LIGHT-WEIGHT SEQUENTIAL SBL ALGORITHM: AN ALTERNATIVE TO OMP

Rohan R. Pote and Bhaskar D. Rao

Department of Electrical and Computer Engineering
University of California San Diego

ABSTRACT

We present a Light-Weight Sequential Sparse Bayesian Learning (LWS-SBL) algorithm as an alternative to the orthogonal matching pursuit (OMP) algorithm for the general sparse signal recovery problem. The proposed approach formulates the recovery problem under the Type-II estimation framework and the stochastic maximum likelihood objective. We compare the computational complexity for the proposed algorithm with OMP and highlight the main differences. For the case of parametric dictionaries, a gridless version is developed by extending the proposed sequential SBL algorithm to locally optimize grid points near potential source locations and it is empirically shown that the performance approaches Cramér-Rao bound. Numerical results using the proposed approach demonstrate the support recovery performance improvements in different scenarios at a small computational price when compared to the OMP algorithm.

Index Terms— Sparse signal recovery, compressed sensing, sparse Bayesian learning, orthogonal matching pursuit, computational complexity, gridless estimation

1. INTRODUCTION

Sparse signal recovery (SSR) has witnessed numerous applications over the past several decades [1–6]. Consequently, many algorithms have been proposed that offer favourable tradeoffs between recovery performance, speed, and storage [7–14]. The underlying problem in SSR can be stated as follows: Given a measurement matrix $\mathbf{\Phi} \in \mathbb{C}^{m\times n}(m < n)$ and measurement vector $\mathbf{y} \in \mathbb{C}^m$ such that

$$\mathbf{y} = \mathbf{\Phi x} + \mathbf{n}, \tag{1}$$

may be corrupted by noise vector $\mathbf{n} \in \mathbb{C}^m$, the goal is to recover the vector $\mathbf{x} \in \mathbb{C}^n$ which is known to be sparse i.e., $\|\mathbf{x}\|_0 \ll n$. The noise, $\mathbf{n}$, is distributed as $\mathcal{CN}(\mathbf{0}, \sigma_n^2\mathbf{I})$. $\mathbf{x}$ and $\mathbf{n}$ are independent of each other. On the computationally favourable paradigm, greedy algorithms (e.g. pursuit algorithms like matching pursuit (MP) [7], orthogonal MP (OMP) [15–18], compressive sampling MP (CoSaMP) [19] etc.) are well-studied algorithms that offer faster recovery at the cost of slight degradation in performance. OMP is a widely employed [20, 21] iterative technique in this category that recovers $\mathbf{x}$ *deterministically* and improves over MP by ensuring that the residual error is orthogonal to already selected columns of $\mathbf{\Phi}$.

In this work, we offer a *stochastic* alternative to the OMP algorithm with a similar complexity while improving support recovery performance. The following contributions are identified:

- A Light-Weight Sequential Sparse Bayesian Learning (LWS-SBL) algorithm is proposed along with recursive update strategies based on the Type-II estimation framework and stochastic maximum likelihood objective.
- The computational complexity of LWS-SBL is analyzed and compared to that for the OMP algorithm. Computationally efficient steps are derived that help to develop a low complexity algorithm that can compete favorably with the widely used OMP algorithm.
- For measurement matrices, $\mathbf{\Phi}$, with a parametric representation (e.g., line spectral estimation, direction-of-arrival (DoA) estimation), a grid-*less* extension of the proposed algorithm is provided.

We provide numerical results to quantify the performance improvement over OMP. The ideas presented in this work can be easily extended to the multiple measurement vector problem [22] and we demonstrate this in the numerical section. *Notations*: We denote vector of components from vector $\mathbf{v}$ indexed by set $\mathbb{S}$ as $\mathbf{v}_{\mathbb{S}}$. Similarly, we denote the matrix resulting from matrix $\mathbf{M}$ after keeping columns with indices in set $\mathbb{Q}$ as $\mathbf{M}_{\mathbb{Q}}$, unless stated otherwise.

2. LIGHT-WEIGHT SEQUENTIAL SBL ALGORITHM

In contrast to the OMP algorithm [14, 15], where $\mathbf{x}$ is implicitly modeled as a deterministic unknown, SBL models $\mathbf{x}$ as a random variable. More specifically, a parameterized Gaussian prior is imposed on $\mathbf{x}$ with mean zero and uncorrelated components. In other words, $\mathbf{x} \sim \mathcal{CN}(\mathbf{0}, \mathbf{\Gamma})$ where $\mathbf{\Gamma}$ is a diagonal matrix; let $\mathrm{diag}(\mathbf{\Gamma}) = \boldsymbol{\gamma}$. Under the zero mean Gaussian noise assumption, $\mathbf{y} \sim \mathcal{CN}(\mathbf{0}, \mathbf{\Phi\Gamma\Phi}^H + \lambda\mathbf{I})$, where λ denotes the noise variance estimate. The SBL approach is to solve (1) in the hyperparameter $\boldsymbol{\gamma}$-space, post marginalization with respect to $\mathbf{x}$. This approach is known as Type-II estimation framework, compared to Type-I where the problem is solved in $\mathbf{x}$ space after marginalization with respect to $\boldsymbol{\gamma}$ [23] (e.g., Lasso algorithm [24]). Note that the Gaussian density imposition on $\mathbf{x}$ is not a limitation, and the method generalizes well to the case when $\mathbf{x}$ was in fact drawn from a non-Gaussian density (e.g., see Section VI.A. in [12]). The hyperparameter $\boldsymbol{\gamma}$ is estimated by minimizing the negative marginal log-likelihood function [12, 25]

