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.2016.2612120, IEEE Transactions on Signal and Information Processing over Networks

Decentralized Joint-Sparse Signal Recovery: A Sparse Bayesian Learning Approach

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

Dept. of ECE, Indian Institute of Science, Bangalore, India
{sakhanna, cmurthy}@ece.iisc.ernet.in

***Abstract*—This work proposes a decentralized, iterative, sparse Bayesian learning (SBL) algorithm for in-network estimation of multiple jointly sparse vectors by a network of nodes, using noisy and underdetermined linear measurements. The proposed algorithm, called *Consensus Based Distributed Sparse Bayesian Learning* or CB-DSBL, exploits the network wide joint sparsity of the unknown sparse vectors to recover them from significantly fewer number of local measurements compared to standalone sparse signal recovery schemes. To reduce the amount of inter-node communication and the associated overheads, the nodes exchange messages with only a small set of bridge nodes. Under this communication scheme, we separately analyze the convergence of the underlying bridge node based Alternating Directions Method of Multipliers (ADMM) iterations used in our proposed algorithm and establish its linear convergence rate. The findings from the convergence analysis of decentralized ADMM are used to accelerate the convergence of the proposed algorithm. Using Monte Carlo simulations as well as real-world data based experiments, we demonstrate the superior performance of our proposed algorithm compared to existing decentralized algorithms: DRL-1, DCOMP and DCSP.**

***Index Terms*—Decentralized Estimation, Distributed Compressive Sensing, Joint Sparsity, Sparse Bayesian Learning, Sensor Networks, Alternating Direction Method of Multipliers.**

I. Introduction

We consider the problem of in-network estimation of multiple joint-sparse vectors by a network of connected agents or processing nodes, using noisy and underdetermined linear measurements. Two or more vectors in $\mathbb{R}^n$ are called *joint-sparse* if, in addition to each vector being individually sparse,[1] their nonzero coefficients belong to a common index set. Joint sparsity occurs naturally in scenarios involving multiple agents trying to learn a sparse representation of a common physical phenomenon. Since the underlying physical phenomenon is the same for all the agents (with similar acquisition modalities), their individual sparse representations/model parameters tend to exhibit joint sparsity. In this work, we consider joint-sparse vectors which belong to Type-2 Joint Sparse Model or JSM-2, one of the three generative models for joint-sparse signals [2]. JSM-2 signal vectors satisfy the property that their nonzero coefficients are uncorrelated within and across the vectors. JSM-2 has been successfully used in several applications such as cooperative spectrum sensing [3], [4], decentralized event detection [5], [6], multi-task compressive sensing [7] and MIMO channel estimation [8]–[10].

This work has appeared in part in [1].

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

[1]A vector in $\mathbb{R}^n$ is said to be k-sparse if only $k(\ll n)$ out of its n coefficients are nonzero.

To further motivate the signal structure of joint sparsity in a distributed setup, consider the problem of detection/classification of randomly occurring events in a field by multiple sensor nodes. Each sensor node j, $1 \leq j \leq L$, employs a dictionary $\Psi_j = [\psi_j^1; \psi_j^2 \dots \psi_j^n]$, whose each column ψ_j^i is the signature corresponding to the i^{th} event, one out of the n events which can potentially occur. In many cases, due to the inability to accurately model the sensing process, the signature vectors ψ_j^i are simply chosen to be the past recordings of j^{th} sensor corresponding to standalone occurrence of the i^{th} event, averaged across multiple experiments [6]. This procedure can result in distinct dictionaries at the individual nodes. For any k $(\ll n)$ events occurring simultaneously, a noisy sensor recording might belong to multiple subspaces, each spanned by different subsets of columns of the local dictionary. In such a scenario, enforcing joint sparsity across the sensor nodes can resolve the ambiguity in selecting the correct subset of columns at each sensor node.

In this work, we consider a distributed setup where each individual joint-sparse vector is estimated by a distinct node in a network comprising multiple nodes, with each node having access to noisy and underdetermined linear measurements of its local sparse vector. By collaborating with each other, these nodes can exploit the joint sparsity of their local sparse vectors to reduce the measurements required per node or improve the quality of their local signal estimates. In [2], it has been shown that the number of local measurements required for common support recovery can be dramatically reduced by exploiting the joint sparsity structure prevalent across the network. In fact, as the nodes increase in number, exact signal reconstruction is possible from as few as k measurements per node, where k denotes the size of the support set. Such a substantial reduction in the number of measurements is highly desirable, especially in applications where the cost or time required to acquire new measurements is high.

Distributed algorithms for JSM-2 signal recovery come in two flavors - centralized and decentralized. In the centralized approach, each node transmits its local measurements to a fusion center (FC) which runs a joint-sparse signal recovery algorithm. The FC then transmits the reconstructed sparse signal estimates back to their respective nodes. In contrast, in a decentralized approach, the goal is to obtain the same centralized solution at all nodes by allowing each node to exchange information with its single hop neighbors in addition to processing its local measurements. Besides being inherently robust to node failures, decentralized schemes also tend to be more energy efficient as the inter-node communication is restricted to relatively short ranges covering only one hop

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.2016.2612120, IEEE Transactions on Signal and Information Processing over Networks

communication links. Decentralized algorithms can also be viewed as a parallel implementation of the recovery algorithm, which is desirable when the computational cost of the overall algorithm is very high, as in big data applications. In this work, we focus on the decentralized approach for solving the sparse signal recovery problem under the JSM-2 signal model.

A. Related Work

In this subsection, we briefly summarize the existing centralized and decentralized algorithms for JSM-2 signal recovery. The earliest work on joint-sparse signal recovery considered extensions of recovery algorithms meant for single measurement vector setup to the centralized multiple measurement vector (MMV) model [11], and demonstrated the significant performance gains that are achievable by exploiting the joint sparsity structure. MMV Basic Matching Pursuit (M-BMP), MMV Orthogonal Matching Pursuit (M-OMP) and MMV FOcal Underdetermined System Solver (M-FOCUSS), introduced in [11], belong to this category. In [12], joint sparsity was exploited for distributed encoding of multiple sparse signals. This work generalized the joint-sparse signals as being generated according to one of the three joint-sparse signal models (JSM-1,2,3). This work also proposed a centralized greedy algorithm called Simultaneous Orthogonal Matching Pursuit (SOMP) [2] for JSM-2 recovery. In [13], an Alternating Direction Method for MMV (ADM-MMV) was proposed which used an ℓ_2/ℓ_1 mixed norm penalty to promote a joint-sparse solution. In [14], the multiple response sparse Bayesian learning (M-SBL) algorithm was proposed as an MMV extension of the SBL algorithm [15]. Unlike the algorithms discussed earlier, M-SBL adopts a hierarchical Bayesian approach by seeking the maximum a posterior probability (MAP) estimate of the JSM-2 signals. In M-SBL, a joint-sparse solution is encouraged by assuming a joint sparsity inducing parameterized prior on the unknown sparse vectors, with the prior parameters learnt directly from the measurements. M-SBL has been shown to outperform deterministic methods based on ℓ_0 norm relaxation such as M-BMP and M-FOCUSS [11] as well as greedy algorithms such as SOMP. AMP-MMV [16] is another Bayesian algorithm which uses Approximate Message Passing (AMP) to obtain marginalized conditional posterior distributions of joint-sparse signals. Owing to their low computational complexity, AMP based algorithms are suitable for recovering signals with large dimensions. However, they have been shown to converge only for large dimensional and randomly constructed measurement matrices. Interested readers are referred to [17] for an excellent study comparing some of the aforementioned centralized JSM-2 signal recovery algorithms.

Among decentralized algorithms, collaborative orthogonal matching pursuit (DCOMP) [18] and collaborative subspace pursuit (DCSP) [19] are greedy algorithms for JSM-2 signal recovery, and both are computationally very fast. However, as demonstrated later in this paper, they do not perform as well as regularization based methods which induce joint sparsity in their solution by employing a suitable penalty. Moreover, both DCOMP and DCSP assume a priori knowledge of the size of the nonzero support set, which could be unknown or hard to estimate. Decentralized row-based LASSO (DR-LASSO) [20] is an iterative alternating minimization algorithm which optimizes a non-convex objective with ℓ_1-ℓ_2 mixed norm based regularization to obtain a joint-sparse solution. In decentralized re-weighted $\ell_1(\ell_2)$ minimization algorithms, or DRL-1,2 [5], a non-convex sum-log-sum penalty is used which induces sparsity more strongly compared to the ℓ_1 norm penalty [21]. The resulting non-convex objective is replaced by a surrogate convex function constructed from iteration dependent weighted ℓ_1/ℓ_2 norm terms. However, both DR-LASSO and DRL-1,2 necessitate cross validation to tune the amount of regularization needed for optimal support recovery performance. DRL-1,2 also requires proper tuning of a so-called smoothing parameter and an ADMM parameter for its optimal performance. By employing a Bayesian approach, we can completely eliminate any need for cross validation, by learning the parameters of a family of signal priors, such that selected signal prior has maximum Bayesian evidence [22], [23]. DCS-AMP [3] is one such decentralized algorithm which employs approximate message passing to learn a parameterized joint sparsity inducing Bernoulli-Gaussian signal prior. Turbo Bayesian Compressive Sensing (Turbo-BCS) [24], another decentralized algorithm, adopts a more relaxed zero mean Gaussian signal prior, with the variance hyperparameters themselves distributed according to an exponential distribution. The relaxed signal prior improves the MSE performance without compromising on the sparsity of the solution. Turbo-BCS, however, involves direct exchange of signal estimates between the nodes, which renders it unsuitable for applications where it is necessary to preserve the privacy of the local signals.

B. Contributions

Our main contributions in this work are as follows:

1) We propose a novel decentralized, iterative, Bayesian joint-sparse signal recovery algorithm called *Consensus Based Distributed Sparse Bayesian Learning* or CB-DSBL. Our proposed algorithm works by establishing network wide consensus with respect to the estimated parameters of a joint sparsity inducing signal prior. The learnt signal prior is subsequently used by the individual nodes to obtain MAP estimates of their local sparse signal vectors.
2) The proposed algorithm employs the Alternating Direction Method of Multipliers (ADMM) to solve a series of iteration dependent consensus optimization problems which require the nodes to exchange messages with each other. To reduce the associated communication overheads, we adopt a bandwidth efficient inter-node communication scheme. This scheme entails the nodes exchanging messages with only a predesignated subset of its single hop neighbors known as *bridge nodes*, as motivated in [25]. In this connection, we analytically establish the relationship between the selected set of bridge nodes and the convergence rate of the ADMM iterations. For the bridge-node based inter-node communication scheme, we show linear rate of convergence for the ADMM iterations when

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.2016.2612120, IEEE Transactions on Signal and Information Processing over Networks

TABLE I: Comparison of Decentralized Joint-Sparse Signal Recovery Algorithms

Decentralized algorithm	Per node, per iteration computational complexity	Per node, per iteration communication complexity	Privacy of local signal estimates	Tunable parameters (if any)	Assumes a priori knowledge of sparsity level
DCSP [19]	$\mathcal{O}(mn+\zeta n+k\log n+m^2)$	$\mathcal{O}(\zeta n+k\log n)$	Yes	None	Yes
DCOMP [18]	$\mathcal{O}(n\zeta+L)$	$\mathcal{O}(\zeta n+L)$	Yes	None	Yes
DRL-1 [5]	$\mathcal{O}((n^2+m^3+nm^2)r_{\max}+\zeta n)$	$\mathcal{O}(\zeta n)$	Yes	Yes	No
DR-LASSO [20]	$\mathcal{O}(n^2mT_1+\zeta nT_2)$	$\mathcal{O}(\zeta nT_2)$	Yes	Yes	No
Turbo-BCS [24]	$\mathcal{O}(n^3+nL+nk^2+k^3+mk)$	$\mathcal{O}(kL)$	No	None	No
DCS-AMP [3]	$\mathcal{O}(mn+\zeta n+c_1 n)$	$\mathcal{O}(\zeta n)$	Yes	Yes	No
CB-DSBL (proposed)	$\mathcal{O}(n^2+m^3+nm^2+\zeta n r_{\max})$	$\mathcal{O}(\zeta n r_{\max})$	Yes	None	No

1. n, m, k and L stand for the dimension of unknown sparse vector, number of local measurements per node, number of nonzero coefficients in the true support and network size, respectively.
2. ζ is the maximum number of communication links activated per node, per communication round.
3. $r_{\max}$ is the number of inner loop ADMM iterations executed per CB-DSBL iteration.
4. In DRL-1, $r_{\max}$ is the number of inner loop ADMM iterations used to obtain an inexact solution to the weighted ℓ_1 norm based subproblem.
5. T_1 and T_2 denote the number of iterations of the two different inner loop iterations executed per DR-LASSO iteration.

applied to a generic consensus optimization problem. The analysis is useful in obtaining a closed form expression for the tunable parameter of our proposed joint sparse signal recovery algorithm, ensuring its fast convergence.

3) We empirically demonstrate the superior MSE and support recovery performance of CB-DSBL compared to existing decentralized algorithms. The experimental results are presented for both synthetic and real world data. In the latter case, we illustrate the performance of the different algorithms for the application of cooperative wideband spectrum sensing in cognitive radios.

In Table I, we compare the existing decentralized joint-sparse signal recovery schemes with respect to their per iteration computational and communication complexity, privacy of local estimates, presence/absence of tunable parameters and dependence on prior knowledge of the sparsity level. As highlighted in the comparison in Table I, CB-DSBL belongs to a handful of decentralized algorithms for joint-sparse signal recovery which do not require a priori knowledge of the sparsity level, rely only on single hop communication, and do not involve direct exchange of local signal estimates between network nodes. Besides this, unlike loopy Belief Propagation (BP) or Approximate Message Passing (AMP) based Bayesian algorithms, CB-DSBL does not suffer from any convergence issues even when the local measurement matrix at each node is dense or not randomly constructed.

The rest of this paper is organized as follows. Section II describes the system model and the problem statement of distributed JSM-2 signal recovery. Section III discusses the centralized M-SBL [14] adapted to our setup, and sets the stage for our proposed decentralized solution. Section IV develops the proposed CB-DSBL algorithm along with a detailed discussion on the convergence properties of the underlying ADMM iterations. Other implementation specific issues are also discussed. Section V compares the performance of proposed algorithm with existing ones with respect to various performance metrics. Section VI illustrates CB-DSBL's performance in a real world application of compressive wideband spectrum sensing. Finally, Section VII concludes the paper.

Notation: Boldface lowercase and uppercase alphabets are used to denote vectors and matrices, respectively. Script styled alphabet (for example $\mathcal{A}$) is used to denote a set. $|\mathcal{A}|$ denotes the cardinality of set $\mathcal{A}$. The term $\mathbf{x}_j^k(i)$ denotes the i^{th} element of vector $\mathbf{x}$ associated with node s_j at k^{th} iteration/time index. The superscript $(.)^T$ denotes the transpose operation. For matrices $\mathbf{A}$ and $\mathbf{B}$ of sizes $m\times n$ and $p\times q$ respectively, $\mathbf{A}\otimes\mathbf{B}$ denotes their Kronecker product, which is of size $mp\times nq$. $\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$ denotes the Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. $\mathbb{E}(\mathbf{x}|\mathbf{y})$ denotes the expectation of random variable/vector $\mathbf{x}$ conditioned on another random variable/vector $\mathbf{y}$.

