Iterative Descent Group Hard Thresholding Algorithms for Block Sparsity

Thierry Chonavel, Abdeldjalil Aissa El Bey, Zahran Hajji

Iterative Descent Group Hard Thresholding Algorithms for Block Sparsity

Thierry Chonavel[a,*], Abdeldjalil Aïssa-El-Bey[a], Zahran Hajji[b]

[a] *IMT Atlantique, UMR CNRS 6285 Lab-STICC, 29238, Brest, France*
[b] *NXP Semiconductors, 134 Avenue Gen. Eisenhower, Toulouse, 31100, France*

Abstract

In this paper we consider the problem of recovering block-sparse structures in a linear regression context. Penalized mean squared criteria are generally considered in such contexts where $\ell_{2,1}$ mixed norm penalty terms is often used as a convex alternative to the $\ell_{2,0}$ penalty. Here, we propose an iterative block cyclic descent algorithm approach to address the case of an $\ell_{2,0}$ penalty. We prove its convergence and illustrate its potential benefit compared to $\ell_{2,1}$ or $\ell_{2,q}$ $(0 < q \leq 1)$ penalization. We also propose a momentum approach for accelerated convergence and an application to sensor positioning for array processing.

Keywords: group sparsity, block coordinate relaxation, $\ell_{2,0}$ regularization, sensor selection, beamforming.

*This work benefited from *DGA (https://www.defense.gouv.fr/dga) financial support via project RAPID n°18-290-6060, lead by Deti Microwave company (www.deti-sa.com).*

*Corresponding author

Email addresses: thierry.chonavel@imt-atlantique.fr (Thierry Chonavel), abdeldjalil.aissaelbey@imt-atlantique.fr (Abdeldjalil Aïssa-El-Bey), zahran.hajji@nxp.com (Zahran Hajji)

1. Introduction

During the last decade, compressed sensing (CS) [1] has become a very popular topic for which a large set of techniques [2] has been proposed. These techniques have spread among various disciplines such as machine learning, wireless communication, medical imaging... The many algorithms that have been proposed to recover sparse signals can be classified into several classes. One of the most popular is dedicated to convex relaxation of minimization criteria involving ℓ_0-norm penalty terms. Well known algorithms of this family are Basis pursuit (BP) method [3] as an ℓ_1-relaxation and Focal Underdetermined System Solver (FOCUSS) algorithm [4] that replaces the ℓ_0-norm by an almost everywhere differentiable ℓ_p norm, with $p \in (0, 1]$. In particular, the choice of the ℓ_1 norm is very popular as it preserves convexity. Unfortunately the ℓ_1-norm penalty term can induce high bias. To overcome this issue, iteratively reweighted algorithms, such as iteratively reweighted ℓ_1-minimization [5] have been proposed. A second class involves greedy algorithms such as Matching Pursuit (MP) [6] or Orthogonal Matching Pursuit (OMP) [7]. They are relatively fast iterative procedures that have been used extensively in applications. However, they don't guarantee the convergence of the cost function to a global minimum, contrary to the first family which potentially offers better performance but requires higher computational effort. A third class consists of iterative thresholding algorithms such as iterative soft thresholding algorithms as a solution of a problem with convex regularization or the iterative hard thresholding in the case of a solution of ℓ_0-norm regularization. They can be seen as a projection, operated by a certain thresholding operator. They can be easily implemented and have relatively

low computational complexity even for large-scale problems. Another family involves cyclic coordinate descent algorithms that can be used in both convex and non-convex penalized least squares problems [8],[9]. Surveys about sparse recovery can be found for instance in [10], [11].

In many applications such as array processing, face recognition [12], or data clustering [13], data are block-sparse i.e zero and non zero elements are grouped into blocks. Then, the algorithms discussed above can be extended to this case. In particular, mixed norms such as $\ell_{2,1}$-norm can be considered leading to algorithms like Group Lasso [14] or Group OMP (GOMP) [15] for greedy algorithms. The major idea of extension to block-sparse signals is to use block-wise formulation of the problem instead of the coordinate-wise one. In [14], a group Lasso algorithm is proposed as a cyclic coordinate descent for $\ell_{2,1}$-norm regularized minimization. It ensures sparsity of the solution, but the group sparsity may not occur in all situations. More generally, several criteria and applications have been considered in the literature to take block sparsity into account (see for instance [16, 17, 18] for some recent references).

Image processing in particular, and specifically in the context of rPCA (robust Principal Components Analysis), is a field where block sparsity has been particularly considered. In rPCA, the block sparse problem is addressed with an observation matrix modeled as the sum of a low rank matrix plus a sparse matrix. In early works [19, 20], sparsity was addressed via an ℓ_1 norm penalty term applied to the sparse matrix. Since then, the case of structured sparsity has been addressed in several works [21, 22, 23] where only a few columns of the sparse matrix are nonzero vectors. Although in these references authors generally start describing the problem introducing

the $\ell_{2,0}$ norm of the sparse matrix, they finally relax the problem to get a convex optimization criterion that involves $\ell_{2,1}$ norm. In [24], an alternating projection method is considered where hard thresholding is used for projection of the sparse matrix on the set of sparse matrices. In [25], the ℓ_0 sparsity penalty has been addressed for rPCA and an ADMM approach with proved convergence has been supplied. But in the many works devoted to it, to our best knowledge works in rPCA do not address the $\ell_{2,0}$ penalty approach.

Among other $\ell_{2,0}$ penalty approaches, in [26] the authors propose and iterative gradient descent algorithm where, at each step, the update is projected on a k-blocks subspace, where k is a desired sparsity degree. A greedy approach provides approximate optimal selection of blocks used to design the projections. In [27], the authors consider the problem of quaternion block sparse recovery and classification using an ADMM approach.

To bridge the gap between $\ell_{2,0}$ and $\ell_{2,1}$ penalties, it is possible to consider $\ell_{2,q}$ penalties, with $0 < q \leq 1$. In [28] the authors proposed an iterative reweighted least squares (IRLS) algorithm to optimize the corresponding criteria.

In this paper, we consider an $\ell_{2,0}$-norm regularized minimization criterion. We propose a cyclic block descent algorithm for solving group hard thresholding algorithm that can impose block sparsity better than group Lasso. First, we consider the case where the mixing matrix is block orthonormal and we propose a group hard thresholding algorithm. Then, we justify its convergence to a stationary point and show that convergence is linear. Moreover, we extend the algorithm to the non-orthonormal case and we propose a momentum approach for convergence acceleration.