$$\min_{\boldsymbol{\gamma}\geq\mathbf{0},\lambda\geq 0} \mathcal{L}(\boldsymbol{\gamma}, \lambda) := \log\det\mathbf{\Sigma}_{\mathbf{y}} + \mathbf{y}^H\mathbf{\Sigma}_{\mathbf{y}}^{-1}\mathbf{y}. \tag{2}$$

where, $\mathbf{\Sigma}_{\mathbf{y}} = \mathbf{\Phi\Gamma\Phi}^H + \lambda\mathbf{I}$. (2) is a non-convex problem in $(\boldsymbol{\gamma}, \lambda)$.

Similar to OMP, sequential SBL algorithm [13] selects one column per iteration. However, in contrast to OMP, each iteration in sequential SBL optimizes the maximum likelihood based cost function in (2). We initialize with $\gamma_j = 0, \forall j \in \{1, \ldots, n\}$, and add a column l that minimizes the negative log-likelihood, the most. The sequential SBL approach in [13] is a framework and does not provide a specific algorithm, but rather suggestions (see Section 4 in [13]) for developing variants with different options, e.g. adding/deleting selected columns, modifying the variances of the columns already selected, etc. To ensure a computational complexity comparable to OMP, we develop a specific algorithm, which like OMP, runs the sequential steps for K iterations only; K denotes the desired support size.

This research was supported by National Science Foundation (NSF) under Grant CCF-2124929 and Grant CCF-2225617, and the UCSD Center for Wireless Communications.

2.1. LWS-SBL Algorithm

We focus on estimating $\boldsymbol{\gamma}$ for a fixed λ. The latter may be estimated [13] but is not discussed in this paper. We begin by separating out the contribution of the j-th column to the cost function, $\mathcal{L}(\boldsymbol{\gamma})$, in (2). Let $\mathbb{T} \subset \{1, \dots, n\}$ denote the set of column indices of $\boldsymbol{\Phi}$ already selected and $\mathbf{C} = \boldsymbol{\Phi}_{\mathbb{T}}\boldsymbol{\Gamma}_{\mathbb{T}}\boldsymbol{\Phi}_{\mathbb{T}}^H + \lambda\mathbf{I}$, where $\boldsymbol{\Gamma}_{\mathbb{T}}$ denotes the diagonal matrix with rows and columns in $\mathbb{T}$. For $j \notin \mathbb{T}$, we can write

$$\mathcal{L}(\boldsymbol{\gamma}_{\mathbb{T}\cup\{j\}}) = \mathcal{L}(\boldsymbol{\gamma}_{\mathbb{T}}) + L(\gamma_j, \mathbf{C}), \tag{3}$$

where $L(\gamma_j, \mathbf{C}) = \log(1 + \gamma_j \boldsymbol{\Phi}_j^H \mathbf{C}^{-1}\boldsymbol{\Phi}_j) - |\boldsymbol{\Phi}_j^H \mathbf{C}^{-1}\mathbf{y}|^2/(\gamma_j^{-1} + \boldsymbol{\Phi}_j^H \mathbf{C}^{-1}\boldsymbol{\Phi}_j)$ (see eq. (18) in [13] for detailed derivation of (3)). Let us introduce the following quantities for ease of presentation:

$$s_j = \boldsymbol{\Phi}_j^H \mathbf{C}^{-1}\boldsymbol{\Phi}_j \text{ and } q_j = \boldsymbol{\Phi}_j^H \mathbf{C}^{-1}\mathbf{y}. \tag{4}$$

Since γ_j is initialized with zero i.e., $\gamma_{\text{prev},j} = 0$, we have $\forall j \notin \mathbb{T}$

$$\Delta L_{\mathbf{C}}(\gamma_j, \gamma_{\text{prev},j}) = L(\gamma_j, \mathbf{C}) - L(\gamma_{\text{prev},j}, \mathbf{C}) = L(\gamma_j, \mathbf{C}). \tag{5}$$

In other words, the change in negative log-likelihood due to updating γ_j is simply its contribution to the cost function. This helps to simplify the objective further. The column to be added is given by

$$l = \underset{j\notin\mathbb{T}}{\arg\min}\ \min_{\gamma_j \geq 0} L(\gamma_j, \mathbf{C})\ \left(= \Delta L_{\mathbf{C}}(\gamma_j, \gamma_{\text{prev},j})\right). \tag{6}$$

Minimization with respect to γ_j can be obtained in closed-form as

$$\gamma_j^{\text{opt}} = \max\left\{\frac{|q_j|^2 - s_j}{s_j^2}, 0\right\} \tag{7}$$