II. Distributed JSM-2 System Model

We consider a network of L nodes/sensors connected as a network described by a bi-directional graph $\mathcal{G}=(\mathcal{J},\mathcal{A})$. $\mathcal{J}=\{1,2,\ldots,L\}$ is the set of vertices in $\mathcal{G}$, each vertex representing a node in the network. Set $\mathcal{A}$ contains the edges in $\mathcal{G}$, each edge representing a single hop error-free communication link between a distinct pair of nodes. Each node is interested in estimating an unknown k-sparse vector $\mathbf{x}_j\in\mathbb{R}^n$ from m locally acquired noisy linear measurements $\mathbf{y}_j\in\mathbb{R}^m$. The generative model of the local measurement vector $\mathbf{y}_j$ at node j is given by

$$\mathbf{y}_j=\boldsymbol{\Phi}_j\mathbf{x}_j+\mathbf{w}_j,\quad 1\le j\le L \tag{1}$$

where, $\boldsymbol{\Phi}_j\in\mathbb{R}^{m\times n}$ is a full rank sensing matrix and $\mathbf{w}_j\in\mathbb{R}^m$ is the measurement noise modeled as zero mean Gaussian distributed with covariance matrix $\sigma_j^2\mathbf{I}_m$. The sparse vectors $\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_L$ at different nodes follow the JSM-2 signal model [12]. This implies that all $\mathbf{x}_j$ share a common support, represented by the index set $\mathcal{S}$. From the JSM-2 model, it also follows that the nonzero coefficients of the sparse vectors are independent within and across the vectors.

The goal is to recover the sparse vectors $\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_L$ at the respective nodes using decentralized processing. In addition to processing the locally available data $\left\{\mathbf{y}_j,\boldsymbol{\Phi}_j,\sigma_j^2\right\}$ at j^{th} node, each node must collaborate with its single hop neighboring nodes to exploit the network wide joint sparsity of the sparse vectors. The decentralized algorithm should be able to generate the centralized solution at each node, as if each node has access to the global information, i.e., $\left\{\mathbf{y}_j,\boldsymbol{\Phi}_j,\sigma_j^2\right\}_{j\in\mathcal{J}}$.

III. Centralized Algorithm for JSM-2

In this section, we briefly recall the centralized M-SBL algorithm [14] for JSM-2 signal recovery and extend it to

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.2016.2612120, IEEE Transactions on Signal and Information Processing over Networks

support distinct measurement matrices $\mathbf{\Phi}_j$ and noise variances σ_j^2 at each node. The centralized algorithm runs at an FC, which assumes complete knowledge of network wide information, $\left\{\mathbf{y}_j, \mathbf{\Phi}_j, \sigma_j^2\right\}_{j=1}^{L}$. For ease of notation, we introduce two variables $\mathbf{X} \triangleq \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L\}$ and $\mathbf{Y} \triangleq \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L\}$ to be used in the sequel. Similar to M-SBL, each of the sparse vectors $\mathbf{x}_j, j \in \mathcal{J}$ is assumed to be distributed according to a parameterized signal prior $p(\mathbf{x}_j; \boldsymbol{\gamma})$ shown below.

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

Further, the joint signal prior $p(\mathbf{X}; \boldsymbol{\gamma})$ is assumed to be

$$p(\mathbf{X}; \boldsymbol{\gamma}) = \prod_{j \in \mathcal{J}} p(\mathbf{x}_j; \boldsymbol{\gamma}). \tag{3}$$

In the above, $\boldsymbol{\gamma} = (\boldsymbol{\gamma}(0), \boldsymbol{\gamma}(1), \ldots, \boldsymbol{\gamma}(n))^T$ is an n dimensional hyperparameter vector, whose i^{th} entry, $\boldsymbol{\gamma}(i)$, models the common variance of $\mathbf{x}_j(i)$ for $1 \leq j \leq L$. Since the signal priors $p(\mathbf{x}_j; \boldsymbol{\gamma})$ are parameterized by a common $\boldsymbol{\gamma}$, if $\boldsymbol{\gamma}$ has a sparse support $\mathcal{S}$, then the MAP estimates of $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L$ will also be jointly sparse with the same common support $\mathcal{S}$. The Gaussian prior in (2) promotes sparsity as it has an alternate interpretation as a parameterized model for the family of variational approximations to a sparsity inducing Student's t-distributed prior [26]. Under this interpretation, finding the hyperparameter vector $\boldsymbol{\gamma}$ which maximizes the likelihood $p(\mathbf{Y}; \boldsymbol{\gamma})$ is equivalent to finding the variational approximation which has the largest Bayesian evidence.

Let $\hat{\boldsymbol{\gamma}}_{\text{ML}}$ denote the maximum likelihood (ML) estimate of hyperparameters of the joint source prior:

$$\hat{\boldsymbol{\gamma}}_{\text{ML}} = \arg\max_{\boldsymbol{\gamma}} \; p(\mathbf{Y}; \boldsymbol{\gamma}) \tag{4}$$

where $p(\mathbf{Y}; \boldsymbol{\gamma})$ is a type-2 likelihood function obtained by marginalizing the joint density $p(\mathbf{Y}, \mathbf{X}; \boldsymbol{\gamma})$ with respect to the unknown vectors in $\mathbf{X}$, i.e.,

$$\begin{aligned} p(\mathbf{Y}; \boldsymbol{\gamma}) &= \prod_{j=1}^{L} \int p(\mathbf{y}_j | \mathbf{x}_j) p(\mathbf{x}_j; \boldsymbol{\gamma}) \mathrm{d}\mathbf{x}_j \\ &= \prod_{j=1}^{L} \mathcal{N}\left(0, \mathbf{\Phi}_j \mathbf{\Gamma} \mathbf{\Phi}_j^T + \sigma_j^2 \mathbf{I}_m\right). \end{aligned} \tag{5}$$

Here $\mathbf{\Gamma} = \text{diag}(\boldsymbol{\gamma})$. We note that $\hat{\boldsymbol{\gamma}}_{\text{ML}}$ cannot be derived in closed form by directly maximizing the likelihood in (5) with respect to $\boldsymbol{\gamma}$. Hence, as suggested in the SBL framework [15], we use the expectation maximization (EM) procedure to maximize $\log p(\mathbf{Y}; \boldsymbol{\gamma})$ by treating $\mathbf{X}$ as hidden variables.

We now discuss the main steps of the EM algorithm to obtain $\hat{\boldsymbol{\gamma}}_{\text{ML}}$. Let $q_\theta(\mathbf{X})$ denote the variational approximation of true conditional density $p(\mathbf{X}|\mathbf{Y}, \boldsymbol{\gamma})$ with variational parameter set $\theta = (\tilde{\boldsymbol{\mu}}_j, \tilde{\mathbf{\Sigma}}_j)_{j \in \mathcal{J}}$. The variational parameters $\tilde{\boldsymbol{\mu}}_j$ and $\tilde{\mathbf{\Sigma}}_j$ represent the conditional mean and covariance of $\mathbf{x}_j$ given $\mathbf{y}_j$. Then, as shown in [27], the log likelihood admits the following decomposition.

$$\begin{aligned} \log p(\mathbf{Y}; \boldsymbol{\gamma}) \quad &= \quad \int q_\theta(\mathbf{X}) \log \frac{p(\mathbf{Y}, \mathbf{X}; \boldsymbol{\gamma})}{q_\theta(\mathbf{X})} \mathrm{d}\mathbf{X} \\ &+ \quad D\left(q_\theta(\mathbf{X}) \;||\; p(\mathbf{X}|\mathbf{Y}; \boldsymbol{\gamma})\right) \end{aligned} \tag{6}$$

where the term $D(q_\theta||p) = \int q_\theta(\mathbf{X}) \log \frac{q_\theta(\mathbf{X})}{p(\mathbf{X}|\mathbf{Y};\boldsymbol{\gamma})} \mathrm{d}\mathbf{X}$ is the *Kullback-Leibler* (KL) divergence between the probability densities q_θ and p. From the non-negativity of $D(q_\theta||p)$ [28], the log likelihood is lower bounded by the first term in the RHS. In the *E-step*, we choose θ to make this variational lower bound tight by minimizing the KL divergence term.

$$\theta^{k+1} = \arg\min_{\theta} \; D(q_\theta(\mathbf{X}) \;||\; p(\mathbf{X}|\mathbf{Y}, \boldsymbol{\gamma}^k)). \tag{7}$$

Here, k denotes the iteration index of EM algorithm. From the LMMSE theory, $p(\mathbf{x}_j|\mathbf{y}_j, \boldsymbol{\gamma}^k)$ is Gaussian with mean $\boldsymbol{\mu}_j^{k+1}$ and covariance $\mathbf{\Sigma}_j^{k+1}$ given by

$$\begin{aligned} &\mathbf{\Sigma}_j^{k+1} = \mathbf{\Gamma}^k - \mathbf{\Gamma}^k \mathbf{\Phi}_j^T \left(\sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \mathbf{\Gamma}^k \mathbf{\Phi}_j^T\right)^{-1} \mathbf{\Phi}_j \mathbf{\Gamma}^k \\ &\text{and } \boldsymbol{\mu}_j^{k+1} = \sigma_j^{-2} \mathbf{\Sigma}_j^{k+1} \mathbf{\Phi}_j^T \mathbf{y}_j. \end{aligned} \tag{8}$$

By choosing $\theta^{k+1} = \{\boldsymbol{\mu}_j^{k+1}, \mathbf{\Sigma}_j^{k+1}\}_{j \in \mathcal{J}}$ and $q_{\theta^{k+1}}(\mathbf{X}) \sim \prod_{j \in \mathcal{J}} \mathcal{N}(\mathbf{x}_j; \boldsymbol{\mu}_j^{k+1}, \mathbf{\Sigma}_j^{k+1})$, the KL divergence term in (7) can be driven to its minimum value of zero.

In the *M-step*, we choose $\boldsymbol{\gamma}$ to maximize the tight variational lower bound obtained in the *E-step*:

$$\begin{aligned} \boldsymbol{\gamma}^{k+1} &= \arg\max_{\boldsymbol{\gamma}} \int q_{\theta^{k+1}}(\mathbf{X}) \log \frac{p(\mathbf{Y}, \mathbf{X}; \boldsymbol{\gamma})}{q_{\theta^{k+1}}(\mathbf{X})} \mathrm{d}\mathbf{X} \\ &= \arg\max_{\boldsymbol{\gamma}} \mathbb{E}_{\mathbf{X} \sim q_{\theta^{k+1}}} \left[\log p(\mathbf{Y}, \mathbf{X}; \boldsymbol{\gamma})\right]. \end{aligned} \tag{9}$$

As shown in Appendix A, the optimization problem (9) can be recast as the following minimization problem.

$$\boldsymbol{\gamma}^{k+1} = \arg\min_{\boldsymbol{\gamma} \in \mathbb{R}_+^n} \sum_{j \in \mathcal{J}} \sum_{i=1}^{n} \left(\log \boldsymbol{\gamma}(i) + \frac{\mathbf{\Sigma}_j^k(i,i) + \boldsymbol{\mu}_j^k(i)^2}{\boldsymbol{\gamma}(i)}\right). \tag{10}$$

From the zero gradient optimality condition in (10), the M-step reduces to the following update rule:

$$\boldsymbol{\gamma}^{k+1}(i) = \frac{1}{L} \sum_{j \in \mathcal{J}} \left(\mathbf{\Sigma}_j^{k+1}(i,i) + \boldsymbol{\mu}_j^{k+1}(i)^2\right) \quad \text{for } 1 \leq i \leq n. \tag{11}$$

By repeatedly iterating between the E-step (8) and the M-step (11), the EM algorithm converges to either a local maximum or a saddle point of $\log p(\mathbf{Y}; \boldsymbol{\gamma})$ [29]. Once $\hat{\boldsymbol{\gamma}}_{\text{ML}}$ is obtained, the MAP estimate of $\mathbf{x}_j$ is evaluated by substituting it in the expression for $\boldsymbol{\mu}_j$ in (8). It is observed that when the EM algorithm converges, the $\boldsymbol{\gamma}(i)$'s belonging to the inactive support tend to zero, resulting in sparse MAP estimates.

IV. Decentralized Algorithm for JSM-2

A. Algorithm Development

In this section, we develop a decentralized version of the centralized algorithm discussed in the previous section. For notational convenience, we introduce an n length vector $\mathbf{a}_j^k = \left(a_{j,1}^k, a_{j,2}^k, \ldots, a_{j,n}^k\right)^T$ maintained at node j, where $a_{j,i}^k = \mathbf{\Sigma}_j^k(i,i) + \boldsymbol{\mu}_j^k(i)^2$, $\mathbf{\Sigma}_j^k$ and $\boldsymbol{\mu}_j^k$ are as defined in (8).

From (11), we observe that the solution of the M-step optimization (10) can be interpreted as an average of the L

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.2016.2612120, IEEE Transactions on Signal and Information Processing over Networks

vectors $\left\{\mathbf{a}_j^{k+1}\right\}_{j=1}^{L}$. The same solution can also be obtained by solving a different minimization problem

$$\boldsymbol{\gamma}^{k+1} = \arg\min_{\boldsymbol{\gamma}\in\mathbb{R}_+^n} \sum_{j\in\mathcal{J}} \|\boldsymbol{\gamma} - \mathbf{a}_j^{k+1}\|_2^2. \tag{12}$$

Unlike the non-convex M-step objective function in (10), the surrogate objective function in (12) is convex in $\boldsymbol{\gamma}$ and therefore can be minimized in a distributed manner using powerful convex optimization techniques. An alternate form of (12) amenable to distributed optimization is given by

$$\begin{aligned} &\min_{\boldsymbol{\gamma}_j\in\mathbb{R}_+^n,\, j\in\mathcal{J}} \sum_{j\in\mathcal{J}} \|\boldsymbol{\gamma}_j - \mathbf{a}_j^{k+1}\|_2^2 \\ &\text{subject to } \boldsymbol{\gamma}_j = \boldsymbol{\gamma}_{j'} \quad \forall\ j\in\mathcal{J},\ j'\in\mathcal{N}_j \end{aligned} \tag{13}$$

where $\mathcal{N}_j$ denotes the set of single hop neighbors of node j. The equality constraints in (13) ensure its equivalence to the unconstrained optimization in (12). Here, the number of equality constraints is equal to $|\mathcal{A}|$, the total number of single hop links in the network. In a naive decentralized implementation of (13), the number of messages exchanged between the nodes grow linearly with the number of consensus constraints. By restricting the nodes to exchange information only through a small set of pre-designated nodes called *bridge nodes*, the number of consensus constraints can be drastically reduced while preserving the equivalence of (12) and (13). Let $\mathcal{B}\subseteq\mathcal{J}$ denote the set of all bridge nodes in the network and $\mathcal{B}_j\subseteq\mathcal{B}$ denote the set of bridge nodes belonging to the single hop neighborhood of node j, then (13) can be rewritten as

$$\begin{aligned} &\underset{\boldsymbol{\gamma}_j\in\mathbb{R}_+^n, j\in\mathcal{J}}{\text{minimize}} \sum_{j\in\mathcal{J}} \|\boldsymbol{\gamma}_j - \mathbf{a}_j^{k+1}\|_2^2 \\ &\text{subject to } \boldsymbol{\gamma}_j = \boldsymbol{\gamma}_b \quad \forall\ j\in\mathcal{J},\ b\in\mathcal{B}_j. \end{aligned} \tag{14}$$

The auxiliary variables $\boldsymbol{\gamma}_b$, called *bridge parameters*, are used to establish consensus among $\boldsymbol{\gamma}_j$. Each bridge parameter $\boldsymbol{\gamma}_b$ is a non negative n length vector maintained by the bridge node b. As motivated in [25], [30], using bridge nodes to impose network wide consensus allows us to trade off between the communication cost and robustness of the distributed optimization algorithm.[2]

The following Lemma provides sufficient conditions on the choice of the bridge node set $\mathcal{B}$ under which (12) and (14) are equivalent. The proof for the Lemma can be found in [25].