We show the benefits of our proposal compared to group Lasso [14] or $\ell_{2,q}$ approach [28]. However, there are computationally less demanding approaches than block coordinate or ADMM descent strategies for the selection of sparse blocks. In particular, the Group Orthogonal Matching Pursuit (GOMP) algorithm that has been introduced in [15] proposes direct extension of the Orthogonal Matching Pursuit (OMP) to the case where block sparse solutions are searched for. GOMP is a straightforward greedy approach that extracts blocks with most energy iteratively from the residual signal. We will show that GOMP is often a powerful with accuracy comparable to that of previously mentioned $\ell_{2,0}$ approaches but that runs much faster because it avoids iterating recursively over the vector of blocks and simply converges after a number of iterations equal to the number of desired nonzero blocks. However, we will show on an example that for some problems GOMP may not be satisfactory. More specifically, we will address the problem of sensor selection for array processing from a set of regularly sampled positions. We show how this problem can be formulated as a block sparse linear model with possibly high dimensions and propose an effective approach to solve it for the proposed algorithm and GOMP. While GOMP suffers from widening of the main lobe of the array diagram as the number of selected sensors decreases, due to convergence to a compact array, the proposed solution retains the beamformer resolution of an array covering all positions.

Our contributions are: $(\boldsymbol{i})$ the proposition of a new algorithm, named Group Hard Thresholding Algorithm (GHTA) for linear models where the signal of interest exhibits group sparsity and the mixing matrix is block orthonormal $(\boldsymbol{ii})$ a theoretical study of the convergence of this algorithm,

(iii) extension to the case where the blocks of the mixing matrix are not orthogonal and an implementation of the algorithm with backtracking line search and higher order momentum to accelerate convergence are proposed, and ***(iv)*** a linear model for the selection of sensor positions for beamformers design in array processing applications and related optimization with GHTA.

The paper is organized as follows. Section 2 describes the large-scale block sparse model and block coordinate relaxation algorithms. Section 3 presents a review of block-wise optimality for $\ell_{2,0}$ penalized least squares problems. Section 4 focuses on the presentation of the Group Hard Thresholding Algorithm (GHTA) and its convergence analysis. In this section we will also briefly discuss an $\ell_{2,0}$ penalty approach via ADMM for rPCA and its connection with our group hard thresholding approach. Section 5 describes the extension of the algorithm to the case of non orthogonal blocks and proposes an improved version of the algorithm. Section 6 presents simulation results. In particular, we illustrate the benefits of GHTA in terms of robustness and performance against $\ell_{2,1}$ or more recent $\ell_{2,q}$ $(0 < q \leq 1)$ approaches and we propose an application to sensor positioning for array processing and show that the GOMP algorithm that outperforms iterative approaches in terms of convergence speed while often leading to good error performance fails for this application while GHTA appears useful. Finally, Section 7 concludes the paper.

Notations Now, let us introduce some notations. Boldface upper case letters and boldface lower case letters denote matrices and vectors, respectively. Transpose, transpose conjugate and conjugate operations are denoted by $(.)^T$, $(.)^H$ and $(.)^*$, respectively. $(.)^\#$ denotes the pseudo-inverse. $\otimes$ is the

Kronecker product. $\boldsymbol{I}_k$ is the $k \times k$ identity matrix and $\mathbf{1}_k$ is the all-one vector of $\mathbb{R}^k$. Letting $\underline{\boldsymbol{z}} \in \mathbb{C}^k$ be a complex-valued vector, we denote by $\boldsymbol{z} = \begin{bmatrix} \mathrm{Re}\,(\underline{\boldsymbol{z}})^T & \mathrm{Im}\,(\underline{\boldsymbol{z}})^T \end{bmatrix}^T$ its representation in $\mathbb{R}^{2k}$. Let us consider a matrix $\underline{\boldsymbol{A}} \in \mathbb{C}^{m \times n}$. Then, $\boldsymbol{A} = \begin{bmatrix} \mathrm{Re}\,(\underline{\boldsymbol{A}}) & -\mathrm{Im}\,(\underline{\boldsymbol{A}}) \\ \mathrm{Im}\,(\underline{\boldsymbol{A}}) & \mathrm{Re}\,(\underline{\boldsymbol{A}}) \end{bmatrix}$ will represent its real-valued matrix version in $\mathbb{R}^{2m \times 2n}$.

2. Problem formulation

Let us consider a large-scale block sparse model, which can be described as a linear connection between a group sparse vector $\underline{\boldsymbol{w}} \in \mathbb{C}^M$ and its observations $\underline{\boldsymbol{y}} \in \mathbb{C}^K$ by

$$\underline{\boldsymbol{y}} = \underline{\boldsymbol{A}}\boldsymbol{w} + \underline{\boldsymbol{\zeta}}, \tag{1}$$

where $\underline{\boldsymbol{\zeta}} \in \mathbb{C}^K$ denotes a zero mean complex circular Gaussian noise vector with covariance matrix $\sigma^2 \boldsymbol{I}$ and $\underline{\boldsymbol{A}} \in \mathbb{C}^{K \times M}$ is a complex-valued mixing matrix. Let us suppose that the entries of the vector $\boldsymbol{w}$ are grouped into N non overlapping blocks. The size of n-th block is denoted by m_n. Sparsity lies in the fact that most blocks have all their entries equal to zero. The block representation of $\underline{\boldsymbol{w}}$ writes $\underline{\boldsymbol{w}} = [\underline{\boldsymbol{w}}_1^T, \underline{\boldsymbol{w}}_2^T, ..., \underline{\boldsymbol{w}}_N^T]^T$.
In the same way, letting $\underline{\boldsymbol{A}} = [\underline{\boldsymbol{A}}_1, \underline{\boldsymbol{A}}_2, ..., \underline{\boldsymbol{A}}_N]$ where $\underline{\boldsymbol{A}}_n \in \mathbb{C}^{K \times m_n}$ the complex system (1) can be rewritten as follows:

$$\underline{\boldsymbol{y}} = \sum_{n=1}^{N} \underline{\boldsymbol{A}}_n \underline{\boldsymbol{w}}_n + \underline{\boldsymbol{\zeta}}. \tag{2}$$

We can apply the complex to real transformation to equation (2) to get the following equivalent real system

$$\boldsymbol{y} = \sum_{n=1}^{N} \boldsymbol{A}_n \boldsymbol{w}_n + \boldsymbol{\zeta}, \tag{3}$$

where $\boldsymbol{y}, \boldsymbol{\zeta} \in \mathbb{R}^{2K}$, $\boldsymbol{A}_n \in \mathbb{R}^{2K \times 2m_n}$ and $\boldsymbol{w}_n \in \mathbb{R}^{2m_n}$.