At the optimal value γ_j^{opt} we have

$$L(\gamma_j^{\text{opt}}, \mathbf{C}) = \begin{cases} \log\frac{|q_j|^2}{s_j} - \frac{|q_j|^2}{s_j} + 1 & \text{if } |q_j|^2 > s_j \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Note that $\log\frac{|q_j|^2}{s_j} - \frac{|q_j|^2}{s_j} + 1 \leq 0$ with equality if and only if $q_j = s_j$. Since $L(\gamma_j^{\text{opt}}, \mathbf{C})$ is a monotonic non-increasing function of $\frac{|q_j|^2}{s_j}$ with $L(\gamma_j^{\text{opt}}, \mathbf{C}) = 0$ if $|q_j|^2 \leq s_j$ we can simplify the underlying problem in (6) as

$$l = \underset{j\notin\mathbb{T}}{\arg\max} \quad R_{\mathbf{C}}(j) := \max\left\{\frac{|q_j|^2}{s_j}, 1\right\}. \tag{9}$$

Algorithm 1 summarizes the proposed steps.

Remark 1. An important characteristic of the sequential SBL [13] and the proposed LWS-SBL Algorithm is that they may be run for more than m iterations if needed, unlike OMP [26]. This flexibility is achieved by avoiding the orthogonal residue computation step as in OMP. Such a flexibility may be useful to correct erroneous support that is possible in initial iterations.

2.1.1. *Efficient updating of q_j and s_j*

To lower the complexity of the steps in Algorithm 1, we exploit the fact that one column of $\boldsymbol{\Phi}$ is added per iteration and update $\mathbf{C}^{-1}$ using matrix inversion lemma. We highlight the value at iteration i with superscript $(\cdot)^{[i]}$. More specifically, let $(\mathbf{C}^{[i]})^{-1}$ denote the value used to compute (q_j, s_j) as per (4), and $l^{[i]}$ denote the column index to be added at iteration i (Step 3 & 4 of Algorithm 1). Then

$$(\mathbf{C}^{[i+1]})^{-1} = (\mathbf{C}^{[i]} + \hat{\gamma}_{l^{[i]}}\boldsymbol{\Phi}_{l^{[i]}}\boldsymbol{\Phi}_{l^{[i]}}^H)^{-1} = (\mathbf{C}^{[i]})^{-1} - \frac{\mathbf{w}^{[i]}(\mathbf{w}^{[i]})^H}{1/\hat{\gamma}_{l^{[i]}} + s_{l^{[i]}}^{[i]}}$$

Algorithm 1: Light-Weight Sequential SBL Algorithm

Result: $\hat{\boldsymbol{\gamma}}$, Posterior mean $\hat{\boldsymbol{\mu}}_{\mathbf{x}} = \hat{\mathbf{x}}$, covariance $\hat{\boldsymbol{\Sigma}}_{\mathbf{x}}$
Input: $\mathbf{y}, \boldsymbol{\Phi}, K$

1 Initialize: $\hat{\boldsymbol{\gamma}} = \mathbf{0}$, $\lambda =$ *some sensible value* (e.g., $0.1\text{var}(\mathbf{y})$), $\mathbf{C}^{-1} = \lambda^{-1}\mathbf{I}, \mathbb{T} = \emptyset$
2 **for** $i := 1$ to K **do**
3 Compute q_j and $s_j, \forall j \notin \mathbb{T}$ as in (4) (efficient recursive implementation in (10))
4 $l = \arg\max_{j\notin\mathbb{T}} \quad R_{\mathbf{C}}(j) := \max\left\{\frac{|q_j|^2}{s_j}, 1\right\}$
5 $\hat{\gamma}_l = \max\left\{\frac{|q_l|^2 - s_l}{s_l^2}, 0\right\}$; rank-one update of $\mathbf{C}^{-1}$
6 $\mathbb{T} = \mathbb{T} \cup l$ (unless $\hat{\gamma}_l = 0$, rare)
7 **end**
8 Compute posterior mean, $\hat{\boldsymbol{\mu}}_{\mathbf{x}}$, and covariance, $\hat{\boldsymbol{\Sigma}}_{\mathbf{x}}$

---

where $\mathbf{w}^{[i]} = (\mathbf{C}^{[i]})^{-1}\boldsymbol{\Phi}_{l^{[i]}}$ and $s_{l^{[i]}}^{[i]} = \boldsymbol{\Phi}_{l^{[i]}}^H(\mathbf{C}^{[i]})^{-1}\boldsymbol{\Phi}_{l^{[i]}}$. Then we can update $\left(q_j^{[i]}, s_j^{[i]}\right)$ to get $\left(q_j^{[i+1]}, s_j^{[i+1]}\right)$ as

$$\begin{aligned} q_j^{[i+1]} &= \boldsymbol{\Phi}_j^H \left(\mathbf{C}^{[i+1]}\right)^{-1}\mathbf{y} = q_j^{[i]} - \frac{\boldsymbol{\Phi}_j^H \mathbf{w}^{[i]}}{1/\hat{\gamma}_{l^{[i]}} + s_{l^{[i]}}^{[i]}}\, q_{l^{[i]}}^{[i]}, \\ s_j^{[i+1]} &= \boldsymbol{\Phi}_j^H \left(\mathbf{C}^{[i+1]}\right)^{-1}\boldsymbol{\Phi}_j = s_j^{[i]} - \frac{|\boldsymbol{\Phi}_j^H \mathbf{w}^{[i]}|^2}{1/\hat{\gamma}_{l^{[i]}} + s_{l^{[i]}}^{[i]}}. \end{aligned} \tag{10}$$