Lemma 1. *For a connected graph $\mathcal{G}$, if the bridge node set $\mathcal{B}\subseteq\mathcal{J}$ satisfies the following conditions*

1) *Each node s_j must be connected to at least one bridge node in $\mathcal{B}$, i.e., $\mathcal{B}_j\neq\phi$ for any $j\in\mathcal{J}$, and,*
2) *If two nodes s_{j_1} and s_{j_2} are single-hop neighbors, then $\mathcal{B}_{j_1}\bigcap\mathcal{B}_{j_2}\neq\phi$ for any $j_1, j_2\in\mathcal{J}$,*

then, in the solution to (14), $\boldsymbol{\gamma}_j$'s are equal for all $j\in\mathcal{J}$.

Fig. 1 illustrates the selection of bridge nodes according to Lemma 1, in a sample network. In this work, we employ

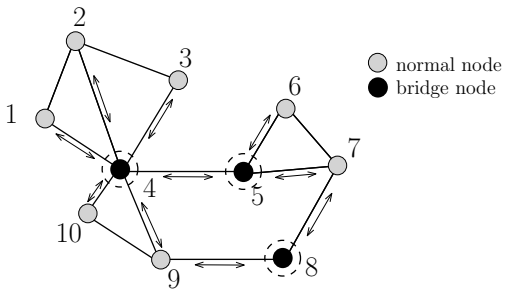


Fig. 1: Selection of bridge nodes in a sample network consisting of 10 nodes. In the proposed scheme, only those edges that have at least one of the vertices as a bridge node are used for communication. The remaining edges are not used for communication. For example, node 9 communicates only with bridge nodes 4 and 8.

the *Alternating Directions Method of Multipliers* (ADMM) algorithm [31] to solve the convex optimization problem in (14). ADMM is the state of the art dual ascent algorithm for solving constrained convex optimization problems, offering a linear convergence rate and a natural extension to a decentralized implementation. We start by constructing an augmented Lagrangian, L_ρ, given by

$$\begin{aligned} L_\rho(\boldsymbol{\gamma}_\mathcal{J}, \boldsymbol{\gamma}_\mathcal{B}, \boldsymbol{\lambda}) \ &\triangleq \ \sum_{j\in\mathcal{J}} \|\boldsymbol{\gamma}_j - \mathbf{a}_j^{k+1}\|_2^2 \\ &+ \sum_{j\in\mathcal{J}}\sum_{b\in\mathcal{B}_j} (\boldsymbol{\lambda}_j^b)^T(\boldsymbol{\gamma}_j - \boldsymbol{\gamma}_b) + \frac{\rho}{2}\sum_{j\in\mathcal{J}}\sum_{b\in\mathcal{B}_j} \|\boldsymbol{\gamma}_j - \boldsymbol{\gamma}_b\|_2^2 \end{aligned} \tag{15}$$

where $\boldsymbol{\lambda}_j^b$ denotes the $n\times 1$ sized Lagrange multiplier vector corresponding to the equality constraint $\boldsymbol{\gamma}_j = \boldsymbol{\gamma}_b$ and ρ is a positive scalar which biases the quadratic consensus penalty term. For lighter notation, we define concatenated vectors $\boldsymbol{\gamma}_\mathcal{J} = \{\boldsymbol{\gamma}_1^T, \boldsymbol{\gamma}_2^T, \ldots, \boldsymbol{\gamma}_L^T\}^T$ and $\boldsymbol{\gamma}_\mathcal{B} = \{\boldsymbol{\gamma}_{b_1}^T, \ldots, \boldsymbol{\gamma}_{b_{|\mathcal{B}|}}^T\}^T$. We also define the $nN_C\times 1$ concatenated Lagrange multiplier vector $\boldsymbol{\lambda}$, where N_C is the number of equality constraints in (14). The solution to (14) is then obtained by executing the following ADMM iterations until convergence:

$$\boldsymbol{\gamma}_\mathcal{J}^{r+1} = \arg\min_{\boldsymbol{\gamma}_\mathcal{J}} L_\rho(\boldsymbol{\gamma}_\mathcal{J}, \boldsymbol{\gamma}_\mathcal{B}^r, \boldsymbol{\lambda}^r) \tag{16}$$

$$\boldsymbol{\gamma}_\mathcal{B}^{r+1} = \arg\min_{\boldsymbol{\gamma}_\mathcal{B}} L_\rho(\boldsymbol{\gamma}_\mathcal{J}^{r+1}, \boldsymbol{\gamma}_\mathcal{B}, \boldsymbol{\lambda}^r) \tag{17}$$

$$(\boldsymbol{\lambda}_j^b)^{r+1} = (\boldsymbol{\lambda}_j^b)^r + \rho(\boldsymbol{\gamma}_j^{r+1} - \boldsymbol{\gamma}_b^{r+1}) \tag{18}$$

$\forall j\in\mathcal{J}, b\in\mathcal{B}_j$. Here, r denotes the ADMM iteration index. In (16-17), the primal variables, $\boldsymbol{\gamma}_\mathcal{J}$ and $\boldsymbol{\gamma}_\mathcal{B}$, are updated in a Gauss-Seidel fashion by minimizing the augmented Lagrangian, L_ρ, evaluated at the previous estimate of the dual variable $\boldsymbol{\lambda}$. By adding an extra quadratic penalty term to the original Lagrangian, the objective in (17) is no longer affine in $\boldsymbol{\gamma}_\mathcal{B}$ and hence has a bounded minimizer. The dual variable $\boldsymbol{\lambda}$ is updated via a gradient-ascent step (18) with a step-size equal to the ADMM parameter ρ. This particular choice of step-size ensures the dual feasibility of the iterates $\{\boldsymbol{\gamma}_\mathcal{J}^{r+1}, \boldsymbol{\gamma}_\mathcal{B}^{r+1}, \boldsymbol{\lambda}^{r+1}\}$ for all r [31]. Since the augmented Lagrangian L_ρ is strictly convex with respect to $\boldsymbol{\gamma}_\mathcal{J}$ and $\boldsymbol{\gamma}_\mathcal{B}$ individually, the zero gradient optimality conditions for (16) and (17) translate into

[2]In an alternate embodiment of the proposed algorithm, the message exchanges could be restricted to occur only through the (trustworthy) bridge nodes, thereby avoiding direct communication between the nodes. In this case, the role of the bridge nodes could be to enforce consensus in $\boldsymbol{\gamma}$ across the nodes, and these nodes need not directly participate in signal reconstruction.

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.2016.2612120, IEEE Transactions on Signal and Information Processing over Networks

simple update equations for $\boldsymbol{\gamma}_j$ and $\boldsymbol{\gamma}_b$:

$$\boldsymbol{\gamma}_j^{r+1} = \frac{2\mathbf{a}_j^{k+1} + \sum_{b\in\mathcal{B}_j}\left(\rho\boldsymbol{\gamma}_b^r - (\boldsymbol{\lambda}_j^b)^r\right)}{2+\rho|\mathcal{B}_j|} \quad \forall\, j\in\mathcal{J} \quad (19)$$

$$\text{and} \quad \boldsymbol{\gamma}_b^{r+1} = \frac{\sum_{j\in\mathcal{N}_b}(\rho\boldsymbol{\gamma}_j^{r+1} + (\boldsymbol{\lambda}_j^b)^r)}{\rho|\mathcal{N}_b|} \quad \forall\, b\in\mathcal{B}. \quad (20)$$

Here $\mathcal{N}_b$ denotes the set of nodes connected to bridge node b. As shown in Appendix B, by eliminating the Lagrange multiplier terms from (18) and (20), the update rule for $\boldsymbol{\gamma}_b$ can be further simplified to

$$\boldsymbol{\gamma}_b^{r+1} = \frac{1}{|\mathcal{N}_b|}\sum_{j\in\mathcal{N}_b}\boldsymbol{\gamma}_j^{r+1} \quad \forall\, b\in\mathcal{B}. \quad (21)$$

In section IV-F, we compare the bridge node based ADMM discussed above with other decentralized optimization techniques in the literature. We show empirically that the bridge node based ADMM scheme is able to flexibly trade off between communication complexity, robustness to node failures, speed of convergence, and signal reconstruction performance. Moreover, the ADMM iterations (16)-(18) can be adapted to handle time varying, asynchronous networks, as suggested in [32]. During the asynchronous operation, the dynamic assignment of bridge nodes can be avoided by designating all nodes as bridge nodes.

B. CB-DSBL Algorithm

We now propose the CB-DSBL algorithm. Essentially, it is a decentralized EM algorithm for finding the ML estimate of the hyperparameters $\boldsymbol{\gamma}$. The algorithm comprises two nested loops. In the outer loop, each node performs the E-step (8) in a standalone manner. In the inner loop, ADMM iterations are performed to solve the M-step optimization in a decentralized manner. Upon convergence of the outer loop, each node $j\in\mathcal{J}$ has the same ML estimate of $\boldsymbol{\gamma}$, which is then used to obtain a MAP estimate of the local sparse vector $\mathbf{x}_j$, similar to the centralized algorithm. The steps of the CB-DSBL algorithm are detailed in Algorithm 1.

Each ADMM iteration in the M-step of the CB-DSBL algorithm involves two rounds of communication (Steps 2 and 4) between the nodes. In the first communication round, each node $j\in J$ transmits $\boldsymbol{\gamma}_j\in\mathbb{R}^n$ to its $|\mathcal{B}_j|$ single hop neighbors. In the second communication round, each bridge node $b\in\mathcal{B}$ transmits $\boldsymbol{\gamma}_b\in\mathbb{R}^n$ to its $|\mathcal{N}_b|$ single hop neighbors. Thus, in each M-step, $2n\sum_{j\in\mathcal{J}}|\mathcal{B}_j|$ real numbers are exchanged between the nodes and their respective bridge nodes. A pragmatic way to select the bridge node set $\mathcal{B}$ is to sort the nodes in decreasing order of their nodal degrees and retain the least number of top most $|\mathcal{B}|$ nodes satisfying the conditions in Lemma 1. In section V, we show empirically that this method of selecting bridge nodes is able to significantly reduce the overall communication complexity of the algorithm.

The inter-node communication can be further optimized by executing only a finite number of ADMM iterations per M-step.[3] In a practical embodiment of the algorithm, running a single ADMM iteration per M-step is sufficient for the CB-DSBL to converge. As shown in Fig. 2, beyond two or three ADMM iterations per M-step, there is only a marginal improvement in the quality of solution as well the convergence speed. Fig. 3 shows that even with a single ADMM iteration per M-step, CB-DSBL typically converges quite rapidly to the centralized solution.

It is also noteworthy to mention that, in CB-DSBL, the nodes are allowed to exchange only their local estimates of the common hyperparameter $\boldsymbol{\gamma}$. Thus, the proposed algorithm is well suited for applications which require the nodes to keep their local signal estimates private.

Algorithm 1 Consensus Based Distributed Sparse Bayesian Learning (CB-DSBL)

Initializations: $k\leftarrow 0$
$\boldsymbol{\gamma}_j^k \leftarrow 10^{-3}\mathbf{1}_{n\times 1} \quad \forall j\in\mathcal{J}$
$\boldsymbol{\gamma}_b^k,\ (\boldsymbol{\lambda}_j^b)^k \leftarrow 0 \quad \forall j\in\mathcal{J},\ b\in\mathcal{B}_j$

while $(k<k_{max})$ & $\left(\Delta\boldsymbol{\gamma}_{\mathcal{J}}>\epsilon\right)$ **do**
E step: Each node s_j, $j\in\mathcal{J}$, updates $\mathbf{a}_j^k$ according to (8).
M step: $r\leftarrow 0$, $\boldsymbol{\gamma}_{\mathcal{J}}^r\leftarrow\boldsymbol{\gamma}_{\mathcal{J}}^k$, $\boldsymbol{\gamma}_{\mathcal{B}}^r\leftarrow\boldsymbol{\gamma}_{\mathcal{B}}^k$, $(\boldsymbol{\lambda})^r\leftarrow(\boldsymbol{\lambda})^k$
while $r<r_{max}$ **do**
1. All nodes $s_{j\in\mathcal{J}}$ update their local estimate of hyperparameters $\boldsymbol{\gamma}_j^r$ according to (19).
2. All nodes $s_{j\in\mathcal{J}}$ transmit the updated $\boldsymbol{\gamma}_j^{r+1}$ estimate to connected bridge nodes $s_{b\in\mathcal{B}_j}$.
3. Each bridge node $s_{b\in\mathcal{B}}$ updates its bridge variable $\boldsymbol{\gamma}_b^r$ according to (21).
4. All bridge nodes $s_{b\in\mathcal{B}}$ transmit updated bridge hyperparameters $\boldsymbol{\gamma}_b^{r+1}$ to nodes in their neighborhood $\mathcal{N}_b$.
5. All nodes $s_{j\in\mathcal{J}}$ update their Lagrange multipliers $(\boldsymbol{\lambda}_j^b)^r, b\in\mathcal{B}_j$ according to (18).
6. $r\leftarrow r+1$

end
$\boldsymbol{\gamma}_{\mathcal{J}}^k\leftarrow\boldsymbol{\gamma}_{\mathcal{J}}^r$, $\boldsymbol{\gamma}_{\mathcal{B}}^k\leftarrow\boldsymbol{\gamma}_{\mathcal{B}}^r$, $(\boldsymbol{\lambda})^k\leftarrow(\boldsymbol{\lambda})^r$
$k\leftarrow k+1$
$\Delta\boldsymbol{\gamma}_{\mathcal{J}}\leftarrow||\boldsymbol{\gamma}_{\mathcal{J}}^k-\boldsymbol{\gamma}_{\mathcal{J}}^{k-1}||_2$
end

C. Convergence of ADMM Iterations in the M-step

In this section, we analyze the convergence of the ADMM iterations (18), (19) and (21) derived for the M-step optimization in CB-DSBL. By doing so, we aim to highlight the effects of the bridge node set $\mathcal{B}$ and the augmented Lagrangian parameter ρ on the convergence of the ADMM iterations.

ADMM has been a popular choice for solving both convex [5], [25], [31], [33], [34] and more recently nonconvex [35] optimization problems as well, in a distributed setup. In its classical form, ADMM solves the following constrained optimization problem:

$$\min_{\mathbf{x},\mathbf{z}} f(\mathbf{x})+g(\mathbf{z})$$
$$\text{subject to } \mathbf{A}\mathbf{x}+\mathbf{B}\mathbf{z}=\mathbf{c}, \quad (22)$$

[3]Even further reduction in internode communication is possible in subsequent rounds of ADMM ($r\geq 2$). Each node j needs to exchange only incremental changes in $\boldsymbol{\gamma}_j^r$, as the initial value $\boldsymbol{\gamma}_j^0$ is already available at the neighboring nodes from the first round of communication.

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.2016.2612120, IEEE Transactions on Signal and Information Processing over Networks

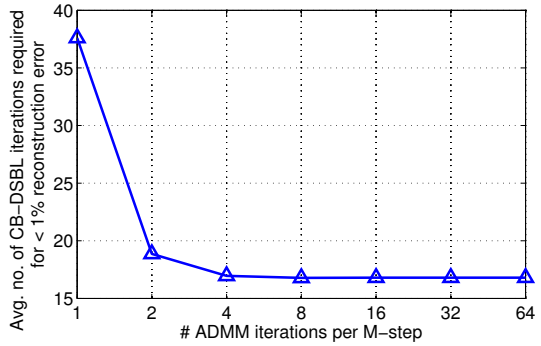
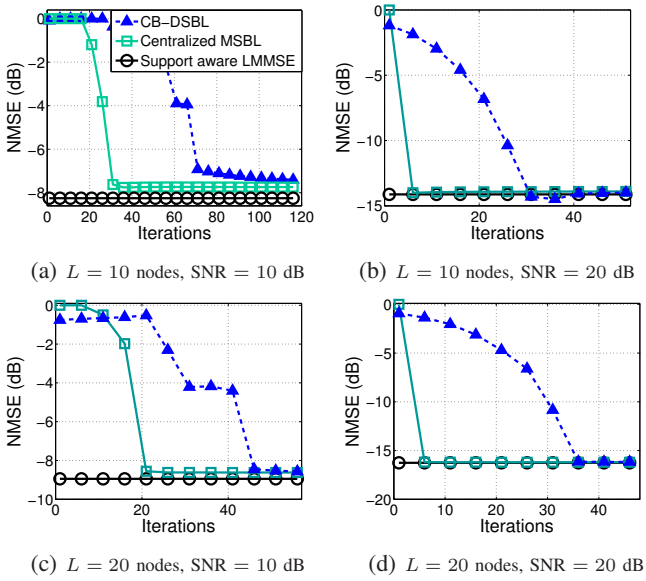


Fig. 2: This plot illustrates the sensitivity of CB-DSBL's outer loop iterations to the number of ADMM iterations executed per M-step in the inner loop of the algorithm. Each point in the curve represents the average number of overall CB-DSBL iterations needed to achieve less than 1% signal reconstruction error for a given number of ADMM iterations executed in the inner loop. Simulation parameters used: $n = 100$, $m = 10$, $L = 10$, 5% sparsity, SNR = 30 dB and #trials = 100.