To address situations where $\boldsymbol{w}$ is block sparse, we propose to minimize a sparsity criterion. In order to recover $\boldsymbol{w}$ in such situations, we can look for the solution of constrained minimization of its ℓ_0-norm:

$$\min_{\boldsymbol{w}} ||\boldsymbol{w}||_0 \quad s.t \quad ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}||_2^2 \leq \epsilon. \tag{4}$$

Optimization problem (4) is an NP-hard combinatorial problem and its direct minimization is computationally very demanding. An alternative is to use hard thresholding [29]. However, our problem is more complicated due to the block-sparsity constraint. To handle this structured sparsity we propose to consider a constrained $\ell_{2,0}$-norm minimization. The $\ell_{2,0}$-norm of the N-grouped vector $\boldsymbol{w}$ is defined as follows:

$$||\boldsymbol{w}||_{2,0} = \sum_{n=1}^{N} \mathbb{1}_{\mathbb{R}_+^*}(||\boldsymbol{w}_n||_2), \tag{5}$$

where $\mathbb{1}_D(.)$ is the indicator function of subset D: $\mathbb{1}_D(\boldsymbol{x}) = 1$ if $\boldsymbol{x} \in D$ and 0 otherwise. In other terms, $||\boldsymbol{w}||_{2,0}$ is the number of non zero groups in $\boldsymbol{w}$. Based on the above definition and accounting for structured sparsity in model (3) yields

$$\min_{\boldsymbol{w}_{1:N}} ||\boldsymbol{w}||_{2,0} \quad \text{s.t.} \quad ||\boldsymbol{y} - \sum_{n=1}^{N} \boldsymbol{A}_n \boldsymbol{w}_n||_2^2 \leq \epsilon,$$

where the choice of the parameter ϵ influences the degree of sparsity of the solution. Considering the Lagrangian of this problem, it can be transformed into a problem in the form

$$\min_{\boldsymbol{w}_{1:N}} \frac{1}{2}||\boldsymbol{y} - \sum_{n=1}^{N} \boldsymbol{A}_n \boldsymbol{w}_n||_2^2 + \alpha||\boldsymbol{w}||_{2,0}, \tag{6}$$

where the regularization parameter α is related to the choice of ϵ. Before solving (6), we shall consider first the following mean squared error minimization problem:

$$\arg\min_{\boldsymbol{w}_{1:N}} F(\boldsymbol{w}_{1:N}) = \arg\min_{\boldsymbol{w}_{1:N}} \frac{1}{2}||\boldsymbol{y} - \sum_{n=1}^{N} \boldsymbol{A}_n \boldsymbol{w}_n||_2^2. \tag{7}$$

For large-scale problems, the minimum of F can be difficult to compute via direct gradient descent. To overcome this problem, the block coordinate relaxation method can be considered. It is described in the next section.

2.1. Review of block coordinate relaxation

In this section, we briefly recall the Block Coordinate Relaxation (BCR) algorithm [30]. Let us consider the optimization problem (7), where $(\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_N) \in \boldsymbol{X}$ and $\boldsymbol{X}$ is a Cartesian product of convex sets: $\boldsymbol{X} = \boldsymbol{X}_1 \times \boldsymbol{X}_2 \times ... \times \boldsymbol{X}_N$, that is, $\boldsymbol{w}^k \in \boldsymbol{X}_k$. Let us assume that for every $n \in \{1, ..., N\}$, the following optimization problem

$$\min_{\boldsymbol{v} \in \boldsymbol{X}_n} F(\boldsymbol{w}_1, ..., \boldsymbol{w}_{n-1}, \boldsymbol{v}, \boldsymbol{w}_{n+1} ..., \boldsymbol{w}_N) \tag{8}$$

has at least one solution. The BCR algorithm, also known as nonlinear Gauss-Seidel algorithm, generates the next sequence of blocks $(\boldsymbol{w}_1^{k+1}, \boldsymbol{w}_2^{k+1}, ..., \boldsymbol{w}_N^{k+1})$ from the current one $(\boldsymbol{w}_1^k, \boldsymbol{w}_2^k, ..., \boldsymbol{w}_N^k)$, by iteratively

computing

$$\boldsymbol{w}_n^{k+1} = \arg\min_{\boldsymbol{v}\in\boldsymbol{X}_n} F(\boldsymbol{w}_1^{k+1}, ..., \boldsymbol{w}_{n-1}^{k+1}, \boldsymbol{v}, \boldsymbol{w}_{n+1}^{k} ..., \boldsymbol{w}_N^{k}), \quad n = 1 : N. \tag{9}$$

Thus, at each iteration, the cost is minimized with respect to each of the block coordinate vectors $\boldsymbol{w}_n^k$, taken in cyclic order. Then, applied to the minimization of F defined in Eq. (7), the BCR updates write

$$\boldsymbol{w}_n^{k+1} = \arg\min_{\boldsymbol{w}_n} ||\boldsymbol{r}_{-n}^k - \boldsymbol{A}_n \boldsymbol{w}_n||_2^2, \tag{10}$$

where $\boldsymbol{r}_{-n}^k = \boldsymbol{y} - \sum_{p<n} \boldsymbol{A}_p \boldsymbol{w}_p^k - \sum_{p>n} \boldsymbol{A}_p \boldsymbol{w}_p^{k-1}$. Let us note $\boldsymbol{w}^* = [\boldsymbol{w}_1^*, \boldsymbol{w}_2^*, ..., \boldsymbol{w}_N^*]$ the solution of this unconstrained algorithm.
It has been shown that block Gauss-Seidel procedure converges for problem (8) when F is strictly convex and coercive and the $\boldsymbol{X}_n$s are closed convex nonempty subsets of $\mathbb{R}^{k_n}$ ($k_n \in \mathbb{N}^*$) [31].

2.2. $\ell_{2,0}$ penalized least squares problem

Let us remember that our goal is to solve the structured sparsity problem by selecting most significant blocks. As the BCR applied to problem (7) does not achieve nonzero blocks selection, one can add an $\ell_{2,0}$ regularization term which leads to minimize the non-convex objective function introduced in (6):

$$\mathcal{L}(\boldsymbol{w}) := ||\boldsymbol{y} - \sum_{n=1}^{N} \boldsymbol{A}_n \boldsymbol{w}_n||_2^2 + \alpha ||\boldsymbol{w}||_{2,0}, \tag{11}$$

where $\alpha > 0$. We note that due to non-convexity of the $\ell_{2,0}$ regularization term, $\mathcal{L}(.)$ can have several local minimizers. In the following, we consider that $\boldsymbol{A}_n^T \boldsymbol{A}_n = \boldsymbol{I}_{2m_n}$ for $n = 1 : N$. The case where we possibly have $\boldsymbol{A}_n^T \boldsymbol{A}_n \neq \boldsymbol{I}_{2m_n}$ will be considered in Section 5.1.

