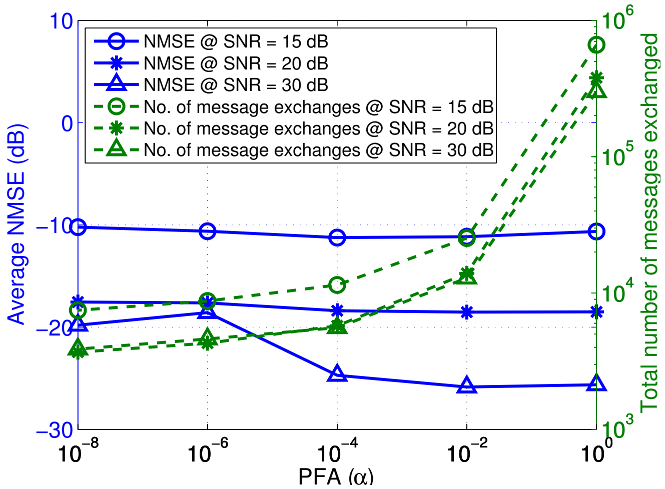


Fig. 13. Plot illustrating the sensitivity of FB-DSBL's performance and communication complexity with respect to the PFA parameter α. Simulation parameters: $n = 50$, $m = 10$, $s = 5$ and $L = 10$ nodes.

called FB-DSBL which requires exchange of only $\mathcal{O}(s \log n)$ sized messages between the network nodes. We showed that the proposed algorithm is a degenerate case of a distributed consensus based stochastic approximation algorithm. From the extensive simulation results presented in this work, we conclude that the proposed FB-DSBL algorithm outperforms existing decentralized algorithms DRL-1, DCSP and DCOMP. In our future work, we plan to exploit the connection of our proposed algorithm with stochastic approximation to derive rigorous convergence guarantees.

Appendix

A. Index-wise LLRT for Hard Support Estimation

Due to the zero mean Gaussian measurement noise and the Gaussian prior $\mathcal{N}(0, \mathbf{\Gamma}_j)$ for the unknown sparse vector $\mathbf{x}_j$ at node j, the likelihood $p(\mathbf{y}_j; \boldsymbol{\gamma}_j^k)$ is given by

$$p(\mathbf{y}_j; \boldsymbol{\gamma}_j) = \mathcal{N}(0, \sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \mathbf{\Gamma}_j \mathbf{\Phi}_j^T). \tag{29}$$

Using (29), we setup the LLRT for i^{th} index at node j as:

Decide in favor of $\mathcal{H}_1$ for index i if,

$$\log \frac{p(\mathbf{y}_j; \boldsymbol{\gamma}_j^k, \boldsymbol{\gamma}_j^k(i) \neq 0)}{p(\mathbf{y}_j; \boldsymbol{\gamma}_j^k, \boldsymbol{\gamma}_j^k(i) = 0)} \geq \theta \tag{30}$$

or equivalently,

$$\log \frac{\mathcal{N}(\mathbf{y}_j; 0, \sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \mathbf{\Gamma}_j^k \mathbf{\Phi}_j^T)}{\mathcal{N}(\mathbf{y}_j; 0, \sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_{j,i}^k \mathbf{\Phi}_j^T)} \geq \theta \tag{31}$$

where $\tilde{\mathbf{\Gamma}}_{j,i}^k$=diag$(\boldsymbol{\gamma}_j^k(1), \ldots.\boldsymbol{\gamma}_j^k(i-1), 0, \boldsymbol{\gamma}_j^k(i+1), \ldots\boldsymbol{\gamma}_j^k(n))$. Using the determinant property: det$(\mathbf{I} + \mathbf{AB})$ = det$(\mathbf{I} + \mathbf{BA})$ and Woodbury matrix identity, (31) simplifies to

$$-\tfrac{1}{2} \log \left(1 + \boldsymbol{\gamma}_j(i) \mathbf{\Phi}_{j,i}^T \left(\sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_{j,i}^k \mathbf{\Phi}_j^T\right)^{-1} \mathbf{\Phi}_{j,i}\right) + \frac{\left(\mathbf{\Phi}_{j,i}^T \left(\sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_{j,i}^k \mathbf{\Phi}_j^T\right)^{-1} \mathbf{y}_j\right)^2}{2\left(\boldsymbol{\gamma}_j(i)^{-1} + \mathbf{\Phi}_{j,i}^T \left(\sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_{j,i}^k \mathbf{\Phi}_j^T\right)^{-1} \mathbf{\Phi}_{j,i}\right)} \geq \theta$$