(a) $L = 10$ nodes, SNR $= 10$ dB (b) $L = 10$ nodes, SNR $= 20$ dB

(c) $L = 20$ nodes, SNR $= 10$ dB (d) $L = 20$ nodes, SNR $= 20$ dB

Fig. 3: Convergence of decentralized CB-DSBL to centralized M-SBL solution for different network sizes and SNRs. The CB-DSBL variant used here executes a single ADMM iteration per EM iteration. Other simulation parameters: $n = 50$, $m = 10$ and 10% sparsity.

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{z} \in \mathbb{R}^m$ are the primal variables. The matrices $\mathbf{A}, \mathbf{B}$ and the vector $\mathbf{c}$ appearing in the linear equality constraint are of appropriate dimensions. The functions $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^m \to \mathbb{R}$ are convex with respect to $\mathbf{x}$ and $\mathbf{z}$, respectively. In [36], the authors have shown linear convergence rate for the classical ADMM iterations under the assumptions of strict convexity and Lipschitz gradient on one of f or g, along with full row rank assumptions for the matrix $\mathbf{A}$. However, in the ADMM formulation of a decentralized consensus optimization problem, the coefficient matrix $\mathbf{A}$ is seldom of full row rank. In [37], the full row rank condition of $\mathbf{A}$ was relaxed and linear rate of convergence was established for decentralized ADMM iterations for a generic convex optimization with linear consensus constraints similar to (13). In [38], the convergence of ADMM for solving an average consensus problem has been analyzed for both

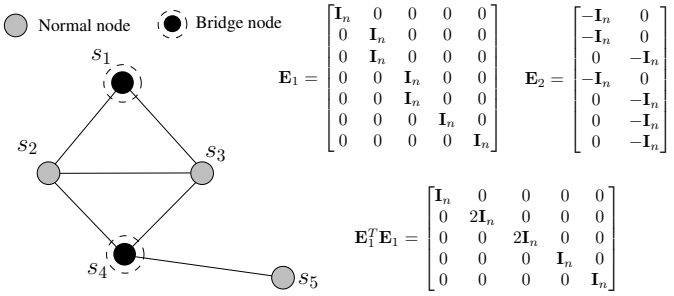


Fig. 4: Construction of block matrices $\mathbf{E}_1$ and $\mathbf{E}_2$ for a sample 5 node network. The matrices $\mathbf{E}_1$ and $\mathbf{E}_2$ are together used to enforce the linear consensus constraints in (13), as shown in (23). Notice the correspondence between the diagonal coefficients of $\mathbf{E}_1^T\mathbf{E}_1$ and the number of bridge node connections per node.

noiseless and noisy communication links. In both [37] and [38], the secondary primary variables indicated by the entries of $\mathbf{z}$ have a one to one correspondence with the communication links between the network nodes. However, such a bijection is not valid for the bridge variables used in our work for enforcing consensus between the primal variables. Due to this, the convergence results of [37], [38] are not applicable to our case. In the sequel, we present the analysis of the convergence of decentralized ADMM iterations for the bridge node internode communication scheme.

We start by defining block matrices $\mathbf{E}_1 = \mathbf{C}_1 \otimes \mathbf{I}_n$ and $\mathbf{E}_2 = \mathbf{C}_2 \otimes \mathbf{I}_n$ of sizes $nN_C \times nL$ and $nN_C \times n|\mathcal{B}|$, respectively. The rows of $\mathbf{C}_1$ and $\mathbf{C}_2$ encode the N_C equality constraints in (14) such that if the i^{th} equality constraint is $\boldsymbol{\gamma}_j = \boldsymbol{\gamma}_{b_k}$, $b_k \in \mathcal{B}$, then $\mathbf{C}_1(i,j) = 1$ and $\mathbf{C}_2(i,k) = -1$; with the rest of the entries in the i^{th} row being zero. It can be shown that the minimum and maximum number of bridge nodes connected to any node in the network is the same as the minimum and maximum eigenvalues of $\mathbf{E}_1^T\mathbf{E}_1$, denoted by $\sigma^2_{\min}$ and $\sigma^2_{\max}$, respectively. Fig. 4 illustrates the construction of the block matrices $\mathbf{E}_1$ and $\mathbf{E}_2$ for an example network consisting of 5 nodes. Using the newly defined terms, the optimization problem in (14) can be rewritten compactly as

$$\min_{\boldsymbol{\gamma}_\mathcal{J}, \boldsymbol{\gamma}_\mathcal{B}} f(\boldsymbol{\gamma}_\mathcal{J}) \quad \text{s.t. } \mathbf{E}_1\boldsymbol{\gamma}_\mathcal{J} + \mathbf{E}_2\boldsymbol{\gamma}_\mathcal{B} = 0 \tag{23}$$

where $f : \mathbb{R}^{nL} \to \mathbb{R}$ denotes the objective function in (14), which depends only on $\boldsymbol{\gamma}_\mathcal{J}$. The augmented Lagrangian L_ρ corresponding to (23) can also be rewritten compactly as

$$\begin{aligned} L_\rho(\boldsymbol{\gamma}_\mathcal{J}, \boldsymbol{\gamma}_\mathcal{B}, \boldsymbol{\lambda}) &= f(\boldsymbol{\gamma}_\mathcal{J}) + \boldsymbol{\lambda}^T(\mathbf{E}_1\boldsymbol{\gamma}_\mathcal{J} + \mathbf{E}_2\boldsymbol{\gamma}_\mathcal{B}) \\ &\quad + \frac{\rho}{2}(\mathbf{E}_1\boldsymbol{\gamma}_\mathcal{J} + \mathbf{E}_2\boldsymbol{\gamma}_\mathcal{B})^T(\mathbf{E}_1\boldsymbol{\gamma}_\mathcal{J} + \mathbf{E}_2\boldsymbol{\gamma}_\mathcal{B}). \end{aligned} \tag{24}$$

By construction, the block matrix $\mathbf{E}_1$ has full column rank, as all its columns are mutually disjoint in support. However, $\mathbf{E}_1$ can be row rank deficient due to repeated rows caused by a node being connected to multiple bridge nodes, which is often the case. Theorem 1 below summarizes the convergence of the ADMM iterations (18), (19) and (21) to their fixed point. The result in Theorem 1 holds for any f that is strongly convex with strong convexity constant m_f, and has a continuous gradient with Lipschitz constant M_f.

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.2016.2612120, IEEE Transactions on Signal and Information Processing over Networks

Theorem 1. *Let $\{\boldsymbol{\gamma}_{\mathcal{J}}^*, \boldsymbol{\gamma}_{\mathcal{B}}^*\}$ and $\boldsymbol{\lambda}^*$ denote the unique primal and dual optimal solutions of (23), and vector $\mathbf{u}$ be constructed as $\mathbf{u} = [(\mathbf{E}_2\boldsymbol{\gamma}_{\mathcal{B}})^T \ \boldsymbol{\lambda}^T]^T$ (similarly for $\mathbf{u}^r, \mathbf{u}^*$). Then, it holds that*

1) The sequence $\mathbf{u}^r$ is Q-linearly[4] convergent to $\mathbf{u}^*$, i.e.,

$$\|\mathbf{u}^{r+1} - \mathbf{u}^*\|_{\mathbf{G}} \leq \frac{1}{1+\delta}\|\mathbf{u}^r - \mathbf{u}^*\|_{\mathbf{G}} \tag{25}$$

where δ is evaluated as

$$\delta = \max_{\mu,\nu\geq 1}\left\{ min\left(\frac{2m_f}{\frac{\nu M_f^2}{\rho(\nu-1)\sigma_{min}^2} + \mu\rho\sigma_{max}^2}, \frac{\sigma_{min}^2}{\nu\sigma_{max}^2}, \frac{\mu-1}{\mu} \right) \right\}. \tag{26}$$

2) The primal sequence $\boldsymbol{\gamma}_{\mathcal{J}}^r$ is R-linearly[5] convergent to $\boldsymbol{\gamma}_{\mathcal{J}}^*$, i.e.,

$$\|\boldsymbol{\gamma}_{\mathcal{J}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{J}}*\|_2 \leq \frac{1}{2m_f}\|\mathbf{u}^r - \mathbf{u}^*\|_{\mathbf{G}} \tag{27}$$

where $\|\cdot\|_{\mathbf{G}}$ is the weighted norm with respect to the diagonal matrix $\mathbf{G} = \text{diag}(\rho I_{n|\mathcal{B}|}, \rho^{-1} I_{N_C})$.

Proof. See Appendices C and D. □

According to Theorem 1, the primal optimality gap $||\boldsymbol{\gamma}_{\mathcal{J}}^r - \boldsymbol{\gamma}_{\mathcal{J}}^*||_2$ decays R-linearly with each ADMM iteration. Moreover, since $\boldsymbol{\gamma}_{\mathcal{J}}^*$ is primal feasible, there is consensus among $\boldsymbol{\gamma}_j, j \in J$ upon convergence, implying that each node effectively minimizes the centralized M-step cost function in (10).

D. Selection of the Augmented Lagrangian Parameter ρ

From (25) and (27) in Theorem 1, we observe that to optimize the decay of the primal optimality gap between $\boldsymbol{\gamma}_{\mathcal{J}}^r$ and $\boldsymbol{\gamma}_{\mathcal{J}}^*$ in each ADMM iteration, the augmented Lagrangian parameter ρ has to be chosen such that it maximizes δ in (26). Theorem 2 reveals the optimal value of ρ and the corresponding value of δ.

Theorem 2. *The optimal value of augmented Lagrangian parameter ρ which uniquely maximizes the δ as defined in (26) is given by*

$$\rho_{opt} = \frac{M_f}{\sigma_{max}\sigma_{min}}\left[\frac{\sqrt{(\kappa-1)^2 + 4\kappa\kappa_f^2} + (\kappa-1)}{\sqrt{(\kappa-1)^2 + 4\kappa\kappa_f^2} - (\kappa-1)}\right]^{\frac{1}{2}}. \tag{28}$$

The corresponding maximal value of δ is given by

$$\delta_{opt} = \frac{2}{\left(\kappa + 1 + \sqrt{(\kappa-1)^2 + 4\kappa\kappa_f^2}\right)} \tag{29}$$

where $\kappa_f = \dfrac{M_f}{m_f}$ represents the condition number of the objective function in (14) and $\kappa = \dfrac{\sigma_{max}^2}{\sigma_{min}^2}$ is the ratio of the maximum and minimum eigenvalues of $\mathbf{E}_1^T\mathbf{E}_1$.

[4]A sequence $x_k : \mathcal{Z}_+ \to \mathbb{R}$ is said to be a *Q-linearly* convergent to L, if there exists a $\mu \in (0,1)$ such that $\lim_{k\to\infty} \frac{|x_{k+1}-L|}{|x_k-L|} = \mu$ [37].

[5]A sequence $x_k : \mathcal{Z}_+ \to \mathbb{R}$ is said to be *R-linearly* convergent to L, if there exists a Q-linearly convergent sequence y_k which converges to zero such that $\lim_{k\to\infty} |x_k - L| \leq y_k$.

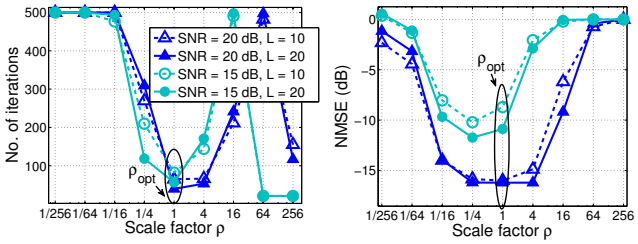


Fig. 5: Left and right plots show the sensitivity of the number of iterations required for convergence and NMSE respectively with respect to the ADMM parameter ρ. The scale factor $\rho = 1$ corresponds to ρ_{opt} in (28).

Proof. See Appendix E. □

From (29), we observe that the convergence rate of the ADMM iteration in the M-step of CB-DSBL algorithm depends upon two factors: κ and κ_f. κ close to its minimum value of unity results in faster convergence of the ADMM iterations. Since the ratio $\kappa = \sigma_{\text{max}}^2 \backslash \sigma_{\text{min}}^2$ is also equal to the ratio of maximum and minimum number of bridge nodes per node in the network, a rule of thumb for bridge node selection would be to ensure that each node is connected to more or less the same number of bridge nodes. The convergence rate also depends upon κ_f, the parameter that determines how well conditioned the function f is. For the case where f is the objective function in (14), it is easy to show that $m_f = M_f = 2$ and $\kappa_f = 1$. Thus, specific to CB-DSBL, the optimal ADMM parameter ρ is given by $\rho_{\text{opt}} = \frac{2}{\sigma_{\text{min}}^2}$ and the corresponding $\delta_{\text{opt}} = \frac{1}{\kappa+1}$. For a given network connectivity graph $\mathcal{G}$, this ρ_{opt} can be computed off-line and programmed in each node. As shown in Fig. 5, the average MSE and mean number of iterations vary widely with ρ, an inappropriate choice of ρ resulting in slow convergence and poor reconstruction performance. Also, the ρ_{opt} computed in (28) is very close to the ρ that results in both the fastest convergence as well as the lowest average MSE.

E. Computational Complexity of CB-DSBL

In this section, we discuss the computational complexity of the steps involved in a single iteration of the CB-DSBL algorithm. The local E-step requires $\mathcal{O}(n^2 + nm^2 + m^3)$ elementary operations at each node. The M-step is executed as multiple (say, r_{max}) ADMM iterations. A single ADMM iteration involves updating of the local hyperparameter estimate $\boldsymbol{\gamma}_j$ and Lagrange multipliers, which takes $\mathcal{O}(\zeta n)$ computations per node, ζ being the highest number of bridge nodes assigned per node in the network. Further, each bridge node $b \in \mathcal{B}$ has to perform an additional $\mathcal{O}(\zeta n)$ computations to update the local bridge parameters $\boldsymbol{\gamma}_b$ in every ADMM iteration. Thus, the overall computational complexity of a single CB-DSBL algorithm at each node is $\mathcal{O}(n^2 + nm^2 + m^3 + \zeta n r_{\text{max}})$. As desired, the computational complexity does not scale with L, i.e., the total number of nodes in the network.