2.1.2. *Computing posterior mean, $\hat{\boldsymbol{\mu}}_{\mathbf{x}}$, and covariance, $\hat{\boldsymbol{\Sigma}}_{\mathbf{x}}$*

The posterior on $\mathbf{x}$ is a complex Gaussian distribution [12, 25]. We report $\hat{\mathbf{x}} = \hat{\boldsymbol{\mu}}_{\mathbf{x}}$ as a point estimate after K iterations using

$$\hat{\boldsymbol{\mu}}_{\mathbf{x}} = \hat{\boldsymbol{\Gamma}}\boldsymbol{\Phi}^H \left(\boldsymbol{\Phi}\hat{\boldsymbol{\Gamma}}\boldsymbol{\Phi}^H + \lambda\mathbf{I}\right)^{-1}\mathbf{y} = \hat{\boldsymbol{\Gamma}}\mathbf{q}^{[K+1]}, \tag{11}$$

where $\mathbf{q}^{[K+1]} = \left[q_1^{[K+1]}, \dots, q_n^{[K+1]}\right]^T$. Note that $\hat{\gamma}_j \neq 0$ only for $j \in \mathbb{T}$, and thus the above equation provides a sparse solution. Also, the diagonal entries of $\hat{\boldsymbol{\Sigma}}_{\mathbf{x}}$ can be easily obtained as following

$$[\hat{\boldsymbol{\Sigma}}_{\mathbf{x}}]_{j,j} = \hat{\gamma}_j - \hat{\gamma}_j^2 \boldsymbol{\Phi}_j^H \left(\boldsymbol{\Phi}\hat{\boldsymbol{\Gamma}}\boldsymbol{\Phi}^H + \lambda\mathbf{I}\right)^{-1}\boldsymbol{\Phi}_j = \hat{\gamma}_j - \hat{\gamma}_j^2 s_j^{[K+1]}, \tag{12}$$

which is similarly non-zero only for $j \in \mathbb{T}$.

2.2. Computational Complexity of Algorithm 1

We analyze the computational complexity for the proposed LWS-SBL algorithm and compare with OMP.

2.2.1. *Efficient computation of $\mathbf{w}^{[i]}$*

The update equations in (10) require the vector $\mathbf{w}^{[i]} = (\mathbf{C}^{[i]})^{-1}\boldsymbol{\Phi}_{l^{[i]}}$ which can be computed by first computing $(\mathbf{C}^{[i]})^{-1}$ from $(\mathbf{C}^{[i-1]})^{-1}$ using a rank-one update, followed by multiplying $\boldsymbol{\Phi}_{l^{[i]}}$. These steps need $O(m^2)$ computations. Instead we propose the following steps

$$\begin{aligned} \mathbf{w}^{[i]} &= (\mathbf{C}^{[i]})^{-1}\boldsymbol{\Phi}_{l^{[i]}} = (\mathbf{C}^{[i-1]})^{-1}\boldsymbol{\Phi}_{l^{[i]}} - \frac{\mathbf{w}^{[i-1]}(\mathbf{w}^{[i-1]})^H\boldsymbol{\Phi}_{l^{[i]}}}{1/\gamma_{l^{[i-1]}} + s_{l^{[i-1]}}^{[i-1]}} \\ &= (\mathbf{C}^{[1]})^{-1}\boldsymbol{\Phi}_{l^{[i]}} - \sum_{\tilde{i}=1}^{i-1} \frac{\mathbf{w}^{[\tilde{i}]}(\mathbf{w}^{[\tilde{i}]})^H\boldsymbol{\Phi}_{l^{[i]}}}{1/\gamma_{l^{[\tilde{i}]}} + s_{l^{[\tilde{i}]}}^{[\tilde{i}]}}, \end{aligned} \tag{13}$$

where $(\mathbf{C}^{[1]})^{-1} = \lambda^{-1}\mathbf{I}$. $(\mathbf{C}^{[1]})^{-1}$ may be pre-computed; the rest requires $O(m(i-1))$ computations, an improvement over $O(m^2)$. In this manner, we avoid computing $\mathbf{C}^{-1}$ in Step 5 of Algorithm 1.

2.2.2. *Computational Complexity*

Analyzing and comparing two algorithms based on computational complexity is challenging because it depends on the metric employed, i.e. number and type of operations, parallel[1] versus sequential computations, storage, etc. For this work, we track the number of arithmetic operations, in particular, multiplications and divisions as they are more computationally demanding than additions or subtractions. We further assume that multiplication and division have similar complexity and do not distinguish them in this aspect, but highlight them by mentioning 'div' for divisions explicitly (e.g., n divisions represented as n 'div'). We summarize the computational complexity for LWS-SBL and compare with OMP in Table 1.