3. Block-wise optimality conditions for $\ell_{2,0}$ penalized least squares problem

Paralleling the discussion in [29], in this section we adapt results on the hard thresholding procedure to the case of block sparsity. The proofs in this section are quite similar to those in [29] and we supply them here for completeness. We first introduce the proximal block hard thresholding operator and then discuss optimality conditions for $\ell_{2,0}$ penalized mean squared error objective.

Theorem 1 *Consider the vector optimization problem* $\min_{\boldsymbol{v}} \mathcal{L}_z(\boldsymbol{v})$ *with*

$$\mathcal{L}_z(\boldsymbol{v}) = ||\boldsymbol{z} - \boldsymbol{v}||_2^2 + \alpha||\boldsymbol{v}||_2^0, \tag{12}$$

where $||\boldsymbol{v}||_2^0 = 0$ *if* $\boldsymbol{v} = \boldsymbol{0}$ *and* $||\boldsymbol{v}||_2^0 = 1$ *otherwise. Then, the minimum is reached for:*

$$\arg\min_{\boldsymbol{v}} \mathcal{L}_z(\boldsymbol{v}) = \begin{cases} \boldsymbol{0} & \text{if } ||\boldsymbol{z}||_2^2 \leq \alpha \\ \boldsymbol{z} & \text{if } ||\boldsymbol{z}||_2^2 > \alpha \end{cases} \tag{13}$$

Proof $\mathcal{L}_z(\boldsymbol{v})$ *can be rewritten as:*

$$\mathcal{L}_z(\boldsymbol{v}) := \boldsymbol{z}^T\boldsymbol{z} - 2\boldsymbol{z}^T\boldsymbol{v} + \boldsymbol{v}^T\boldsymbol{v} + \alpha||\boldsymbol{v}||_2^0 \tag{14}$$

To derive the minimum of $\mathcal{L}_z(\boldsymbol{v})$*, we distinguish cases* $\boldsymbol{v} = \boldsymbol{0}$ *and* $\boldsymbol{v} \neq \boldsymbol{0}$*. If* $\boldsymbol{v} = \boldsymbol{0}$*, then* $\ell_z(\boldsymbol{v}) = ||\boldsymbol{z}||_2$*. If* $\boldsymbol{v} \neq \boldsymbol{0}$*,* $\mathcal{L}_z(\boldsymbol{v}) = ||\boldsymbol{z}||_2^2 - 2\boldsymbol{z}^T\boldsymbol{v} + ||\boldsymbol{v}||_2^2 + \alpha$ *and the minimum is achieved at* $\boldsymbol{v} = \boldsymbol{z}$ *leading to* $\mathcal{L}_z(\boldsymbol{v}) = \alpha$*. Thus if* $||\boldsymbol{z}||_2^2 \leq \alpha$*, the minimum is reached for* $\boldsymbol{v} = \boldsymbol{0}$ *and if* $||\boldsymbol{z}||_2^2 > \alpha$ *it is reached for* $\boldsymbol{v} = \boldsymbol{z}$*.*

Based on the above theorem, we define the block hard thresholding operator by

$$H_\alpha(\boldsymbol{z}) = \begin{cases} \boldsymbol{0} & \text{if } ||\boldsymbol{z}||_2^2 \leq \alpha \\ \boldsymbol{z} & \text{if } ||\boldsymbol{z}||_2^2 > \alpha. \end{cases} \tag{15}$$

Performing the minimization of $\mathcal{L}(\boldsymbol{w})$ along the n-th block coordinate, amounts to minimize $||\boldsymbol{r}_{-n} - \boldsymbol{A}_n\boldsymbol{w}_n||_2^2 + \alpha||\boldsymbol{w}_n||_2^0$. The following theorem connects the operator H_α and the set of global minimizers of (12).

Theorem 2 *Let $\boldsymbol{z}_n = \boldsymbol{A}_n^T\boldsymbol{r}_{-n}$, with $\boldsymbol{r}_{-n} = \boldsymbol{y} - \sum_{p\neq n}\boldsymbol{A}_p\boldsymbol{w}_p$ and $\mathcal{F} = \cap_{n=1}^N\mathcal{F}_n$ where $\mathcal{F}_n = \{\boldsymbol{w} : \boldsymbol{w}_n = H_\alpha(\boldsymbol{z}_n)\}$. Then, letting $\mathcal{G}$ denote the set of the global optimizers of (11), we have $\mathcal{G} \subset \mathcal{F}$.*

Proof *Let us introduce the following notations: $\boldsymbol{w}_{-n}(\boldsymbol{u})$ will denote the vector $\boldsymbol{w}$ where block $\boldsymbol{w}_n$ has been replaced by $\boldsymbol{u}$. Then, we have*

$$\begin{aligned}
\mathcal{L}(\boldsymbol{w}) &= ||\boldsymbol{y} - \textstyle\sum_{n=1}^N \boldsymbol{A}_n\boldsymbol{w}_n||_2^2 + \alpha||\boldsymbol{w}||_{2,0} \\
&= ||\boldsymbol{r}_{-n} - \boldsymbol{A}_n\boldsymbol{w}_n||_2^2 + \alpha||\boldsymbol{w}_{-n}(\boldsymbol{0})||_{2,0} + \alpha||\boldsymbol{w}_n||_2^0 \\
&= \mathcal{L}(\boldsymbol{w}_{-n}(\boldsymbol{0})) - 2(\boldsymbol{A}_n\boldsymbol{w}_n)^T\boldsymbol{r}_{-n} + (\boldsymbol{A}_n\boldsymbol{w}_n)^T(\boldsymbol{A}_n\boldsymbol{w}_n) \;+\; \alpha||\boldsymbol{w}_n||_2^0 \\
&= \mathcal{L}(\boldsymbol{w}_{-n}(\boldsymbol{0})) - ||\boldsymbol{z}_n||_2^2 + ||\boldsymbol{z}_n||_2^2 - 2\boldsymbol{w}_n^T\boldsymbol{z}_n + ||\boldsymbol{w}_n||_2^2 + \alpha||\boldsymbol{w}_n||_2^0 \\
&= \mathcal{L}(\boldsymbol{w}_{-n}(\boldsymbol{0})) - ||\boldsymbol{z}_n||_2^2 + \mathcal{L}_{\boldsymbol{z}_n}(\boldsymbol{w}_n) \\
&\geq \mathcal{L}(\boldsymbol{w}_{-n}(\boldsymbol{0})) - ||\boldsymbol{z}_n||_2^2 + \min_{\boldsymbol{w}_n}\mathcal{L}_{\boldsymbol{z}_n}(\boldsymbol{w}_n) \\
&\geq \mathcal{L}(\boldsymbol{w}_{-n}(H_\alpha(\boldsymbol{z}_n)))
\end{aligned} \tag{16}$$