F. Other CB-DSBL Variations

There are several alternatives to the aforementioned bridge node based ADMM technique that could potentially be used

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.2016.2612120, IEEE Transactions on Signal and Information Processing over Networks

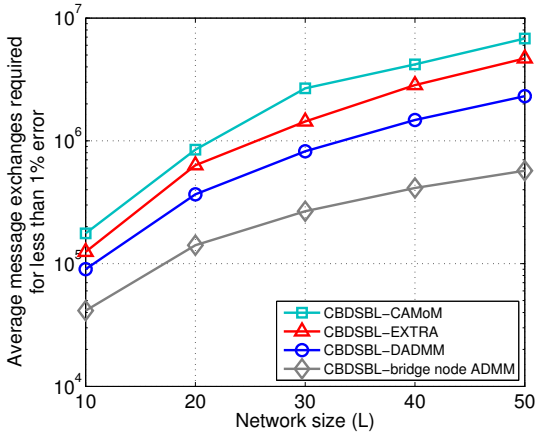


Fig. 6: Comparison of the communication complexity of CB-DSBL variants based on 'bridge node' ADMM [25], CA-MoM [39], D-ADMM [34] and EXTRA [40] algorithms. The plot shows the average number of messages exchanged between nodes in order to achieve less than 1% signal reconstruction error (-20 dB NMSE). Other simulation parameters: $n = 50$, $m = 10$, 10% sparsity, SNR = 30 dB, # trials = 500.

to solve the M-step optimization in (13). In this section, we present empirical results comparing the performance and communication complexity of four different variations of the proposed CB-DSBL algorithm based on (i) bridge node based ADMM [25] (ii) Distributed ADMM (D-ADMM) [34] (iii) Consensus averaging Method of Multipliers (CA-MoM) [39], and (iv) EXact firsT ordeR Algorithm (EXTRA) [40]. The first three ADMM implementations differ in their use of auxiliary variables in enforcing the consensus constraints in (13). Each of these decentralized algorithms is endowed with at least $\mathcal{O}(\frac{1}{k})$ convergence rate, where k stands for the iteration count. Besides these four, there are proximal gradient based methods [41], [42] relying on Nesterov-type acceleration techniques which also offer linear convergence rates. However, these algorithms require the objective function to be bounded and involve multiple communication rounds per iteration, which is of major concern in our work. As shown in Fig. 6, the proposed CB-DSBL variant relying on the bridge node based ADMM scheme is the most communication efficient one.

G. Implementation Issues

CB-DSBL algorithm can be seen as a decentralized EM algorithm to find the ML estimate of the hyperparameters $\boldsymbol{\gamma}$ of a sparsity inducing prior. CB-DSBL, not surprisingly, also inherits the property of the EM algorithm of converging to a local maximum of the ML cost function $\log p(\mathbf{Y}|\boldsymbol{\gamma})$. However, getting trapped in a local maximum is not a problem, as it has been shown in [15] that all local maxima of the $\log p(\mathbf{Y}|\boldsymbol{\gamma})$ are at most m-sparse and hence qualify as reasonably good solutions to our original sparse model estimation problem. In our work, we initialize the EM algorithm with $\boldsymbol{\gamma}$ whose all entries are close to zero.

In practice, hard thresholding of $\boldsymbol{\gamma}$ is required to identify the nonzero support set. In this work, we remove all coefficients from the active support set for which $\boldsymbol{\gamma}(i), 1 \leq i \leq n$ is below the local noise variance. It must be noted that if the local noise variance at each node is unknown, it can also be estimated along with $\boldsymbol{\gamma}$ within the EM framework, as discussed in [14].

Another common issue is that of the wide variation in the energy of the nonzero entries of $\mathbf{x}_j$ across the network. Specifically, in distributed event classification by sensors of different types [6], each sensor node may employ its own distinct sensing modality and hence may perceive a different SNR. In such cases, a preconditioning step which normalizes the local response vector to unit energy is recommended for fast convergence of the CB-DSBL algorithm. The final signal estimates can be re-adjusted to undo the pre-conditioning.

V. Simulation Results

In this section, we present simulation results to examine the performance and complexity aspects of the proposed CB-DSBL algorithm when compared with existing decentralized algorithms: DRL-1 [5], DCOMP [18] and DCSP [19]. The centralized M-SBL [14] is also included in the study as a performance benchmark for CB-DSBL. Since DRL-1 [5] has been shown to have superior performance compared to $\ell_1 - \ell_2$ norm penalty based algorithms, we skip ADM-MMV [13] from our comparisons. The CB-DSBL variant considered here executes two ADMM iterations in the inner loop for every EM iteration in the outer loop. The value of the augmented Lagrangian parameter, ρ, is chosen according to (28). For each experiment, the set $\mathcal{B}$ of bridge nodes is selected as described in section IV-B. The local measurement matrices $\boldsymbol{\Phi}_j$ are chosen to be normalized Gaussian random matrices. The nonzero signal coefficients are sampled independently from the Rademacher distribution, unless mentioned otherwise. For each trial, the connections between the nodes are assumed according to a randomly generated Erdös-Renyi graph with a node connection probability of 0.8. In the final step of M-SBL and CB-DSBL algorithms, the active support is identified by element-wise thresholding the local hyperparameter vector $\boldsymbol{\gamma}_j$ at node j using the threshold $4\sigma_j^2$, where σ_j^2 denotes the local measurement noise variance.

A. Performance versus SNR

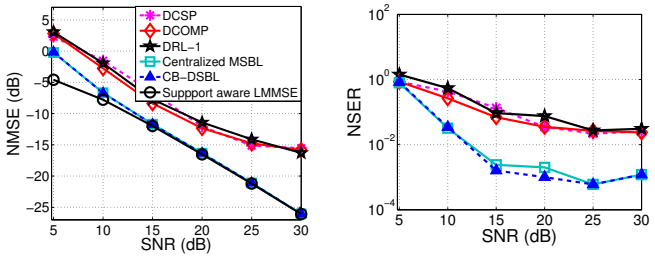
In the first set of experiments, we compare the normalized mean squared error (NMSE) and the normalized support error rate (NSER) of different algorithms for a range of SNRs. The support-aware LMMSE estimator sets the MSE performance benchmark for all the support agnostic algorithms considered here. The NMSE and NSER error metrics are defined as

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

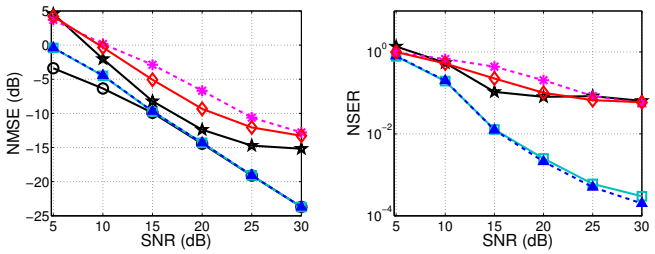
$$\text{NSER} = \frac{1}{L}\sum_{j=1}^{L}\frac{|\mathcal{S}\backslash\hat{\mathcal{S}}_j| + |\hat{\mathcal{S}}_j\backslash\mathcal{S}|}{|\mathcal{S}|}$$

where $\mathcal{S}$ is the true common support and $\hat{\mathcal{S}}_j$ is the support estimated at node j. The network size is fixed to $L = 10$ nodes. As seen in Fig. 7, CB-DSBL matches the performance of centralized M-SBL in all cases. For higher SNR (≥ 15 dB), it can be seen that both M-SBL and proposed CB-DSBL

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.2016.2612120, IEEE Transactions on Signal and Information Processing over Networks



(a) Nonzero coefficients drawn from Rademacher distribution



(b) Nonzero coefficients drawn from Gaussian distribution

Fig. 7: Left and right figures in the above plot the NMSE and NSER respectively for different SNRs. Other simulation parameters: $L = 10$ nodes, $n = 50$, $m = 10$ and 10% sparsity.

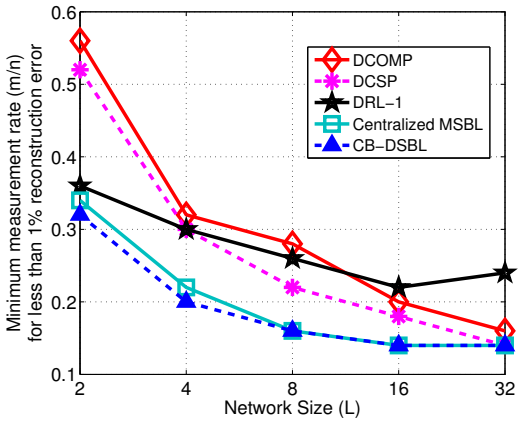


Fig. 8: NMSE phase transition plots of different algorithms illustrating the dependence of minimum measurement rate required to guarantee less than 1% signal reconstruction error on the network size, for signal sparsity rate fixed at 10%. Other simulation parameters: $n = 50$ and SNR = 30 dB.

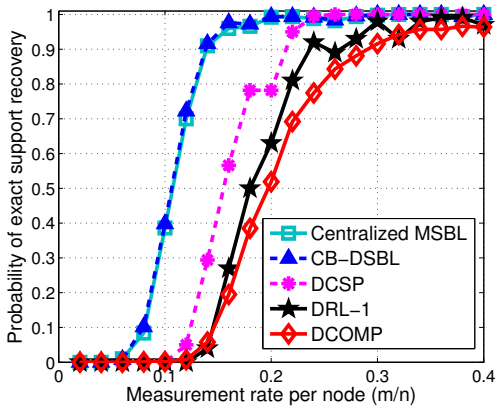


Fig. 9: Exact support recovery probability versus measurement rate. Simulation parameters: $n = 50$, 10% sparsity, SNR = 15 dB and $L = 10$ nodes.

are MSE optimal. CB-DSBL also outperforms DRL-1 and DCOMP in terms of both MSE and support recovery. This is attributed to the fact that the Gaussian prior used in CB-DSBL with its alternate interpretation as a variational approximation to the Student's t-distribution is more capable of inducing sparsity in comparison to the sum-log-sum penalty used in DRL-1. The poor performance of DCOMP is primarily due to its sequential approach towards support recovery which prevents any corrections to be applied to the support estimate at each step of the algorithm. Contrary to [19], DCSP fails to perform better than DCOMP. This is because DCSP works only when the number of measurements exceeds $2k$, where k is the size of the nonzero support.

B. Tradeoff between Measurement Rate and Network Size

In the second set of experiments, we characterize the NMSE phase transition of the different algorithms in the L-(m/n) plane to identify the minimum measurement rate (m/n) needed to ensure less than 1% signal reconstruction error (or, NMSE ≤ -20 dB), for different network sizes (L), and a fixed sparsity rate $(k/n = 0.1)$. As shown in Fig. 8, for the same network size, CB-DSBL is able to successfully recover the unknown signals at a much lower measurement rate compared to DRL-1, DCOMP and DCSP. This plot brings out the significant benefit of using collaboration between nodes and taking advantage of the JSM-2 model in reducing the number of measurements required per node for successful signal recovery. of local measurements (see section IV-E).

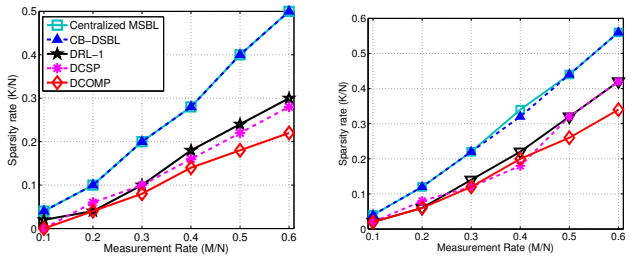
C. Performance versus Measurement Rate (m/n)

In the third set of experiments, we compare the algorithms with respect to their ability to recover the exact support for different undersampling ratios. As seen in Fig. 9, for a similar network size, CB-DSBL is able to exploit the joint sparsity structure better than DCOMP, DCSP and DRL-1, and can correctly recover the support from significantly fewer number of measurements per node. Once again, CB-DSBL has identical support recovery performance as the centralized M-SBL, which was one of our design goals.

D. Phase Transition Characteristics

Here, we compare the phase transition behavior of different algorithms under NMSE and support recovery based pass/fail criteria. Fig. 10a plots the MSE phase transition of different algorithms where any point below the phase transition curve represents a sparsity rate (k/n) and measurement rate (m/n) tuple which results in an NMSE smaller than -20 dB corresponding to smaller than 1 percent signal reconstruction error. Likewise, in Fig. 10b, points below the support recovery phase transition curve represent $(k/n, m/n)$ tuples which result in more than 90 percent accurate nonzero support reconstruction across all the nodes. Again, we see that the CB-DSBL and centralized M-SBL have identical performance and both are capable of signal reconstruction from considerably fewer measurements compared to DRL-1, DCOMP and DCSP.

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.2016.2612120, IEEE Transactions on Signal and Information Processing over Networks



(a) NMSE phase transition (b) Support recovery phase transition

Fig. 10: Phase transition plots for the different joint-sparse signal recovery algorithms. Other simulation parameters: $n = 50$, $L = 5$ nodes, SNR = 30 dB and number of trials = 200.

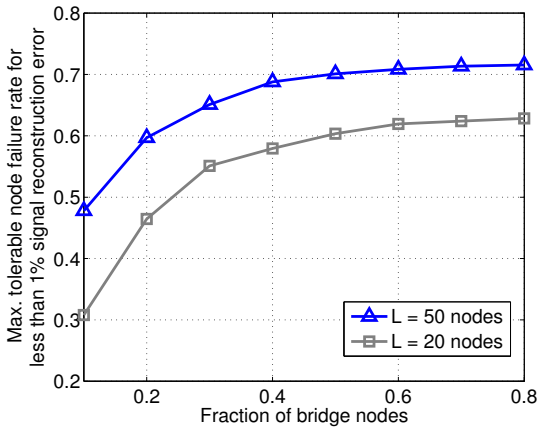


Fig. 11: Plot illustrating the trade off between the density of bridge nodes and the robustness of the proposed CB-DSBL algorithm to random node failures. For a given fraction of bridge nodes (no. of bridge nodes / L), each point on the curve represents the average node failure rate that can be tolerated by CB-DSBL while still achieving less than 1% signal reconstruction error.

E. Robustness to Random Node Failures

Here, we demonstrate empirically that increasing the number of bridge nodes in the CB-DSBL algorithm makes it more robust to random node failures. As shown in Fig. 11, by gradually increasing the density of bridge nodes in the network, the CB-DSBL algorithm is able to tolerate higher rates of node failures without compromising on signal reconstruction performance. More interestingly, only a relatively small fraction of nodes need to be bridge nodes ($< 10\%$ of the total network size) to ensure that CB-DSBL operates robustly in the face of random node failures.

F. Communication Complexity

Lastly, we compare the decentralized algorithms with respect to the total number of messages exchanged between the nodes during the estimation of the unknown vectors. As seen in Fig. 12, the greedy algorithms DCSP and DCOMP are the most communication efficient algorithms as these algorithms have the lowest per iteration communication complexity (see Table I) and run for very few iterations. CB-DSBL and DRL-1 on the other hand have higher communication complexity, with the proposed scheme requiring fewer overall message exchanges compared to DRL-1.

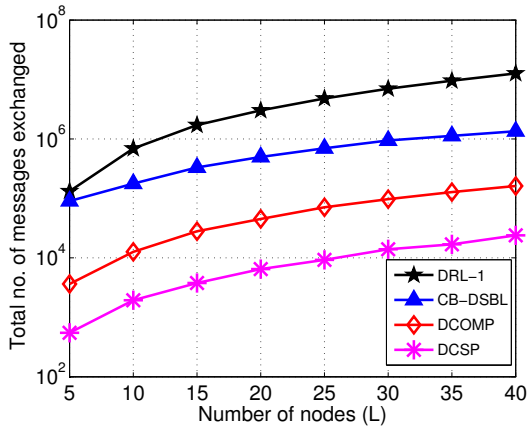


Fig. 12: Average total number of messages exchanged between the nodes for different network sizes. Each message comprises a single real number. Simulation parameters: $n = 50$, $m = 10$, 10% sparsity, SNR = 20 dB.

VI. Distributed Wideband Spectrum Sensing - A Real World Example

To find out how CB-DSBL and other competing distributed JSM-2 algorithms perform in a real world setup, we consider the wideband spectrum sensing problem described below. In wideband spectrum sensing, the goal is to efficiently estimate the occupancy of radio spectrum spanning a wide range of frequencies. Spectrum sensing is a crucial component in the implementation of cognitive radio (CR) networks. In a CR network, the secondary users perform spectrum sensing in order exploit the spectral opportunities which arise due to sparse utilization of the radio spectrum by the primary/licensed users [4], [43]–[46].