	LWS-SBL	**OMP (QR)** [14, 27]
Iteration $i+1$	$mn + 2mp + n + m$ $+2(n-i) + (n-i)$ 'div'	$mn + 2mp + n + m$
Solution $\hat{\mathbf{x}}$	$m(3K-1) + 2K$	$(K-1)K/2 + K$ 'div'

LWS-SBL vs. OMP: Computational Complexity ($p = i - 1$)

Table 1.

Remark 2. The extra $2(n-i) + (n-i)$ 'div' for LWS-SBL (than OMP) corresponds to updating s_j and dividing $|q_j|^2$ and $s_j, \forall j \notin \mathbb{T}$.

During the first iteration s_j does not depend on j as $(\mathbf{C}^{[1]})^{-1} = \lambda^{-1}\mathbf{I}$ and thus the divisions may be avoided. *Note that both the algorithms have $O(mn)$ per iteration computational complexity*.

3. GRIDLESS LWS-SBL ALGORITHM

We now discuss the case when $\mathbf{\Phi}$ has a parametric representation, and extend LWS-SBL to perform gridless parameter estimation. Consider the following parametric data model

$$\mathbf{y} = \mathbf{\Phi}_{\boldsymbol{\theta}}\mathbf{x} + \mathbf{n}, \tag{14}$$

where the kth column of $\mathbf{\Phi}_{\boldsymbol{\theta}} \in \mathbb{C}^{m \times K}$ is a vector function of the parameter θ_k i.e., $[\mathbf{\Phi}_{\boldsymbol{\theta}}]_k = \boldsymbol{\phi}(\theta_k)$ for some known $\boldsymbol{\phi}(.), k \in \{1, \ldots, K\}$. $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_K]^T$ and θ_k's lie in some known continuous domain. K denotes the number of active sources. Model assumptions made in (1) are also applicable in (14). The above problem is ubiquitous, with applications such as line spectral estimation [28] and direction-of-arrival (DoA) estimation [29] for narrowband signals; we emphasize the latter as means for exposition.

3.1. Grid-based Remodeling of (14)

The proposed Algorithm 1 in its present form is not amenable to the parametric problem in (14). Mathematically, the approach can be readily developed by modifying equation (6) to reflect optimization over the continuous parameter space rather than the index set. However, this step when implemented will require a grid search though the grid is not predetermined. As a practical implementation of this concept, we adopt a two-step approach where in step-1 we recover an on-grid support set using Algorithm 1. This is followed by step-2 where we locally optimize the recovered grid points over a smaller parameter space. This step is inspired by a previous work by the authors [30] on batch EM-SBL where all components of $\boldsymbol{\gamma}$ are simultaneously updated. In contrast to [30], this paper proposes a *fixed iterations* strategy to identify a sparse solution, with an emphasis on computational complexity reduction.

Assuming a grid size n, we introduce a dictionary $\mathbf{\Phi}$ with array manifold vectors as columns i.e., $[\mathbf{\Phi}]_j = \boldsymbol{\phi}(\theta_j), \theta_j \in [\frac{-\pi}{2}, \frac{\pi}{2}), j \in \{1, \ldots, n\}$. For example, consider a uniform linear array (ULA) with m sensors and $d = \bar{\lambda}/2$ distance between adjacent sensors to prevent ambiguity in DoA estimation; $\bar{\lambda}$ denotes the wavelength of the incoming narrowband source signals. Then $[\mathbf{\Phi}]_j = \boldsymbol{\phi}(\theta_j) = [1, \exp(-i\pi u_j), \ldots, \exp(-i(m-1)\pi u_j)]^T, u_j = \sin\theta_j$.

[1]Note that updating q_j's and s_j's may be parallelized.

Algorithm 2: Gridless LWS-SBL Algorithm

Result: $\tilde{\boldsymbol{\theta}} = [\tilde{\theta}_1, \ldots, \tilde{\theta}_K]^T, \tilde{\boldsymbol{\gamma}}, \tilde{\boldsymbol{\mu}}_{\mathbf{x}} = \tilde{\mathbf{x}}, \tilde{\mathbf{\Sigma}}_{\mathbf{x}}$
Input: $\mathbf{y}, \mathbf{\Phi}, K$; $(\hat{\boldsymbol{\gamma}}, \mathbf{C}^{-1}, \lambda)$ from Algorithm 1; $\tilde{n}$, ITER
Initialize: $\tilde{\boldsymbol{\gamma}} = \hat{\boldsymbol{\gamma}}$
for iter $:= 1$ to ITER **do**
  for $i := 1$ to K **do**
    Rank-one update of $\mathbf{C}^{-1}$ to get $\mathbf{C}_{-l_i}^{-1}$
    $\mathbf{\Phi}_{\text{local}} :=$ local dictionary of size $\tilde{n}$ around u_{l_i}
    Compute $q(u)$ and $s(u)$ as in (16) using $\mathbf{\Phi}_{\text{local}}$
    Solve (19); update $(\tilde{\gamma}_{l_i}, u_{l_i}) := (\gamma^{\text{opt}}, u^{\text{opt}})$
    Rank-one update of $\mathbf{C}_{-l_i}^{-1}$ to get $\mathbf{C}^{-1}$
  end
end
Compute $\tilde{\boldsymbol{\mu}}_{\mathbf{x}}, \tilde{\mathbf{\Sigma}}_{\mathbf{x}}$; $\tilde{\theta}_i = \arcsin u_{l_i}, i \in \{1, \ldots, K\}$