The fourth equation holds because $\boldsymbol{A}_n^T\boldsymbol{A}_n = \boldsymbol{I}$. To justify the last inequality, one can consider both cases described in the proof of Theorem 1. For $\boldsymbol{w} = \boldsymbol{w}^* \in \mathcal{G}$, we have $\boldsymbol{z}_n^* = \boldsymbol{A}_n^T\boldsymbol{r}_{-n}^*$ where $\boldsymbol{r}_{-n}^* = \boldsymbol{y} - \sum_{p\neq n}\boldsymbol{A}_p\boldsymbol{w}_p^*$. At local minima of $\mathcal{L}(.)$, inequalities in (16) should become equalities for any $n \in 1:N$. Thus, we get $\mathcal{G} \subset \mathcal{F}$.

As in [29], let us define the sets $\Gamma_0 = \{\, n : \boldsymbol{w}_n = \boldsymbol{0}\}$ and $\Gamma_1 = \{\, n : ||\boldsymbol{w}_n||_2^2 > \alpha\}$. Let $\boldsymbol{w}^* \in \mathcal{F}$ denote a fixed point of the BCR algorithm applied to $\mathcal{L}(\boldsymbol{w})$.

Then, the next theorem supplies necessary conditions that must satisfy any global minimizer of $\mathcal{L}(.)$.

Theorem 3 ***Optimality conditions*** *Let* $\boldsymbol{w}^*$ *denote a global minimizer of* $\mathcal{L}(.)$. *Then,*

$$\begin{cases} ||\boldsymbol{A}_n^T \boldsymbol{r}_{-n}^*||_2^2 \leq \alpha & \text{if } n \in \Gamma_0 \\ ||\boldsymbol{w}_n^*||_2^2 > \alpha & \text{if } n \in \Gamma_1 \\ \boldsymbol{A}_n^T(\boldsymbol{r}_{-n}^* - \boldsymbol{A}_n \boldsymbol{w}_n^*) = \boldsymbol{0} & \text{if } n \in \Gamma_1 \end{cases} \tag{17}$$

Proof *From Theorem 2,* $\boldsymbol{w}_n^* = H_\alpha(\boldsymbol{A}_n^T \boldsymbol{r}_{-n}^*)$. *Thus, from the definition of* H_α, *if* $n \in \Gamma_0$, *we have* $\boldsymbol{w}_n^* = \boldsymbol{A}_n^T \boldsymbol{r}_{-n}^* = \boldsymbol{0}$, *while if* $n \in \Gamma_1$ *we must have* $||H_\alpha(\boldsymbol{A}_n^T \boldsymbol{r}_{-n}^*)||_2^2 = ||\boldsymbol{A}_n^T \boldsymbol{r}_{-n}^*||_2^2 = ||\boldsymbol{w}_n^*||_2^2 < \alpha$. *Now, for* $n \in \Gamma_1$,

$$\begin{aligned} \boldsymbol{w}_n^* &= \boldsymbol{A}_n^T \boldsymbol{r}_{-n}^* \\ &= \boldsymbol{A}_n^T(\boldsymbol{r}_{-n}^* - \boldsymbol{A}_n \boldsymbol{w}_n^*) + \boldsymbol{w}_n^* \quad \text{for} \quad n = 1:N \end{aligned} \tag{18}$$

and we get $\boldsymbol{A}_n^T(\boldsymbol{r}_{-n}^* - \boldsymbol{A}_n \boldsymbol{w}_n^*) = \boldsymbol{0}$.

4. Group Hard Thresholding Algorithm (GHTA)

We propose a novel algorithm, inspired from Block Coordinate Relaxation (BCR), that iteratively performs hard thresholding of blocks $\boldsymbol{w}_n$. The algorithm, named GHTA (Group Hard Thresholding Algorithm) is summarized in the following Algorithm 1:

Algorithm 1 Group Hard Thresholding Algorithm (GHTA) for isometric blocks

Input: $\boldsymbol{A}, \boldsymbol{y}$, α, sizes $\{m_n\}_{n=1:N}$ of the blocks in w.
Initialization: $\boldsymbol{w}^1 = [\boldsymbol{w}_1^1, \boldsymbol{w}_2^1, ..., \boldsymbol{w}_N^1]$, $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}$.
for $k = 1, 2, \ldots$ **do**
    Let $(n_1, \ldots, n_N)$ a random permutation of $(1, \ldots, N)$
    for $n \in (n_1, \ldots, n_N)$ **do**
        $\boldsymbol{z_n^k} = \boldsymbol{A_n^T}(\boldsymbol{r} + \boldsymbol{A_n}\boldsymbol{w_n^k})$
        $\boldsymbol{w_n^{k+1}} = H_\alpha(\boldsymbol{z_n^k})$ (see Eq. (15))
        $\boldsymbol{r} = \boldsymbol{r} - \boldsymbol{A_n}(\boldsymbol{w_n^{k+1}} - \boldsymbol{w_n^k})$.
    end for
end for
Return: $\boldsymbol{w}^* = [\boldsymbol{w}_1^*, \boldsymbol{w}_2^*, ..., \boldsymbol{w}_N^*]$.

4.1. Convergence Analysis

We now prove the convergence of the proposed GHTA algorithm. The following result extends the work in [8] and [9] to the block sparse problems with $\ell_{2,0}$ regularization.

Theorem 4 *The GTHA algorithm converges to a fixed point of* $\mathcal{L}(\boldsymbol{w})$.