Moving terms independent of $\mathbf{y}_j$ to the RHS and dividing both sides by the term $\mathbf{\Phi}_{j,i}^T \left(\sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_{j,i}^k \mathbf{\Phi}_j^T\right)^{-1} \mathbf{\Phi}_{j,i}$ yields the Neyman-Pearson test given below

$$\frac{\left(\mathbf{\Phi}_{j,i}^T \left(\sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_{j,i}^k \mathbf{\Phi}_j^T\right)^{-1} \mathbf{y}_j\right)^2}{\mathbf{\Phi}_{j,i}^T \left(\sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_{j,i}^k \mathbf{\Phi}_j^T\right)^{-1} \mathbf{\Phi}_{j,i}} \geq h(\theta, \boldsymbol{\gamma}_j^k(i)) \times \left\{\frac{1}{\boldsymbol{\gamma}_j(i)} + \mathbf{\Phi}_{j,i}^T \left(\sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_{j,i}^k \mathbf{\Phi}_j^T\right)^{-1} \mathbf{\Phi}_{j,i}\right\} \tag{32}$$

where

$$h(\theta, \boldsymbol{\gamma}_j^k(i)) \triangleq \left(\frac{2\theta + \log\left(1 + \boldsymbol{\gamma}_j(i) \mathbf{\Phi}_{j,i}^T \left(\sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_{j,i}^k \mathbf{\Phi}_j^T\right)^{-1} \mathbf{\Phi}_{j,i}\right)}{\mathbf{\Phi}_{j,i}^T \left(\sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_{j,i}^k \mathbf{\Phi}_j^T\right)^{-1} \mathbf{\Phi}_{j,i}}\right).$$

References

[1] M. Duarte, S. Sarvotham, D. Baron, M. Wakin, and R. Baraniuk, "Distributed compressed sensing of jointly sparse signals," in *Proc. Asilomar Conf. on Signals, Syst., and Comput.*, Nov. 2005, pp. 1537–1541.

[2] M. Masood, L. Afify, and T. Al-Naffouri, "Efficient coordinated recovery of sparse channels in massive MIMO," *IEEE Trans. Signal Process.*, vol. 63, no. 1, pp. 104–118, Jan. 2015.

[3] M. Masood and T. Al-Naffouri, "Sparse reconstruction using distribution agnostic Bayesian matching pursuit," *IEEE Trans. Signal Process.*, vol. 61, no. 21, pp. 5298–5309, Nov. 2013.

[4] R. Prasad, C. Murthy, and B. Rao, "Joint approximately sparse channel estimation and data detection in OFDM systems using sparse Bayesian learning," *IEEE Trans. Signal Process.*, vol. 62, no. 14, pp. 3591–3603, Jul. 2014.

[5] J. Bazerque and G. Giannakis, "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1847–1862, Mar. 2010.

[6] F. Zeng, C. Li, and Z. Tian, "Distributed compressive spectrum sensing in cooperative multihop cognitive networks," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 37–48, Feb 2011.

[7] Q. Ling, Z. Wen, and W. Yin, "Decentralized jointly sparse optimization by reweighted ℓ_q minimization," *IEEE Trans. Signal Process.*, vol. 61, no. 5, pp. 1165–1170, Mar. 2013.

[8] J. Le Roux, P. Boufounos, K. Kang, and J. Hershey, "Source localization in reverberant environments using sparse optimization," in *Proc. ICASSP*, May 2013, pp. 4310–4314.

[9] S. Cotter, B. Rao, K. Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2477–2488, Jul. 2005.

[10] D. Wipf and B. Rao, "An empirical Bayesian strategy for solving the simultaneous sparse approximation problem," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3704–3716, Jul. 2007.

[11] H. Lu, X. Long, and J. Lv, "A fast algorithm for recovery of jointly sparse vectors based on the alternating direction methods," in *Proc. AISTATS*, vol. 15, Apr. 2011, pp. 461–469.

[12] M. Hyder and K. Mahata, "A robust algorithm for joint-sparse recovery," *IEEE Signal Process. Lett.*, vol. 16, no. 12, pp. 1091–1094, Dec. 2009.

[13] A. Bollig, S. Corroy, and R. Mathar, "Joint sparse spectrum reconstruction and information fusion via ℓ_1-minimization," in *Proc. VTC*, May 2012, pp. 1–5.

[14] S. Fosson, J. Matamoros, C. Anten-Haro, and E. Magli, "Distributed support detection of jointly sparse signals," in *Proc. ICASSP*, May 2014, pp. 6434–6438.

[15] A. Rakotomamonjy, "Surveying and comparing simultaneous sparse approximation (or group-lasso) algorithms," *Signal Processing*, vol. 91, no. 7, pp. 1505 – 1526, Jul. 2011.

[16] T. Wimalajeewa and P. Varshney, "Cooperative sparsity pattern recovery in distributed networks via distributed OMP," in *Proc. ICASSP*, May 2013, pp. 5288–5292.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSIPN.2016.2632041, IEEE Transactions on Signal and Information Processing over Networks

[17] G. Li, T. Wimalajeewa, and P. Varshney, "Decentralized subspace pursuit for joint sparsity pattern recovery," in *Proc. ICASSP*, May 2014, pp. 3365–3369.