3.2. Gridless LWS-SBL Algorithm

The goal is to further maximize the likelihood after the initial K iterations by locally optimizing the selected grid points. This enables to go beyond the limitations of the initial grid. We begin by separating out the l_i-th dictionary component, $i \in \{1, \ldots, K\}$, selected previously as part of Algorithm 1 and optimize its contribution to the likelihood not only with respect to (w.r.t.) the corresponding γ_{l_i}, but also w.r.t. the grid point $u_{l_i} = \sin\theta_{l_i}$. Thus, we write

$$\mathcal{L}(\boldsymbol{\gamma}_{\mathbb{T}}) = \mathcal{L}(\boldsymbol{\gamma}_{\mathbb{T}\setminus\{l_i\}}) + \tilde{L}(\gamma_{l_i}, u_{l_i}, \mathbf{C}_{-l_i}), \tag{15}$$

where $\mathbf{C}_{-l_i} = \mathbf{\Phi}_{\mathbb{T}\setminus\{l_i\}}\mathbf{\Gamma}_{\mathbb{T}\setminus\{l_i\}}\mathbf{\Phi}_{\mathbb{T}\setminus\{l_i\}}^H + \lambda\mathbf{I}$, $\tilde{L}(\gamma, u, \mathbf{C}_{-l_i}) = \log(1 + \gamma s(u)) - |q(u)|^2/(\gamma^{-1} + s(u))$, and

$$s(u) = \phi(u)^H\mathbf{C}_{-l_i}^{-1}\phi(u), \quad q(u) = \phi(u)^H\mathbf{C}_{-l_i}^{-1}\mathbf{y}. \tag{16}$$

For ease of exposition, we represent a grid point with the notation $\Psi = (\gamma, u)$. We locally maximize the likelihood in the neighbourhood of $\Psi_{\text{prev}} = (\hat{\gamma}_{l_i}, u_{l_i})$ by equivalently minimizing

$$\Delta\tilde{L}_{\mathbf{C}_{-l_i}}(\Psi, \Psi_{\text{prev}}) = \tilde{L}(\gamma, u, \mathbf{C}_{-l_i}) - \tilde{L}(\hat{\gamma}_{l_i}, u_{l_i}, \mathbf{C}_{-l_i}). \tag{17}$$

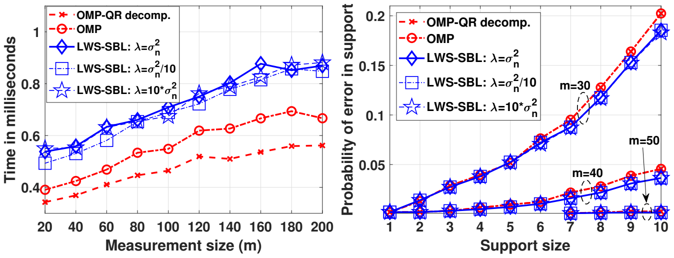
Note that unlike in the previous section, the second term above is non-zero. However, it does not depend on Ψ and is therefore fixed. Another perspective to understand this is that we are replacing the same grid point Ψ_{prev} with any other point in its neighbourhood, and thus in order to maximize likelihood we only need to minimize the first term on RHS in (17). The underlying problem reduces to

$$\Psi^{\text{opt}} = (\gamma^{\text{opt}}, u^{\text{opt}}) = \underset{u \in [u_{l_i} - \delta, u_{l_i} + \delta)}{\arg\min} \min_{\gamma \geq 0} \tilde{L}(\gamma, u, \mathbf{C}_{-l_i}), \tag{18}$$

for some δ[2]. Similar steps as in previous section can be followed to simplify the objective and we get

$$u^{\text{opt}} = \underset{u \in [u_{l_i} - \delta, u_{l_i} + \delta)}{\arg\max} \tilde{R}_{\mathbf{C}_{-l_i}}(u) := \max\left\{\frac{|q(u)|^2}{s(u)}, 1\right\},$$
$$\text{and } \gamma^{\text{opt}} = \max\left\{\frac{|q(u^{\text{opt}})|^2 - s(u^{\text{opt}})}{s(u^{\text{opt}})^2}, 0\right\}. \tag{19}$$

[2]We set $\delta = 1/n$ during the first iteration to avoid any grid point overlap. Future iterations may involve non-uniform grid, and a similar asymmetric δ in either directions is used to avoid the same issue.



(a) (b)

Fig. 1. (a) Computation time vs. m. (b) Probability of error in support vs. support size, K.

We implement the optimization for u in the neighbourhood of Ψ_{prev} by a fine grid of size $\tilde{n}$. We summarize the steps for the proposed gridless approach in Algorithm 2.