Proof *The proof involves several steps.*

1. *First, let us note that* ***GHTA produces a non increasing sequence***

$\{\boldsymbol{w}^k\}_k$. *Indeed, if* $(n_1, \ldots, n_N) = (1, \ldots, N)$

$$\begin{aligned} \mathcal{L}(\boldsymbol{w}_1^k, \boldsymbol{w}_2^k, ..., \boldsymbol{w}_n^k, ..., \boldsymbol{w}_N^k) &\geq \mathcal{L}(\boldsymbol{w}_1^{k+1}, \boldsymbol{w}_2^k, ..., \boldsymbol{w}_n^k, ..., \boldsymbol{w}_N^k) \\ &\geq \mathcal{L}(\boldsymbol{w}_1^{k+1}, \boldsymbol{w}_2^{k+1}, ..., \boldsymbol{w}_n^k, ..., \boldsymbol{w}_N^k) \\ &\vdots \\ &\geq \mathcal{L}(\boldsymbol{w}_1^{k+1}, \boldsymbol{w}_2^{k+1}, ..., \boldsymbol{w}_n^{k+1}, ..., \boldsymbol{w}_N^{k+1}). \end{aligned} \tag{19}$$

Thus, $\mathcal{L}(\boldsymbol{w}^k) \geq \mathcal{L}(\boldsymbol{w}^{k+1})$. *For different ordering of block updates the same conclusion applies as* $\mathcal{L}(.)$ *is non increasing between successive block updates.*

2. ***The sequence*** $\{\boldsymbol{w}^k\}_k$ ***is bounded***. *Indeed, note that, for* $k \geq 1$, $0 \leq \mathcal{L}(\boldsymbol{w}^k) \leq \mathcal{L}(\boldsymbol{w}^0)$ *and* $\{\mathcal{L}(\boldsymbol{w}^k)\}$ *is decreasing, lower bounded and thus it converges. But, on another hand* $\lim_{\|\boldsymbol{w}\| \to \infty} \mathcal{L}(\boldsymbol{w}) = +\infty$ *since* $0 \leq ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}||_2^2 \leq \mathcal{L}(\boldsymbol{w})$. *Thus* $\{\boldsymbol{w}^k\}_k$ *is bounded too.*
3. **Definitions**. *Let* $\mathcal{E}_{ij}$ *denote the* j^{th} *subspace of* $\mathbb{R}^M$ *with at most* i *nonzero blocks:* $i = 0 : N$ *and* $j = 1 : n_i$ *, with* $n_i = \binom{K}{i}$. *Let* $\tilde{\boldsymbol{w}}_{ij} = \arg\min_{\boldsymbol{w} \in \mathcal{E}_{ij}} ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}||_2^2$. *If* $\tilde{\boldsymbol{w}}_{ij} \in \mathcal{E}_{ij}$ *and if it does not lie in any set* $\mathcal{E}_{i'j'}$ *with* $i' < i$, *we denote it by* $\boldsymbol{w}_{ij}^*$, *otherwise we denote it by* $\boldsymbol{w}_{i'j'}^*$ *where* i' *is the smallest possible index such that* $\tilde{\boldsymbol{w}}_{ij} = \tilde{\boldsymbol{w}}_{i'j'} (\in \mathcal{E}_{i'j'} \subset \mathcal{E}_{ij})$.
4. ***The local minima of*** $\mathcal{L}(.)$ ***are the*** $\boldsymbol{w}_{ij}^*\boldsymbol{s}$. *First, note that a local minimum* $\boldsymbol{w}^*$ *belongs to some set* $\mathcal{E}_{ij}$, *where* i *is chosen with minimum possible index among such sets.* $\boldsymbol{w}^*$ *also represents a local minimum of the restriction of* $\mathcal{L}(.)$ *to this set* $\mathcal{E}_{ij}$ *where the penalty term of* $\mathcal{L}(.)$ *remains constant (equal to* αi*). Thus* $\boldsymbol{w}^*$ *must be of the form* $\boldsymbol{w}_{ij}^*$. *Conversely, by construction, for any* $\boldsymbol{w}_{ij}^*$, $\boldsymbol{w}_{ij}^* \notin \mathcal{E}_{i'j'}$ *for* $i' < i$ *and*

$j = 1 : n_{i'}$. *Thus, from continuity of* $\boldsymbol{w} \rightarrow ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}||_2^2$, *there exists a neighbourhood* $\mathcal{B}_{ij}$ *of* $\boldsymbol{w}_{ij}^*$ *such that for* $\boldsymbol{w} \in \mathcal{B}_{ij}$, *we have*

$$\left| ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}||_2^2 - ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}_{ij}^*||_2^2 \right| \leq \frac{\alpha}{2} \tag{20}$$

and $\mathcal{B}_{ij} \cap \mathcal{E}_{i'j'} = \emptyset$ *for* $i^{'} < i$ *and* $j = 1 : n_{i'}$. *As* $\| \boldsymbol{w} \|_{2,0}$ *is constant on* $\mathcal{B}_{ij} \cap \mathcal{E}_{ij}$, *it is clear that for* $\boldsymbol{w} \in \mathcal{B}_{ij} \cap \mathcal{E}_{ij} - \{\boldsymbol{w}_{ij}^*\}$ *and we have* $\mathcal{L}(\boldsymbol{w}) > \mathcal{L}(\boldsymbol{w}_{ij}^*)$ *In addition, for* $\boldsymbol{w} \in \mathcal{B}_{ij} - \mathcal{E}_{ij}$,

$$\begin{aligned} \mathcal{L}(\boldsymbol{w}) &\geq ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}||_2^2 + (i+1)\alpha \\ &\geq ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}_{ij}^*||_2^2 - \tfrac{\alpha}{2} + (i+1)\alpha \\ &\geq \mathcal{L}(\boldsymbol{w}_{ij}^*) + \tfrac{\alpha}{2} \\ &> \mathcal{L}(\boldsymbol{w}_{ij}^*), \end{aligned} \tag{21}$$

where the second inequality hods from (20). Thus, local minima of $\mathcal{L}(.)$ *belong to the set* $\mathcal{S} = \{\boldsymbol{w}_{ij}^*\}_{i,j}$.

5. *As it is bounded in* $\mathbb{R}^M$, **the sequence** $\{\boldsymbol{w}^k\}_k$ **admits at least one accumulation point**. *As the sequence* $\{\mathcal{L}(\boldsymbol{w}^k)\}_k$ *decreases, it is clear that* ***for any two accumulation points*** $\boldsymbol{w}_1^*$ ***and*** $\boldsymbol{w}_2^*$ ***of*** $\{\boldsymbol{w}^k\}_k$ ***we have*** $\mathcal{L}(\boldsymbol{w}_1^*) = \mathcal{L}(\boldsymbol{w}_2^*)$.
6. ***An accumulation point of the algorithm must belong to the set*** $\mathcal{S} = \{\boldsymbol{w}_{ij}^*\}_{i,j}$. *If this was not true, there would exist an accumulation point of* $\{\boldsymbol{w}^k\}_k$, *say* $\boldsymbol{w}^*$, *with* $\boldsymbol{w}^* \notin \mathcal{S}$. *Let* $\mathcal{E}_{ij}$ *the set with smallest index* i *(*$i > 0$*) such that* $\boldsymbol{w}^* \in \mathcal{E}_{ij}$. *Paralleling the discussion in step 4 we define a neighborhood* $\mathcal{B}^*$ *of* $\boldsymbol{w}^*$ *such that* $\mathcal{B}^* \cap \mathcal{E}_{i'j'} = \emptyset$, *for* $i' < i$, *and* $\left| \, ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}||_2^2 - ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}^*||_2^2 \, \right| < \frac{\alpha}{2}$, $\forall \boldsymbol{w} \in \mathcal{B}^*$.
   Let $\{\overline{\boldsymbol{w}}^k\}_k$ *a subsequence of* $\{\boldsymbol{w}^k\}_k$ *in* $\mathcal{B}^*$. *Clearly,* $\forall \boldsymbol{w} \in \mathcal{B}^* - \mathcal{E}_{ij}$,