A. Compressive Wideband Spectrum Sensing System Model

Consider a CR network comprising P primary users and L secondary users. The total available radio spectrum is partitioned into B frequency sub-bands of equal size. The secondary users sense the spectrum to determine the spectrum holes, i.e., the sub-bands that are not occupied by primary users. Most spectrum sensing techniques consist of two main steps [45], [46]. The first step is to obtain a frequency domain representation of the received wideband signal. This is followed by multiple sub-band level energy detection tests in order to estimate the wideband spectral occupancy of the primary users. Given below is the frequency domain representation of the sampled baseband signal received by the j^{th} secondary user:

$$\mathbf{r}_j^f = \sum_{i=1}^{P} \mathbf{D}_{p,j}^f \mathbf{s}_p^f + \mathbf{w}_j^f \tag{30}$$

where $\mathbf{D}_{p,j}^f = \text{diag}\left(\mathbf{h}_{p,j}^f\right)$ is an $n \times n$ diagonal channel gain matrix representing the wireless channel between primary user p and secondary user j. The diagonal matrix $\mathbf{D}_{p,j}^f$ alludes to the frequency selective nature of channel across the sub-bands, but a flat fading behavior within sub-bands. The $n \times 1$ sized complex vectors: $\mathbf{s}_p^f$, $\mathbf{r}_j^f$ and $\mathbf{w}_j^f$ denote the frequency domain versions of the signal transmitted by the p^{th} primary user,

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.2016.2612120, IEEE Transactions on Signal and Information Processing over Networks

and the signal and noise received at the j^{th} secondary user, respectively. Since the primary users collectively transmit in very few of the B sub-bands, each $\mathbf{s}_p^f, 1 \leq p \leq P$ can be modeled as an approximately sparse vector with most of its coefficients close to zero, and with a few large coefficients coinciding with the sub-band(s) occupied by the primary user p. Consequently, the vector $\mathbf{D}_{p,j}^f \mathbf{s}_p^f$ is also approximately sparse, and so is the summation $\sum_{i=1}^P \mathbf{D}_{p,j}^f \mathbf{s}_p^f$ in (30).

Since the received signal component $\mathbf{x}_j^f = \sum_{i=1}^P \mathbf{D}_{p,j}^f \mathbf{s}_p^f$ in (30) is a wideband signal, acquiring it at Nyquist or higher rate can be prohibitive. Often, due to bandwidth and sampling rate constraints, the secondary users resort to sliding-window, narrow-band processing, i.e., covering a small number of sub-bands at a time, in order to determine the band spectrum occupancy [47]. In a compressive sensing based approach [4], each secondary user implements an *Analog-to-Information-Converter* (AIC), which directly outputs low rate compressive measurements of the received wideband signal. Each secondary user acquires m compressive measurement samples in the form of an $m \times 1$ vector $\mathbf{y}_j^t$ as shown below,

$$\mathbf{y}_j^t = \mathbf{\Phi}_j \mathbf{r}_j^t, \qquad 1 \leq j \leq L, \tag{31}$$

where $\mathbf{r}_j^t$ is the discrete-time representation of the received signal sampled at Nyquist rate, f_S. $\mathbf{\Phi}_j$ is the $m \times n$ sized compressive sampling matrix. Since $m \ll n$, the effective sampling rate $(m/n)f_S$ is significantly lower than the conventional sampling rate f_S. Combining (30) and (31), we can write,

$$\mathbf{y}_j^t \;=\; \mathbf{\Phi}_j \mathbf{F}^H \left(\sum_{i=1}^{P} \mathbf{D}_{p,j}^f \mathbf{s}_p^f \right) + \mathbf{\Phi}_j \mathbf{F}^H \mathbf{w}_j^f \;=\; \mathbf{\Phi}_j \mathbf{F}^H \mathbf{x}_j^f + \tilde{\mathbf{w}}_j^t \tag{32}$$

where $\mathbf{F}$ is the DFT matrix of order n, $\tilde{\mathbf{w}}_j^t$ is the effective measurement noise vector of length m, and $\mathbf{x}_j^f = \sum_{i=1}^P \mathbf{D}_{p,j}^f \mathbf{s}_p^f$ is an $n \times 1$ vector denoting the frequency domain representation of the received signal collectively transmitted by all primary users. Since the secondary users perceive a common spectral occupancy pattern associated with the primary users, the frequency domain vectors $\mathbf{x}_1^f, \mathbf{x}_2^f, \ldots, \mathbf{x}_L^f$ exhibit joint sparsity, and therefore can be recovered efficiently by a JSM-2 signal recovery algorithm.

B. Experimental Setup

For collecting experimental data, *Universal Software Radio Peripheral* (USRP) units (model N-210) were used to realize the primary and the secondary users. Due to the limited bandwidth of the USRP hardware, a scaled down version of the wideband spectrum sensing problem is considered, where the total available frequency band of 1 MHz, centered at 1.1 GHz, is divided into 128 non-overlapping, 7.8125 KHz wide sub-bands. A single USRP unit was configured to mimic five primary users transmitting QPSK modulated RF signal in sub-bands $\{(-59,-49),(-21),(-7),(2,3,4,5,6,7),(21)\}$ (see Fig. 13). A separate USRP unit was configured to collect $n = 256$ samples of down-converted baseband signal, sampled at twice the Nyquist rate. A Random Modulator Pre-Integrator (RMPI) based AIC [48] was simulated in Matlab by taking

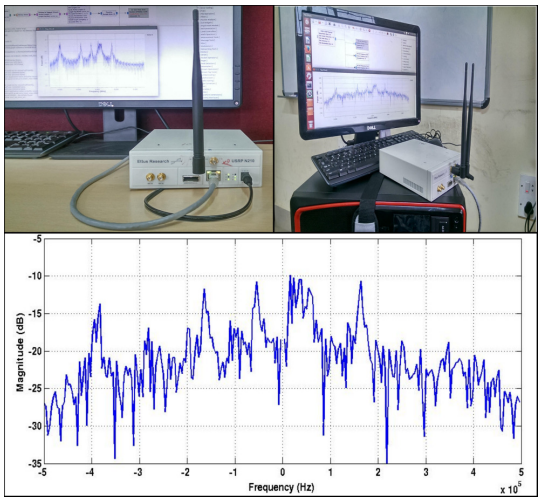


Fig. 13: Top left: a USRP unit configured as a wideband transmitter, transmitting in 11 out of 128 frequency sub-bands. Top right: a mobile USRP station configured to capture the entire wideband signal at Nyquist sampling rate. Bottom: frequency spectrum of the down-converted baseband signal received by one of the secondary users. The five peaks correspond to the five active primary users.

$\mathbf{\Phi}_j$ to be a column normalized Bernoulli ($p = 0.5$) random matrix with ± 1 entries, to generate $m = 32$ compressive measurements according to (31), which were then fed to the recovery algorithms. Multiple recordings were taken at 10 distinct spatial locations, one for each secondary user, in order to capture the effect of both temporal and spatial variations of the wireless channel. The SNR recorded at the CR nodes varied from -2.4dB to 7.8dB. The SNR here is defined as the ratio of total power in signal sub-bands to that in noise sub-bands. All performance metrics are averaged over 100 independent trials.

C. Performance Analysis

We compare the Receiver Operating Characteristics (ROC) of various decentralized JSM-2 recovery algorithms. For MSBL, CB-DSBL, and DRL-1, the ROC plots are obtained by sweeping the threshold value used to identify the occupied frequency bins. For DCOMP and DCSP, the ROC plots are obtained by sweeping the sparsity size k input to these greedy algorithms. To account for spectral leakage, we adopt a pragmatic approach for computing the detection and false alarm probabilities. If a frequency bin is declared as active, it qualifies as a successful detection provided that the detected bin or one of its immediate left or right bins coincides with one of the signal sub-bands. Otherwise, a false alarm is declared. Fig. 14 compares the ROCs of different recovery schemes for measurement compression rate of 12.5%. Due to hard-sparse nature of the output of DCOMP and DCSP algorithms, their false alarm rates never attain the maximal value of one, as reflected in the figure. In spectrum sensing, the goal is to achieve high detection probability while maintaining a low false alarm rate. In this regard, both centralized M-SBL and the proposed CB-DSBL outperform DCOMP, DCSP and DRL-1. In our experiments, both M-SBL and CB-DSBL are able to identify occupied spectrum bins and spectrum holes

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.2016.2612120, IEEE Transactions on Signal and Information Processing over Networks

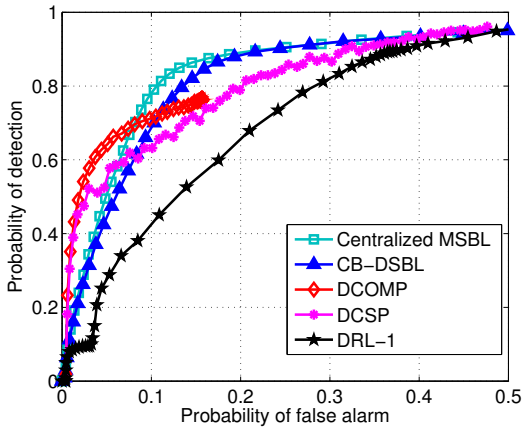


Fig. 14: ROC performance for 12.5% measurement compression ratio, $L = 10$, and SNR ranging from -2.4dB to 7.8dB across the secondary users.

with 90% and 80% accuracy, respectively. In comparison, for DCSP and DRL-1, in order to achieve the same accuracy of detecting 90% of the occupied bins, these algorithms miss 35% of the unoccupied bins. Also, when fed with the correct sparsity level (k), the greedy algorithms DCSP and DCOMP have poor detection performance (less than 60% accuracy). In contrast, owing to their automatic relevance determination property [23], both CB-DSBL and M-SBL are able to correctly infer the sparsity level of the occupied spectrum directly from the measurements, resulting in superior performance.

VII. Conclusions

In this paper, we proposed a novel iterative Bayesian algorithm called CB-DSBL for decentralized estimation of joint-sparse signals by multiple nodes in a network. The CB-DSBL algorithm employs an ADMM based decentralized EM procedure to efficiently learn the parameters of a joint sparsity inducing signal prior which is shared by all the nodes, and is subsequently used in the MAP estimation of the local signals. Experimental results showed that CB-DSBL outperforms existing decentralized algorithms: DRL-1, DCOMP and DCSP, in terms of both NMSE as well as support recovery performance. We also established the R-linear convergence of the underlying decentralized ADMM iterations. The amount of inter-node communication during the ADMM iterations is controlled by restricting each node to exchange information with only a small subset of its single hop neighbors. For this bridge node based communication scheme, the ADMM convergence results presented here are also applicable to any consensus driven optimization of a convex objective function. Future extensions of this work could encompass exploiting any inter vector correlation between the jointly sparse signals. Also, it would be interesting to analyze the convergence of CB-DSBL algorithm in the presence of noisy communication links between nodes and under asynchronous network operation.

Appendix

A. Derivation of the M-step Cost Function

The conditional expectation in (9) can be simplified as:

$$\begin{aligned}&\mathbb{E}_{\mathbf{X}}\left[\log p(\mathbf{Y},\mathbf{X};\boldsymbol{\gamma})|\mathbf{Y};\boldsymbol{\gamma}^k\right]\\&=\mathbb{E}_{[\mathbf{X}|\mathbf{Y};\boldsymbol{\gamma}^k]}[\log p(\mathbf{Y}|\mathbf{X})+\log p(\mathbf{X};\boldsymbol{\gamma})]\\&=\mathbb{E}_{[\mathbf{X}|\mathbf{Y};\boldsymbol{\gamma}^k]}\log p(\mathbf{Y}|\mathbf{X})+\sum_{j\in\mathcal{J}}\mathbb{E}_{[\mathbf{x}_j|\mathbf{y}_j;\boldsymbol{\gamma}^k]}\log p(\mathbf{x}_j;\boldsymbol{\gamma}).\end{aligned}\quad(33)$$

Using (2), and discarding the terms independent of $\boldsymbol{\gamma}$ in (33), the M-step objective function $Q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^k)$ is given by

$$\begin{aligned}Q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^k)&=\sum_{j\in\mathcal{J}}\mathbb{E}_{[\mathbf{x}_j|\mathbf{y}_j,\boldsymbol{\gamma}^k]}\left(-\frac{1}{2}\log|\boldsymbol{\Gamma}|-\frac{1}{2}\mathbf{x}_j^T\boldsymbol{\Gamma}^{-1}\mathbf{x}_j\right)\\&=-\frac{1}{2}\sum_{j\in\mathcal{J}}\left(\log|\boldsymbol{\Gamma}|+\sum_{i=1}^{n}\frac{\mathbb{E}_{[\mathbf{x}_j\sim\mathcal{N}(\boldsymbol{\mu}_j^{k+1},\boldsymbol{\Sigma}_j^{k+1})]}\mathbf{x}_j(i)^2}{\boldsymbol{\gamma}(i)}\right)\\&=-\frac{1}{2}\sum_{j\in\mathcal{J}}\sum_{i=1}^{n}\left(\log\boldsymbol{\gamma}(i)+\frac{\boldsymbol{\Sigma}_j^{k+1}(i,i)+\boldsymbol{\mu}_j^{k+1}(i)^2}{\boldsymbol{\gamma}(i)}\right).\end{aligned}\quad(34)$$

B. Derivation of the Simplified Update for $\boldsymbol{\gamma}_b$

By summing the dual variable update rule (18) across all nodes, the following holds for all $b\in\mathcal{B}$

$$\sum_{j\in\mathcal{N}_b}(\boldsymbol{\lambda}_j^b)^{r+1}=\sum_{j\in\mathcal{N}_b}(\boldsymbol{\lambda}_j^b)^{r}+\rho\sum_{j\in\mathcal{N}_b}\boldsymbol{\gamma}_j^{r+1}-\rho|\mathcal{N}_b|\boldsymbol{\gamma}_b^{r+1}.\quad(35)$$

Plugging (20) in (35), we obtain

$$\sum_{j\in\mathcal{N}_b}(\boldsymbol{\lambda}_j^b)^{r+1}=0\qquad\forall\, b\in\mathcal{B}.\quad(36)$$

Using (36) in (20), we obtain the simplified update for $\boldsymbol{\gamma}_b$.

C. Proof of Theorem 1

The proof of the convergence of ADMM discussed in the sequel is a based on the proof given in [37]. However, our proof differs from the one in [37] due to the different scheme adopted here, which uses the auxiliary/bridge nodes to enforce consensus between the nodes. We make the following assumptions about the objective function f in (23).