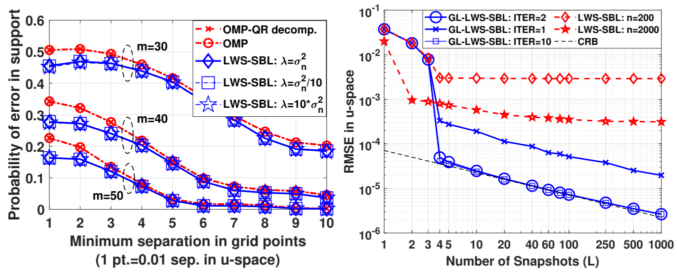
Remark 3. The proposed approach includes the previous grid point Ψ_{prev} in the search and thus ensures that the likelihood increases steadily over the iterations. Also, Algorithm 2 iterates over the same source multiple times, which helps to account for the potential errors in localizing other sources in the set $\mathbb{T}\backslash\{l_i\}$.

4. NUMERICAL RESULTS

We evaluate the proposed sequential SBL algorithms for support recovery performance and computational time, and compare with OMP. For OMP, we implement two variations which differ only in the estimation of the sparse coefficients, $\hat{\mathbf{x}}$, a) using the QR decomposition of selected columns b) using pseudo-inverse. Therefore, the two differ in the amount of computational time to run, and serve as benchmark for comparison with the proposed algorithm. For the proposed algorithm, we plot three curves a) $\lambda = \sigma_n^2$ b) $\lambda = \sigma_n^2/10$ c) $\lambda = 10\ \sigma_n^2$. This helps to assess the performance under different assumptions of knowledge of σ_n^2. To evaluate the support recovery performance we adopt the distance metric $\text{dist}(\mathbb{S}, \hat{\mathbb{S}}) = 1 - |\mathbb{S} \cap \hat{\mathbb{S}}| / \max\{|\mathbb{S}|, |\hat{\mathbb{S}}|\}$, where $\mathbb{S}$ and $\hat{\mathbb{S}}$ denote the actual and recovered support sets respectively, used in [31] (see eq. (3.29) and Fig. 3.6. in [31]). We present results for ULA geometry, but similar results were obtained for $\mathbf{\Phi}$ with Gaussian random entries. The grid size is $n = 200$. Thus the grid spacing between adjacent points is $2/n = 0.01$ in u-space. The signal-to-noise ratio (SNR) is computed per source at the receiver and is fixed at 30 dB. Results are averaged over 500 realizations, unless otherwise specified.

Experiment 1: In Fig. 1 (a) we plot the amount of time required to run the different algorithms as a function of measurement size, m. The simulations are carried out in MATLAB 9.4.0.813654 in a Windows 10 system using a 2.7 GHz CPU. As evident from the plot, the proposed sequential SBL algorithm requires larger amount of time than the OMP algorithm. OMP using QR decomposition requires the least amount of time to complete. The plots also help to ascertain the *linear dependency of their respective complexities on* m.

Experiment 2: In Fig. 1 (b), we plot the support recovery performance as a function of support size. We consider different measurement sizes $m = \{30, 40, 50\}$. We ensure that the minimum separation between sources is at least 10 grid points or equivalently 0.1 in u-space. As expected the probability of error in support increases as the support size increases, and is higher for smaller measurement sizes. For $m = 50$, the probability of error is zero for sizes up to 6, beyond which it is non-zero. It is evident from the curves that the proposed LWS-SBL algorithm has *better support recovery performance than OMP*. Also, the support recovery performance for



(a) (b)

Fig. 2. (a) Probability of error in support vs. minimum allowed separation between support elements. (b) RMSE vs. number of snapshots, L (plots using $\lambda = \sigma_n^2$).

different noise variance parameter $\lambda \in \{0.1\sigma_n^2, \sigma_n^2, 10\sigma_n^2\}$ setting within LWS-SBL is comparable, indicating *low sensitivity to these variations*. On the other hand, the sparse coefficients estimation, $\hat{\mathbf{x}}$, were observed to be more sensitive to setting λ but the effect was mild.

Experiment 3: In Fig. 2 (a) we plot the support recovery performance as a function of the minimum separation between the support elements. Note that the dictionary and support size ($K = 10$) are fixed along the x-axis, only the support *set* changes to ensure the minimum separation between sources is as per the x-axis tick point. We consider $m = \{30, 40, 50\}$. As observed from the Fig. 2 (a), the proposed LWS-SBL algorithm has a better support recovery performance than OMP. *The difference is more prominent when the sources are closely located.* Again, in this experiment it was observed that the support recovery performance was less sensitive to setting λ parameter differently from the true value (to $\lambda \in \{0.1\sigma_n^2, 10\sigma_n^2\}$). The $\hat{\mathbf{x}}$ estimation was more sensitive to such settings, albeit mildly.