[18] D. Sundman, S. Chatterjee, and M. Skoglund, "Distributed greedy pursuit algorithms," *Signal Processing*, vol. 105, pp. 298 – 315, 2014.

[19] Q. Ling and Z. Tian, "Decentralized support detection of multiple measurement vectors with joint sparsity," in *Proc. ICASSP*, Jan. 2011, pp. 2996–2999.

[20] A. Makhzani and S. Valaee, "Distributed spectrum sensing in cognitive radios via graphical models," in *Proc. CAMSAP*, Dec. 2013, pp. 376–379.

[21] G. Tzagkarakis, J. L. Starck, and P. Tsakalides, "Joint sparse signal ensemble reconstruction in a wsn using decentralized Bayesian matching pursuit," in *Signal Processing Conference, 2011 19th European*, Aug 2011, pp. 338–342.

[22] S. Khanna and C. R. Murthy, "Decentralized joint-sparse signal recovery: A sparse Bayesian learning approach," *IEEE Trans. Signal Inf. Process. over Netw.*, vol. PP, no. 99, pp. 1–1, 2016.

[23] W. Chen and I. J. Wassell, "A decentralized Bayesian algorithm for distributed compressive sensing in networked sensing systems," *IEEE Trans. Wireless Commun.*, vol. 15, no. 2, pp. 1282–1292, Feb 2016.

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

[25] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[26] M. Hurtado, C. Muravchik, and A. Nehorai, "Enhanced sparse Bayesian learning via statistical thresholding for signals in structured noise," *IEEE Trans. Signal Process.*, vol. 61, no. 21, pp. 5430–5443, Nov. 2013.

[27] M. E. Tipping, A. Faul, and J. J. T. Avenue, "Fast marginal likelihood maximisation for sparse Bayesian models," in *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, Jan. 2003, pp. 3–6.

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

[29] S. A. Aldosari and J. M. F. Moura, "Fusion in sensor networks with communication constraints," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, ser. IPSN '04. New York, NY, USA: ACM, Apr. 2004, pp. 108–115.

[30] A. Ribeiro and G. Giannakis, "Bandwidth-constrained distributed estimation for wireless sensor networks-part 1: Gaussian case," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 1131–1143, Mar. 2006.

[31] X. Wu and Z. Tian, "Optimized data fusion in bandwidth and energy constrained sensor networks," in *Proc. ICASSP*, vol. 4, May 2006, pp. 713–716.

[32] P. Bianchi, G. Fort, W. Hachem, and J. Jakubowicz, "Convergence of a distributed parameter estimator for sensor networks with local averaging of the estimates," in *Proc. ICASSP*, May 2011, pp. 3764–3767.

[33] P. Bianchi, G. Fort, and W. Hachem, "Performance of a distributed stochastic approximation algorithm," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7405–7418, Nov 2013.

[34] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[35] S. M. Herbert Robbins, "A stochastic approximation method," *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951.

[36] D. Donoho and J. Tanner, "Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 367, no. 1906, pp. 4273–4293, 2009.

Saurabh Khanna received the B. Tech. degree in Electrical Engineering from the Indian Institute of Technology, Kanpur in 2007. From 2007 to 2016, he was with Texas Instruments, Bangalore working on firmware and algorithm design for WLAN transceivers, Global Navigation Satellite System (GNSS) based user localization and FMCW radars. He is currently pursuing Ph. D. degree in Electrical Communication Engineering at Indian Institute of Science, Bangalore, India. His research interests are in the areas of sparse signal processing, statistical learning theory and structured signal processing.

Chandra R. Murthy (S'03–M'06–SM'11) received the B. Tech. degree in Electrical Engineering from the Indian Institute of Technology, Madras in 1998, the M. S. and Ph. D. degrees in Electrical and Computer Engineering from Purdue University and the University of California, San Diego, in 2000 and 2006, respectively. From 2000 to 2002, he worked as an engineer for Qualcomm Inc., where he worked on WCDMA baseband transceiver design and 802.11b baseband receivers. From Aug. 2006 to Aug. 2007, he worked as a staff engineer at Beceem Communications Inc. on advanced receiver architectures for the 802.16e Mobile WiMAX standard. In Sept. 2007, he joined the Department of Electrical Communication Engineering at the Indian Institute of Science, Bangalore, India, where he is currently working as an Associate Professor.

His research interests are in the areas of energy harvesting communications, multiuser MIMO systems, and sparse signal recovery techniques applied to wireless communications. His paper won the best paper award in the Communications Track in the National Conference on Communications 2014. He was an associate editor for the IEEE Signal Processing Letters during 2012-16. He is an elected member of the IEEE SPCOM Technical Committee for the years 2014-2016, and has been reelected for the years 2017-2019. He is currently serving as the Chair of the IEEE Signal Processing Society, Bangalore Chapter, and as an associate editor for the IEEE Transactions on Signal Processing.