1) f is twice differentiable and strongly convex in $\boldsymbol{\gamma}_{\mathcal{J}}$. This implies that there exists $m_f\in\mathbb{R}_+\backslash\{0\}$ such that, for all $\boldsymbol{\gamma}_{\mathcal{J}},\boldsymbol{\gamma}_{\mathcal{J}}^{'}$, the following holds

$$\langle\nabla f(\boldsymbol{\gamma}_{\mathcal{J}})^T-\nabla f(\boldsymbol{\gamma}_{\mathcal{J}}^{'})^T,\boldsymbol{\gamma}_{\mathcal{J}}-\boldsymbol{\gamma}_{\mathcal{J}}^{'}\rangle\;\geq\;m_f||\boldsymbol{\gamma}_{\mathcal{J}}-\boldsymbol{\gamma}_{\mathcal{J}}^{'}||_2^2.\quad(37)$$

2) ∇f is Lipschitz continuous, i.e., there exists a positive scalar M_f such that, for all $\boldsymbol{\gamma}_{\mathcal{J}},\boldsymbol{\gamma}_{\mathcal{J}}^{'}$, we have

$$||\nabla f(\boldsymbol{\gamma}_{\mathcal{J}})-\nabla f(\boldsymbol{\gamma}_{\mathcal{J}}^{'})||_2\leq M_f||\boldsymbol{\gamma}_{\mathcal{J}}-\boldsymbol{\gamma}_{\mathcal{J}}^{'}||_2.\quad(38)$$

Let r denote the ADMM iteration count. From the zero subgradient optimality conditions corresponding to (16) and (17), we have

$$\nabla f(\boldsymbol{\gamma}_{\mathcal{J}}^{r+1})^T+\mathbf{E}_1^T\boldsymbol{\lambda}^r+\rho\mathbf{E}_1^T\mathbf{E}_1\boldsymbol{\gamma}_{\mathcal{J}}^{r+1}+\rho\mathbf{E}_1^T\mathbf{E}_2\boldsymbol{\gamma}_{\mathcal{B}}^{r}=0\quad(39)$$

$$\mathbf{E}_2^T\boldsymbol{\lambda}^r+\rho\mathbf{E}_2^T\mathbf{E}_2\boldsymbol{\gamma}_{\mathcal{B}}^{r+1}+\rho\mathbf{E}_2^T\mathbf{E}_1\boldsymbol{\gamma}_{\mathcal{J}}^{r+1}=0.\quad(40)$$

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.2016.2612120, IEEE Transactions on Signal and Information Processing over Networks

From the dual variable update equation, we have,

$$\boldsymbol{\lambda}^{r+1} = \boldsymbol{\lambda}^r + \rho(\mathbf{E}_1\boldsymbol{\gamma}_{\mathcal{J}}^{r+1} + \mathbf{E}_2\boldsymbol{\gamma}_{\mathcal{B}}^{r+1}). \quad (41)$$

Premultiplying (41) with $\mathbf{E}_1^T$ and $\mathbf{E}_2^T$ followed by its summation to (39) and (40) respectively gives

$$\nabla f(\boldsymbol{\gamma}_{\mathcal{J}}^{r+1})^T + \mathbf{E}_1^T\boldsymbol{\lambda}^{r+1} + \rho\mathbf{E}_1^T\mathbf{E}_2(\boldsymbol{\gamma}_{\mathcal{B}}^{r} - \boldsymbol{\gamma}_{\mathcal{B}}^{r+1}) = 0. \quad (42)$$

$$\mathbf{E}_2^T\boldsymbol{\lambda}^{r+1} = 0. \quad (43)$$

By initializing $\boldsymbol{\lambda}$ equal to zero, $\boldsymbol{\lambda}^r$ always lies in the nullspace $\mathcal{N}(\mathbf{E}_2^T)$, physically implying that the sum of the Lagrange multipliers of nodes connected to a given bridge node is always equal to zero. Let $\boldsymbol{\gamma}_{\mathcal{J}}^r \to \boldsymbol{\gamma}_{\mathcal{J}}^*$, $\boldsymbol{\gamma}_{\mathcal{B}}^r \to \boldsymbol{\gamma}_{\mathcal{B}}^*$ and $\boldsymbol{\lambda}^r \to \boldsymbol{\lambda}^*$ as $r \to \infty$, then putting $r \to \infty$ in (41), (42) and (43) gives

$$\nabla f(\boldsymbol{\gamma}_{\mathcal{J}}^*)^T + \mathbf{E}_1^T\boldsymbol{\lambda}^* = 0 \quad (44)$$

$$\mathbf{E}_2^T\boldsymbol{\lambda}^* = 0 \quad (45)$$

$$\mathbf{E}_1\boldsymbol{\gamma}_{\mathcal{J}}^* + \mathbf{E}_2\boldsymbol{\gamma}_{\mathcal{B}}^* = 0. \quad (46)$$

Note that the condition (46) implies consensus among $\boldsymbol{\gamma}_j, j \in \mathcal{J}$, upon convergence. By subtracting (44), (45) and (46) from (42), (43) and (41), respectively, we get the desired difference terms needed for showing convergence results.

$$\nabla f(\boldsymbol{\gamma}_{\mathcal{J}}^{r+1})^T - \nabla f(\boldsymbol{\gamma}_{\mathcal{J}}^*)^T + \mathbf{E}_1^T(\boldsymbol{\lambda}^{r+1} - \boldsymbol{\lambda}^*) + \rho\mathbf{E}_1^T\mathbf{E}_2(\boldsymbol{\gamma}_{\mathcal{B}}^r - \boldsymbol{\gamma}_{\mathcal{B}}^{r+1}) = 0 \quad (47)$$

$$\mathbf{E}_2^T(\boldsymbol{\lambda}^{r+1} - \boldsymbol{\lambda}^*) = 0 \quad (48)$$

$$\boldsymbol{\lambda}^{r+1} - \boldsymbol{\lambda}^r = \rho\mathbf{E}_1(\boldsymbol{\gamma}_{\mathcal{J}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{J}}^*) + \rho\mathbf{E}_2(\boldsymbol{\gamma}_{\mathcal{B}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{B}}^*). \quad (49)$$

Premultiplying (49) with $\mathbf{E}_2^T$ and using (43), we obtain,

$$\mathbf{E}_2^T\mathbf{E}_1(\boldsymbol{\gamma}_{\mathcal{J}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{J}}^*) = -\mathbf{E}_2^T\mathbf{E}_2(\boldsymbol{\gamma}_{\mathcal{B}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{B}}^*). \quad (50)$$

From the strong convexity of f and using (47), we can write,

$$\begin{aligned}
m_f||\boldsymbol{\gamma}_{\mathcal{J}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{J}}^*||_2^2 &\le \langle \mathbf{E}_1^T(\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^{r+1}), \boldsymbol{\gamma}_{\mathcal{J}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{J}}^*\rangle \\
&+ \rho\langle \mathbf{E}_1^T\mathbf{E}_2(\boldsymbol{\gamma}_{\mathcal{B}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{B}}^r), (\boldsymbol{\gamma}_{\mathcal{J}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{J}}^*)\rangle \\
&= \langle (\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^{r+1}), \mathbf{E}_1(\boldsymbol{\gamma}_{\mathcal{J}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{J}}^*)\rangle \\
&+ \rho\langle (\boldsymbol{\gamma}_{\mathcal{B}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{B}}^r), \mathbf{E}_2^T\mathbf{E}_1(\boldsymbol{\gamma}_{\mathcal{J}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{J}}^*)\rangle \\
&= \langle (\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^{r+1}), \mathbf{E}_1(\boldsymbol{\gamma}_{\mathcal{J}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{J}}^*)\rangle \\
&- \rho\langle (\boldsymbol{\gamma}_{\mathcal{B}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{B}}^r), \mathbf{E}_2^T\mathbf{E}_2(\boldsymbol{\gamma}_{\mathcal{B}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{B}}^*)\rangle \\
&= \langle (\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^{r+1}), \frac{1}{\rho}(\boldsymbol{\lambda}^{r+1} - \boldsymbol{\lambda}^r) - \mathbf{E}_2(\boldsymbol{\gamma}_{\mathcal{B}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{B}}^*)\rangle \\
&- \rho\langle (\boldsymbol{\gamma}_{\mathcal{B}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{B}}^r), \mathbf{E}_2^T\mathbf{E}_2(\boldsymbol{\gamma}_{\mathcal{B}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{B}}^*)\rangle \\
&= \frac{1}{\rho}\langle (\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^{r+1}), (\boldsymbol{\lambda}^{r+1} - \boldsymbol{\lambda}^r)\rangle \\
&+ \rho\langle \mathbf{E}_2(\boldsymbol{\gamma}_{\mathcal{B}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{B}}^r), \mathbf{E}_2(\boldsymbol{\gamma}_{\mathcal{B}}^* - \boldsymbol{\gamma}_{\mathcal{B}}^{r+1})\rangle. \quad (51)
\end{aligned}$$

Here, the first identity is obtained by using a property of the inner product. The second, third and fourth identities are obtained by using (50), (49) and (48) respectively. By defining $\mathbf{u} = [(\mathbf{E}_2\boldsymbol{\gamma}_{\mathcal{B}})^T \mid \boldsymbol{\lambda}^T]^T$, the RHS in (51) can be expressed as a matrix norm $||\mathbf{u}^r - \mathbf{u}^{r+1}||_{\mathbf{G}}^2 = (\mathbf{u}^r - \mathbf{u}^{r+1})^T\mathbf{G}(\mathbf{u}^{r+1} - \mathbf{u}^*)$, where $\mathbf{G}$ is given by

$$\mathbf{G} = \begin{bmatrix} \rho I_{n|B|} & 0 \\ 0 & \frac{1}{\rho}I_{N_C} \end{bmatrix}.$$

Using the identity:

$$2(\mathbf{u}^r - \mathbf{u}^{r+1})^T\mathbf{G}(\mathbf{u}^{r+1} - \mathbf{u}^*) = ||\mathbf{u}^r - \mathbf{u}^*||_{\mathbf{G}}^2 - ||\mathbf{u}^{r+1} - \mathbf{u}^*||_{\mathbf{G}}^2 - ||\mathbf{u}^r - \mathbf{u}^{r+1}||_{\mathbf{G}}^2, \quad (52)$$

the inequality in (51) can be rewritten as

$$m_f||\boldsymbol{\gamma}_{\mathcal{J}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{J}}^*||_2^2 \le \frac{1}{2}\left(||\mathbf{u}^r - \mathbf{u}^*||_{\mathbf{G}}^2 - ||\mathbf{u}^{r+1} - \mathbf{u}^*||_{\mathbf{G}}^2 - ||\mathbf{u}^r - \mathbf{u}^{r+1}||_{\mathbf{G}}^2\right) \quad (53)$$

By discarding the non-positive terms in the LHS of (53), we obtain the following upper bound on the primal optimality gap.

$$||\boldsymbol{\gamma}_{\mathcal{J}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{J}}^*||_2^2 \;\le\; \frac{1}{2m_f}||\mathbf{u}^r - \mathbf{u}^*||_{\mathbf{G}}^2. \quad (54)$$

In Appendix D, we prove the monotonic convergence of $\mathbf{u}^r$ to $\mathbf{u}^*$. Thus, from the monotonic decay of the RHS in (54), we have R-linear convergence of $\boldsymbol{\gamma}_{\mathcal{J}}^r$ to $\boldsymbol{\gamma}_{\mathcal{J}}^*$.

D. Proof of monotonic convergence of $\mathbf{u}^r$ *to* $\mathbf{u}^*$

In order to prove monotonic convergence of $\mathbf{u}^r$ to $\mathbf{u}^*$, it is sufficient to show that there exists a $\delta > 0$ such that

$$||\mathbf{u}^{r+1} - \mathbf{u}^*||_{\mathbf{G}}^2 \;\le\; \frac{1}{1+\delta}||\mathbf{u}^r - \mathbf{u}^*||_{\mathbf{G}}^2. \quad (55)$$

By rearranging the terms in (53), we have

$$||\mathbf{u}^{r+1} - \mathbf{u}^*||_{\mathbf{G}}^2 \;\le\; ||\mathbf{u}^r - \mathbf{u}^*||_{\mathbf{G}}^2 - ||\mathbf{u}^{r+1} - \mathbf{u}^r||_{\mathbf{G}}^2 - 2m_f||\boldsymbol{\gamma}_{\mathcal{J}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{J}}^*||_2^2. \quad (56)$$

By comparing terms in (55) and (56), we observe that if

$$2m_f||\boldsymbol{\gamma}_{\mathcal{J}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{J}}^*||_2^2 + ||\mathbf{u}^{r+1} - \mathbf{u}^r||_{\mathbf{G}}^2 \ge \delta||\mathbf{u}^{r+1} - \mathbf{u}^*||_{\mathbf{G}}^2, \quad (57)$$

or equivalently,

$$\begin{aligned}
2m_f||\boldsymbol{\gamma}_{\mathcal{J}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{J}}^*||_2^2 + \rho||\mathbf{E}_2(\boldsymbol{\gamma}_{\mathcal{B}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{B}}^r)||_2^2 + \frac{1}{\rho}||\boldsymbol{\lambda}^{r+1} - \boldsymbol{\lambda}^r||_2^2 \\
\ge \delta\left(\rho||\mathbf{E}_2(\boldsymbol{\gamma}_{\mathcal{B}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{B}}^*)||_2^2 + \frac{1}{\rho}||\boldsymbol{\lambda}^{r+1} - \boldsymbol{\lambda}^*||_2^2\right), \quad (58)
\end{aligned}$$

holds, then $\mathbf{u}^r$ converges monotonically to $\mathbf{u}^*$. We now proceed to derive upper bounds for $||\mathbf{E}_2(\boldsymbol{\gamma}_{\mathcal{B}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{B}}^*)||_2$ and $||\boldsymbol{\lambda}^{r+1} - \boldsymbol{\lambda}^*||_2$ in terms of the LHS. These upper bounds will be used in the sequel to establish the inequality in (58).

- *An upper bound for* $\rho||\mathbf{E}_2(\boldsymbol{\gamma}_{\mathcal{B}}^{k+1} - \boldsymbol{\gamma}_{\mathcal{B}}^*)||_2$
  Note that for any two vectors $\mathbf{a}$, $\mathbf{b}$ and a scalar $\mu > 1$

  $$||\mathbf{a} + \mathbf{b}||_2^2 \;\ge\; (1-\mu)||\mathbf{a}||_2^2 + \left(1 - \frac{1}{\mu}\right)||\mathbf{b}||_2^2. \quad (59)$$

  Applying inequality (59) to (49), we get the following upper bound.

  $$\rho||\mathbf{E}_2(\boldsymbol{\gamma}_{\mathcal{B}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{B}}^*)||_2^2 \le \left(\frac{\mu}{\mu - 1}\right)\frac{1}{\rho}||\boldsymbol{\lambda}^{r+1} - \boldsymbol{\lambda}^r||_2^2 + \left(\mu\rho\sigma_{\max}^2(\mathbf{E}_1)\right)||\boldsymbol{\gamma}_{\mathcal{J}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{J}}^*||_2^2. \quad (60)$$

  Here, $\sigma_{\max}(\mathbf{E}_1)$ is the largest singular value of $\mathbf{E}_1$.

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.2016.2612120, IEEE Transactions on Signal and Information Processing over Networks

- *An upper bound for* $\frac{1}{\rho}||\boldsymbol{\lambda}^{r+1}-\boldsymbol{\lambda}^*||_2$
  Similar application of inequality (59) to (47) results in an upper bound for $\frac{1}{\rho}||\boldsymbol{\lambda}^{r+1}-\boldsymbol{\lambda}^*||_2$ as shown below.

$$\begin{aligned}||\mathbf{E}_1^T(\boldsymbol{\lambda}^{r+1}-\boldsymbol{\lambda}^*)||_2^2 \leq \frac{\nu}{(\nu-1)}||\nabla f(\boldsymbol{\gamma}_{\mathcal{J}}^{r+1})^T-\nabla f(\boldsymbol{\gamma}_{\mathcal{J}}^*)^T||_2^2 \\ +\ \nu||\rho\mathbf{E}_1^T\mathbf{E}_2(\boldsymbol{\gamma}_{\mathcal{B}}^r-\boldsymbol{\gamma}_{\mathcal{B}}^{r+1})||_2^2 \\ \implies \frac{1}{\rho}||\boldsymbol{\lambda}^{r+1}-\boldsymbol{\lambda}^*||_2^2 \leq \\ \frac{\nu}{\rho(\nu-1)\sigma_{\min}^2(\mathbf{E}_1)}||\nabla f(\boldsymbol{\gamma}_{\mathcal{J}}^{r+1})^T-\nabla f(\boldsymbol{\gamma}_{\mathcal{J}}^*)^T||_2^2 \\ +\ \frac{\nu\rho\sigma_{\max}^2(\mathbf{E}_1)}{\sigma_{\min}^2(\mathbf{E}_1)}||\mathbf{E}_2(\boldsymbol{\gamma}_{\mathcal{B}}^r-\boldsymbol{\gamma}_{\mathcal{B}}^{r+1})||_2^2. \qquad (61)\end{aligned}$$

From the Lipschitz continuity of ∇f (38), we obtain the following modified upper bound.

$$\begin{aligned}\frac{1}{\rho}||\boldsymbol{\lambda}^{r+1}-\boldsymbol{\lambda}^*||_2^2 \leq \frac{\nu M_f^2}{\rho(\nu-1)\sigma_{\min}^2(\mathbf{E}_1)}||\boldsymbol{\gamma}_{\mathcal{J}}^{r+1}-\boldsymbol{\gamma}_{\mathcal{J}}^*||_2^2 \\ +\frac{\nu\rho\sigma_{\max}^2(\mathbf{E}_1)}{\sigma_{\min}^2(\mathbf{E}_1)}||\mathbf{E}_2(\boldsymbol{\gamma}_{\mathcal{B}}^r-\boldsymbol{\gamma}_{\mathcal{B}}^{r+1})||_2^2. \qquad (62)\end{aligned}$$

Here, $\sigma_{\min}(\mathbf{E}_1)$ denotes the smallest singular value of $\mathbf{E}_1$ and ν is a positive scalar greater than unity.