Experiment 4: In Fig. 2 (b) we plot the root mean squared error (RMSE) as a function of number of snapshots (L) for the gridless (GL) LWS-SBL algorithm presented in Algorithm 2. Parameters used: $m = 50$, $\tilde{n} = 10n$, support size $K = 10$, minimum separation between support elements of 0.1 in u-space, and the results are averaged over 100 realizations. As evident from the plots, even ITER$= 1$ ensures that the error steadily decreases with L. ITER$= 2$ *(blue curve with circle markers) was sufficient to further ensure that the gap with the Cramér-Rao bound (CRB) is small.* It was also observed that, when fewer snapshots ($L < 4$) are available, a finer grid may be needed to prevent support recovery errors within LWS-SBL (compare red curve with pentagram markers and blue curve with circle markers). *Another alternative can be to run the LWS-SBL i.e., Algorithm 1 for more than K iterations.* Finally, note that the Algorithm 2 may be extended in a multi-resolution fashion to control the grid size at any iteration. In this case, the effective grid resolution of $\frac{2}{n} \cdot \frac{1}{10n}$ was achieved with $O(n)$ complexity instead of $O(n^2)$ required by running the grid based algorithm with a fine initial grid.

5. CONCLUSION

We proposed a Light-Weight Sequential SBL (LWS-SBL) algorithm. The computational complexity of LWS-SBL is also $O(mn)$, similar to that of OMP. We demonstrated the improved support recovery performance using parametric dictionaries with high mutual coherence, and allowing sources to be closely separated. We also proposed a two-step gridless algorithm that allows to go beyond the initial grid limitation by locally optimizing the likelihood function around potential source locations. This method is shown to empirically approach CRB. Future work includes efficient implementation of the gridless extension of the proposed algorithm.

6. REFERENCES

[1] I.F. Gorodnitsky, J.S. George, and B.D. Rao, "Neuromagnetic source imaging with FOCUSS: a recursive weighted minimum norm algorithm.," *J. Electroencephalog. Clinical Neurophysiol.*, vol. 95, no.4, pp. 231–251, 1995.

[2] B. K. Natarajan, "Sparse Approximate Solutions to Linear Systems," *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, 1995.

[3] D. L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 508–518, 2000.

[4] M. Vetterli, P. Marziliano, and T. Blu, "Sampling signals with finite rate of innovation," *IEEE Transactions on Signal Processing*, vol. 50, no. 6, pp. 1417–1428, 2002.

[5] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.

[6] J. Trzasko and A. Manduca, "Highly undersampled magnetic resonance image reconstruction via homotopic ℓ_0 -minimization," *IEEE Transactions on Medical Imaging*, vol. 28, no. 1, pp. 106–121, 2009.

[7] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[8] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[9] B. D. Rao and K. Kreutz-Delgado, "An affine scaling methodology for best basis selection," *IEEE Transactions on Signal Processing*, vol. 47, no. 1, pp. 187–200, 1999.

[10] J.A. Tropp, "Just relax: convex programming methods for identifying sparse signals in noise," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 1030–1051, 2006.

[11] D. Malioutov, M. Cetin, and A.S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 3010–3022, 2005.

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

[13] M. E. Tipping and A. C. Faul, "Fast marginal likelihood maximisation for sparse Bayesian models," in *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*. 03–06 Jan 2003, vol. R4, pp. 276–283, PMLR.

[14] Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*, Cambridge University Press, Cambridge ; New York :, 1 edition, 2012.

[15] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, 1993, pp. 40–44 vol.1.

[16] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Constructive Approximation*, vol. 13, pp. 57–98, 1997.

[17] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[18] B. L. Sturm and M. G. Christensen, "Comparison of orthogonal matching pursuit implementations," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, 2012, pp. 220–224.

[19] D. Needell and J.A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[20] G.Z. Karabulut and A. Yongacoglu, "Sparse channel estimation using orthogonal matching pursuit algorithm," in *IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall. 2004*, 2004, vol. 6, pp. 3880–3884 Vol. 6.

[21] J. A. Becerra, M. J. Madero-Ayora, J. Reina-Tosina, C. Crespo-Cadenas, J. García-Frías, and G. Arce, "A doubly orthogonal matching pursuit algorithm for sparse predistortion of power amplifiers," *IEEE Microwave and Wireless Components Letters*, vol. 28, no. 8, pp. 726–728, 2018.

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

[23] R. Giri and B. D. Rao, "Type I and Type II bayesian methods for sparse signal recovery using scale mixtures," *IEEE Transactions on Signal Processing*, vol. 64, no. 13, pp. 3418–3428, 2016.

[24] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[25] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[26] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*, Birkhäuser New York, NY, 2013.

[27] S.F. Cotter, J. Adler, B.D. Rao, and K. Kreutz-Delgado, "Forward sequential algorithm for best basis selection," pp. 235–244, 1999.

[28] P. Stoica and R. Moses, *Spectral Analysis of Signals*, Prentice-Hall, Upper Saddle River, NJ, USA, 2005.

[29] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *IEEE Signal Processing Magazine*, vol. 13, no. 4, pp. 67–94, July 1996.

[30] R. R. Pote and B. D. Rao, "Maximum likelihood-based gridless doa estimation using structured covariance matrix recovery and sbl with grid refinement," *arXiv*, 2022.

[31] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer New York, NY, 2010.