$$\mathcal{L}(\boldsymbol{w}) \geq ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}||_2^2 + (i+1)\alpha \geq ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}^*||_2^2 - \frac{\alpha}{2} + (i+1)\alpha \geq \mathcal{L}(\boldsymbol{w}^*) + \frac{\alpha}{2},$$

showing thus that the sequence $\{\overline{\boldsymbol{w}}^k\}_k$ *must lie inside* $\mathcal{B}^* \cap \mathcal{E}_{ij}$. *Let* $k_1^{ij}, \ldots, k_i^{ij}$ *the indices of nonzero blocks that define* $\mathcal{E}_{ij}$. *In GHTA, due to random index permutation inside the main loop, infinitely many permutations* $n_1, \ldots, n_N$ *are such that* $(n_1, \ldots, n_i)$ *is a random permutation of* $k_1^{ij}, \ldots, k_i^{ij}$ *among the loops starting from* $\overline{\boldsymbol{w}}_1$, $\overline{\boldsymbol{w}}_2, \ldots$ *and* $\{\overline{\boldsymbol{w}}^k\}_k$ *will be chosen as such a subsequence of elements of* $\{\boldsymbol{w}^k\}_k$.

Let $\boldsymbol{A}^{ij}$ *denote the matrix defined by the blocks of* $\boldsymbol{A}$ *corresponding to non zero blocks of* $\mathcal{E}_{ij}$ *and* $\boldsymbol{A}^{ij} = [\boldsymbol{A}_{k_1^{ij}}, \ldots, \boldsymbol{A}_{k_i^{ij}}]$ *the corresponding block decomposition. Then, starting from* $\overline{\boldsymbol{w}}^k \in \mathcal{E}_{ij}$ *one can see that the first* i *inner loops of GHTA amount to a loop of a block Gauss-Seidel algorithm for the linear system* $[(\boldsymbol{A}^{ij})^T \boldsymbol{A}^{ij}]\boldsymbol{x} = (\boldsymbol{A}^{ij})^T \boldsymbol{y}$, *or equivalently the quadratic problem* $\min_{\boldsymbol{x}} \parallel \boldsymbol{A}^{ij}\boldsymbol{x} - \boldsymbol{y} \parallel_2^2$. *Let* $\tilde{\boldsymbol{w}}^k \in \mathcal{E}_{ij}$ *the vector obtained after these* i *inner loops. Note that the solution* $[(\boldsymbol{A}^{ij})^T \boldsymbol{A}^{ij}]^{-1} \boldsymbol{A}^{ij^T} \boldsymbol{y}$ *of* $\min_{\boldsymbol{x}} \parallel \boldsymbol{A}^{ij}\boldsymbol{x} - \boldsymbol{y} \parallel_2^2$ *is defined by nonzero block entries of* $\boldsymbol{w}_{ij}^*$:

$$\arg\min_{\boldsymbol{x}} \parallel \boldsymbol{A}^{ij}\boldsymbol{x} - \boldsymbol{y} \parallel_2^2 = \begin{bmatrix} [\boldsymbol{w}_{ij}^*]_{k_1^{ij}} \\ \vdots \\ [\boldsymbol{w}_{ij}^*]_{k_i^{ij}} \end{bmatrix}. \tag{22}$$

As the matrix $(\boldsymbol{A}^{ij})^T \boldsymbol{A}^{ij}$ *is symmetric positive,it is known that the randomized block Gauss-Seidel shows linear convergence ([32], Theorem 2) and there exists a fixed* $\lambda \in (0, 1)$ *such that* $\parallel \tilde{\boldsymbol{w}}^k - \boldsymbol{w}_{ij}^* \parallel \leq \lambda \parallel \overline{\boldsymbol{w}}^k - \boldsymbol{w}_{ij}^* \parallel$.

The sequence $\{\overline{\boldsymbol{w}}^k\}_k$ *of* $\mathcal{E}_{ij}$ *converges to* $\boldsymbol{w}^*$. *In the same way, due to continuity of the block Gauss-Seidel transform restricted to* $\mathcal{E}_{ij}$, *the sequence* $\{\tilde{\boldsymbol{w}}^k\}_k$ *of* $\mathcal{E}_{ij}$ *converges to some point* $\tilde{\boldsymbol{w}}^*$ *in* $\mathcal{E}_{ij}$. *As* $\mathcal{L}(\tilde{\boldsymbol{w}}^k) \leq \mathcal{L}(\overline{\boldsymbol{w}}^k)$, *we also have* $\mathcal{L}(\tilde{\boldsymbol{w}}^*) \leq \mathcal{L}(\boldsymbol{w}^*)$. *But, we also have* $\mathcal{L}(\tilde{\boldsymbol{w}}^k) \leq$

$\mathcal{L}(\overline{\boldsymbol{w}}^{k+1})$, *leading to* $\mathcal{L}(\tilde{\boldsymbol{w}}^*) \geq \mathcal{L}(\boldsymbol{w}^*)$. *Thus* $\mathcal{L}(\tilde{\boldsymbol{w}}^*) = \mathcal{L}(\boldsymbol{w}^*)$. *This shows that a loop of block Gauss-Seidel procedure transforms* $\boldsymbol{w}^*$ *into* $\tilde{\boldsymbol{w}}^*$, *without decreasing the corresponding value of* $\mathcal{L}(.)$, *what cannot occur but if the minimum of* $||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}||_2^2$ *over* $\mathcal{E}_{ij}$ *is reached, that is* $\boldsymbol{w}^* = \boldsymbol{w}_{ij}^*$, *what contradicts the initial hypothesis. Thus, we have shown that accumulation points of GHTA belong to* $\mathcal{S}$.