By summing the upper bounds in (60) and (62), we get

$$\begin{aligned}\rho||\mathbf{E}_2(\boldsymbol{\gamma}_{\mathcal{B}}^{r+1}-\boldsymbol{\gamma}_{\mathcal{B}}^*)||_2^2+\frac{1}{\rho}||\boldsymbol{\lambda}^{r+1}-\boldsymbol{\lambda}^*||_2^2 \leq \\ \frac{1}{\delta}\Big(2m_f||\boldsymbol{\gamma}_{\mathcal{J}}^{r+1}-\boldsymbol{\gamma}_{\mathcal{J}}^*||_2^2+\rho||\mathbf{E}_2(\boldsymbol{\gamma}_{\mathcal{B}}^r-\boldsymbol{\gamma}_{\mathcal{B}}^{r+1})||_2^2 \\ +\frac{1}{\rho}||\boldsymbol{\lambda}^{r+1}-\boldsymbol{\lambda}^r||_2^2\Big) \qquad (63)\end{aligned}$$

where

$$\delta \triangleq \left(\max_{\mu,\nu\geq 1}\left(\frac{\frac{\nu M_f^2}{\rho(\nu-1)\sigma_{\min}^2(\mathbf{E}_1)}+\mu\rho\sigma_{\max}^2(\mathbf{E}_1)}{2m_f},\nu\kappa,\frac{\mu}{\mu-1}\right)\right)^{-1}. \qquad (64)$$

Thus, for δ as defined above, the inequality (58) holds and consequently the inequality (55) also holds, thereby establishing the Q-linear convergence of the sequence $\mathbf{u}^k$ to $\mathbf{u}^*$.

E. Proof of Theorem 2

Let δ_{opt} denote the maximum value of δ for any $\rho>0$. Then, we can write

$$\begin{aligned}\delta_{\text{opt}} &= \max_{\rho>0}\left(\max_{\mu,\nu\geq 1}\left(\min\left(f_1(\mu,\nu,\rho),f_2(\nu),f_3(\mu)\right)\right)\right) \\ &= \max_{\mu,\nu\geq 1}\left(\max_{\rho>0}\left(\min\left(f_1(\mu,\nu,\rho),f_2(\nu),f_3(\mu)\right)\right)\right) \qquad (65)\end{aligned}$$

where the scalar functions f_1, f_2 and f_3 represent the three terms inside the minimum operator in (26). The following two Lemmas summarize the optimization of δ in (65).

Lemma 2. *$\delta_{opt}=\max\limits_{\mu,\nu\geq 1}\left\{min\left(\bar{f}_1(\mu,\nu),f_2(\nu),f_3(\mu)\right)\right\}$ where, $\bar{f}_1(\mu,\nu)\triangleq\max\limits_{\rho>0} f_1(\mu,\nu,\rho)$.*

Proof. See Appendix F. □

Lemma 3. *There exists a unique $(\mu,\nu)=(\mu^*,\nu^*)$ which simultaneously satisfies: (i) $\bar{f}_1=f_2=f_3$, and (ii) $\mu\geq 1,\nu\geq 1$. Further, such a (μ^*,ν^*) maximizes $g(\mu,\nu)=min\left(\bar{f}_1(\mu,\nu),f_2(\nu),f_3(\mu)\right)$ over $\mu,\nu\geq 1$.*

Proof. See Appendix G. □

The scalar function f_1 in Lemma 2 is maximized at $\rho=\frac{M_f}{\sigma_{\max}\sigma_{\min}}\sqrt{\frac{\nu}{\mu(\nu-1)}}$ to give $\bar{f}_1=\frac{M_f}{\sigma_{\min}\sigma_{\max}}\sqrt{\frac{\nu}{\mu(\nu-1)}}$. Further, by solving for the unique tuple (μ^*,ν^*) which satisfies the two optimality conditions specified in Lemma 3, the optimal augmented Lagrangian parameter ρ and corresponding optimal δ can be shown to be equal to the ρ_{opt} and δ_{opt} as defined in Theorem 2.

F. Proof of Lemma 2

Let $\bar{\rho}\triangleq\arg\max\limits_{\rho>0} f_1$. Then, by restricting the feasible set in (65), we have,

$$\begin{aligned}\delta_{\text{opt}} &\geq \max_{\mu,\nu\geq 1}\left[\max_{\rho=\rho_{\mu,\nu}}\left\{\min\left(f_1(\mu,\nu,\rho),f_2(\nu),f_3(\mu)\right)\right\}\right] \\ &= \max_{\mu,\nu\geq 1}\left\{\min\left(\tilde{f}_1(\mu,\nu),f_2(\nu),f_3(\mu)\right)\right\}. \qquad (66)\end{aligned}$$

On the other hand, from (65) and using $\tilde{f}_1\geq f_1$, we have,

$$\begin{aligned}\delta_{\text{opt}} &= \max_{\mu,\nu\geq 1}\left[\max_{\rho>0}\left\{\min\left(f_1(\mu,\nu,\rho),f_2(\nu),f_3(\mu)\right)\right\}\right] \\ &\leq \max_{\mu,\nu\geq 1}\left\{\min\left(\tilde{f}_1(\mu,\nu),f_2(\nu),f_3(\mu)\right)\right\}. \qquad (67)\end{aligned}$$

Combining (66) and (67) establishes Lemma 2.

G. Proof of Lemma 3

In order to prove the Lemma, we claim the following.

a) For any $\epsilon>0$, there exist positive constants B_μ and B_ν such that $g(\mu,\nu)\leq\epsilon$ when either $\mu\geq B_\mu$ or $\nu\geq B_\nu$ holds.
b) Any point (μ,ν) which satisfies condition 2 but does not satisfy condition 1 cannot be a local maximum of g.

Note that claim (a) holds trivially for $B_\mu=\frac{m_f^2}{\kappa\mathcal{M}_f^2\epsilon^2}$ and $B_\nu=\frac{1}{\kappa\epsilon}$. In order to verify claim (b), let us consider a point (μ_0,ν_0) which satisfies condition 2, but not condition 1. Then, we need to consider three cases.

- *Case-I:* $\tilde{f}_1$, f_2 and f_3 are distinct at (μ_0,ν_0). Without loss of generality (WLOG), let $g=\tilde{f}_1$ at (μ_0,ν_0). Then, from the continuity of $\tilde{f}_1,f_2,f_3$, there exists an ϵ (>0) ball B_ϵ, centered at (μ_0,ν_0) and with radius ϵ inside which $g=\tilde{f}_1$ holds. Since, inside B_ϵ, g is strictly monotonic with respect to μ and ν, there exists $(\mu,\nu)\in\mathcal{B}_\epsilon$ such that $g(\mu,\nu)>g(\mu_0,\nu_0)$. Hence, (μ_0,ν_0) is not a local maximum.
- *Case-II:* At (μ_0,ν_0), any two of $\tilde{f}_1$, f_2 and f_3 are equal and strictly greater than the remaining one. The same arguments as Case-I apply here as well.
- *Case-III:* At (μ_0,ν_0), any two of $\tilde{f}_1$, f_2 and f_3 are equal and strictly less than the remaining one. WLOG, let $\tilde{f}_1=f_2<f_3$. Let $\mathcal{C}(\mu,\nu)$ denote the continuous

curve in (μ, ν) plane whose each point satisfies $\tilde{f}_1 = f_2$. Clearly, (μ_0, ν_0) also lies on the curve $\mathcal{C}$. Moreover, there are an uncountably infinite number of points of $\mathcal{C}$ inside B_ϵ, with B_ϵ defined as in Case-I. Due to the monotonicity of g along $\mathcal{C}$, there exists $(\mu, \nu) \in \mathcal{B}_\epsilon$ such that $g(\mu, \nu) > g(\mu_0, \nu_0)$. Hence, (μ_0, ν_0) is not a local maximum.

From claim (a) and the fact that at the boundary points ($\mu = 1$ or $\nu = 1$), the objective g evaluates to zero, we may restrict our search for the global maximizer of g to the set $\mathcal{D} = \{(\mu, \nu) \mid 1 \leq \mu \leq B_\mu, 1 \leq \nu \leq B_\nu\}$. Then, from claim (b), uniqueness of $(\mu^*, \nu^*) \in D$ and Weierstrass theorem, it follows that (μ^*, ν^*) is indeed the unique global maximizer of the continuous function g. Thus, the proof is complete.

REFERENCES

[1] S. Khanna and C. R. Murthy, "Decentralized Bayesian learning of jointly sparse signals," in *Proc. GLOBECOM*, Dec 2014, pp. 3103–3108.

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

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

[4] Z. Fanzi, 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, 2011.

[5] 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, 2013.

[6] N. H. Nguyen, N. M. Nasrabadi, and T. D. Tran, "Robust multi-sensor classification via joint sparse representation," in *Proc. 14th Int. Conf. Inform. Fusion*, July 2011, pp. 1–8.

[7] S. Ji, D. Dunson, and L. Carin, "Multitask compressive sensing," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 92–106, Jan 2009.

[8] R. Prasad, C. R. Murthy, and B. D. Rao, "Joint channel estimation and data detection in MIMO-OFDM systems: A sparse Bayesian learning approach," *IEEE Trans. Signal Process.*, vol. 63, no. 20, pp. 5369–5382, Oct 2015.

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

[10] Y. Barbotin, A. Hormati, S. Rangan, and M. Vetterli, "Estimation of sparse MIMO channels with common support," *IEEE Trans. Commun.*, vol. 60, no. 12, pp. 3705–3716, 2012.

[11] S. F. Cotter, B. D. 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, 2005.

[12] D. Baron, M. F. Duarte, M. B. Wakin, S. Sarvotham, and R. G. Baraniuk, "Distributed compressive sensing," *CoRR*, vol. abs/0901.3403, 2009. [Online]. Available: http://arxiv.org/abs/0901.3403

[13] H. Lu, X. Long, and J. Lv, "A fast algorithm for recovery of jointly sparse vectors based on the alternating direction methods," *J. Mach. Learn. Res*, pp. 461–469, 2011.

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

[15] ——, "Sparse bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2153–2164, 2004.

[16] J. Ziniel and P. Schniter, "Efficient high-dimensional inference in the multiple measurement vector problem," *IEEE Trans. Signal Process.*, vol. 61, no. 2, pp. 340–354, Jan 2013.

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

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

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

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

[21] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, "Optimization with sparsity-inducing penalties," *Foundations and Trends in Machine Learning*, vol. 4, no. 1, pp. 1–106, Jan 2012.

[22] D. J. MacKay, "Bayesian interpolation," *Neural Computation*, vol. 4, pp. 415–447, 1991.

[23] D. P. Wipf and S. S. Nagarajan, "A new view of automatic relevance determination," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2007, pp. 1625–1632.

[24] D. Yang, H. Li, and G. D. Peterson, "Space-time turbo bayesian compressed sensing for UWB systems," in *Proc. ICC*, May 2010, pp. 1–6.

[25] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc WSNs with noisy links; part I: Distributed estimation of deterministic signals," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 350–364, 2008.

[26] J. Palmer, B. D. Rao, and D. P. Wipf, "Perspectives on sparse Bayesian learning," in *Advances in Neural Information Processing Systems*, 2004, pp. 249–256.

[27] R. Neal and G. E. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*. Kluwer Academic Publishers, 1998, pp. 355–368.

[28] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.

[29] 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*, vol. 39, no. 1, pp. 1–38, 1977.

[30] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed expectation-maximization algorithm for density estimation and classification using wireless sensor networks," in *Proc. ICASSP*, Mar 2008, pp. 1989–1992.

[31] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[32] R. Zhang and J. Kwok, "Asynchronous distributed ADMM for consensus optimization," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. JMLR Workshop and Conference Proceedings, 2014, pp. 1701–1709.

[33] J. Matamoros, S. M. Fosson, E. Magli, and C. Anton-Haro, "Distributed ADMM for in-network reconstruction of sparse signals with innovations," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 4, pp. 225–234, Dec 2015.

[34] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Puschel, "D-ADMM: A communication-efficient distributed algorithm for separable optimization," *IEEE Trans. Signal Process.*, vol. 61, no. 10, pp. 2718–2723, 2013.

[35] T. Erseghe, "A distributed and maximum-likelihood sensor network localization algorithm based upon a nonconvex problem formulation," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 4, pp. 247–258, Dec 2015.

[36] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," *Rice University CAAM Technical Report TR12-14*, 2012.

[37] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1750–1761, Apr. 2014.

[38] T. Erseghe, D. Zennaro, E. Dall'Anese, and L. Vangelista, "Fast consensus by the alternating direction multipliers method," *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5523–5537, Nov 2011.

[39] H. Zhu, G. B. Giannakis, and A. Cano, "Distributed in-network channel decoding," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 3970–3983, Oct 2009.

[40] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[41] D. Jakovetic, J. Xavier, and J. M. F. Moura, "Fast distributed gradient methods," *IEEE Trans. Automat. Control*, vol. 59, no. 5, pp. 1131–1146, May 2014.

[42] A. I. Chen and A. Ozdaglar, "A fast distributed proximal-gradient method," in *Proc. Allerton Conf. on Commun., Control and Comput.*, Oct 2012, pp. 601–608.

[43] Y. Wang, A. Pandharipande, Y. L. Polo, and G. Leus, "Distributed compressive wide-band spectrum sensing," in *Information Theory and Applications Workshop, 2009*, Feb 2009, pp. 178–183.

[44] J. A. Bazerque and G. B. Giannakis, "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1847–1862, 2010.
[45] Z. Quan, S. Cui, A. H. Sayed, and H. V. Poor, "Wideband spectrum sensing in cognitive radio networks," in *Proc. ICC*, May 2008, pp. 901–906.
[46] H. Sun, A. Nallanathan, C. X. Wang, and Y. Chen, "Wideband spectrum sensing for cognitive radio networks: A survey," *IEEE Wireless Communications*, vol. 20, no. 2, pp. 74–81, Apr 2013.
[47] A. Sharma and C. R. Murthy, "Group testing-based spectrum hole search for cognitive radios," *IEEE Trans. Veh. Technol.*, vol. 63, no. 8, pp. 3794–3805, Oct 2014.
[48] J. Yoo, S. Becker, M. Monge, M. Loh, E. Candès, and A. Emami-Neyestanak, "Design and implementation of a fully integrated compressed-sensing signal acquisition system," in *Proc. ICASSP*, Mar 2012, pp. 5325–5328.

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 2014-16. 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.