7. **GHTA converges to a point of** $\mathcal{S} = \{\boldsymbol{w}_{ij}^*\}_{i,j}$. *Due to randomness of* $\mathbf{y}$, *for distinct elements* $\boldsymbol{w}_{ij}^*$ *of* $\mathcal{S}$ *the corresponding values* $\mathcal{L}(\boldsymbol{w}_{ij}^*) = \min_{\boldsymbol{w} \in \mathcal{E}_{ij}} ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}||_2^2 + i\alpha$ *are distinct almost surely. We also know from step 1 that* $\{\mathcal{L}(\boldsymbol{w}^k)\}_k$ *decreases to some limit* ℓ *Then, from steps 5 and 6, there is a single accumulation point* $\{\boldsymbol{w}^k\}_k$, *say* $\boldsymbol{w}^*$, *with* $\boldsymbol{w}^* \in \mathcal{S}$, *such that* $\mathcal{L}(\boldsymbol{w}^*) = \ell$. *But as the* $\{\mathcal{L}(\boldsymbol{w}_{ij}^*)\}_{\boldsymbol{w}_{ij}^* \in \mathcal{S}}$ *are all distinct,* $\{\boldsymbol{w}^k\}_k$ *admits a single accumulation point. In addition, from step 2 the sequence* $\{\boldsymbol{w}^k\}_k$ *is bounded. Hence it converges to a point of* $\mathcal{S}$, *what concludes the proof.*

Then, we can prove the following result about the convergence speed of GHTA:

Theorem 5 *The GTHA algorithm converges to its limit* $\boldsymbol{w}^*$ *at linear speed:*

$$\exists \lambda \in (0,1),\ \exists k_0 > 0,\ k > k_0 \Rightarrow ||\boldsymbol{w}^{k+1} - \boldsymbol{w}^*|| \leq \lambda ||\boldsymbol{w}^k - \boldsymbol{w}^*||$$

Proof *As discussed in Theorem 4, from [32], Theorem 2, the randomized block for least squares problems Gauss-Seidel algorithm converges at linear speed. In addition, from the proof of Theorem 4 the sequence* $\{\boldsymbol{w}^k\}_k$ *generated by GHTA converges to some point* $\boldsymbol{w}_{ij}^*$ *and for* k *large enough, say* $k > k_0$,

we have seen that the sequence must lie in a neighborhood $\mathcal{B}^*$ of $\boldsymbol{w}_{ij}^*$ with, in addition, $\boldsymbol{w}^k \in \mathcal{E}_{ij}$. This shows that for $k > k_0$ the blocks that do not contribute to $\mathcal{E}_{ij}$ remain equal to $\mathbf{0}$ and GHTA boils down to the randomized block Gauss-Seidel algorithm in $\mathcal{E}_{ij}$ for the problem (22): $\arg\min_{\boldsymbol{x}} \parallel \boldsymbol{A}^{ij}\boldsymbol{x} - \boldsymbol{y} \parallel_2^2$. *Then the conclusion holds from linear convergence of randomized block Gauss-Seidel algorithm.*

4.2. Connection with ADMM block sparse approaches

As discussed in Section 1 our approach as connections with earlier works and in particular with rPCA (robust PCA). Thus, we are going to discuss our approach in this context. For rPCA we search for a decomposition of a matrix $\boldsymbol{M}$ in the form $\boldsymbol{M} \approx \boldsymbol{L} + \boldsymbol{S}$ where $\boldsymbol{L}$ is low rank and $\boldsymbol{S}$ is sparse. In the case where structured column sparsity is searched, this can lead to the following optimization problem

$$\min_{\boldsymbol{L},\boldsymbol{S}} \parallel \boldsymbol{M} - \boldsymbol{L} - \boldsymbol{S} \parallel_F^2 + \alpha \parallel \boldsymbol{S} \parallel_{2,0} + \beta \parallel \boldsymbol{L} \parallel_\star, \tag{23}$$

where the nuclear norm $\parallel . \parallel_\star$ is a tractable approximation of the matrix rank operator. To this end, we could use an ADMM approach where the proximal gradient step with respect to the proximal norm of $\boldsymbol{L}$ is standard and the minimization step wrt $\boldsymbol{S}$ is obtained via an iteration of GHTA. In fact, GHTA loop appears to be quite close to the proximal gradient of $\parallel . \parallel_{2,0}$.

In [22] the ADMM version of rPCA applies to the following augmented Lagrangian (ALM)

$$\begin{aligned} \mathcal{L}(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{Y}) &= \parallel \boldsymbol{L} \parallel_\star + \parallel \boldsymbol{S} \parallel_{2,1} + Tr(\boldsymbol{Y}^T(\boldsymbol{M} - \boldsymbol{L} - \boldsymbol{S}) + \frac{\rho}{2} \parallel \boldsymbol{M} - \boldsymbol{L} - \boldsymbol{S} \parallel_F^2 \\ &= \parallel \boldsymbol{L} \parallel_\star + \parallel \boldsymbol{S} \parallel_{2,1} + \frac{\rho}{2} \parallel \boldsymbol{M} - \boldsymbol{L} - \boldsymbol{S} + \tfrac{1}{\rho}\boldsymbol{Y} \parallel_F^2 + Ct \end{aligned} \tag{24}$$

Then, letting $\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_M]$ a matrix with SVD $\boldsymbol{B} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, recall the form of proximal operators for $\| \, . \, \|_*$ and $\| \, . \, \|_{2,1}$ are given by

$$\begin{aligned} \arg\min_{\boldsymbol{X}} \lambda \| \, X \, \|_\star + \tfrac{1}{2} \| \, \boldsymbol{B} - \boldsymbol{X} \, \|_F^2 &= \boldsymbol{U} \, \mathcal{S}.(\Sigma, \lambda) \boldsymbol{V}^T \\ &= \mathcal{S}_\star(\boldsymbol{B}, \lambda) \\ \arg\min_{\boldsymbol{X}} \lambda \| \, X \, \|_{2,1} + \| \, \boldsymbol{B} - \boldsymbol{X} \, \|_F^2 &= \boldsymbol{B} \, \mathrm{diag}\{\mathcal{S}(\|\boldsymbol{b}_1\|_2, \lambda), \ldots, \mathcal{S}(\|\boldsymbol{b}_M\|_2, \lambda)\} \\ &= \mathcal{S}_{2,1}(\boldsymbol{B}, \lambda) \end{aligned} \tag{25}$$

where $\mathcal{S}(x, \lambda) = sign(x) \max(|x| - \lambda, 0)$ is the soft thresholding operator and $op.(C)$ indicates entrywise application of operator op to matrix C. Then, alternate proximal gradient update applied to $\boldsymbol{L}$ and $\boldsymbol{S}$ alternatively yields the rPCA-ADMM algorithm:

- Iterate
  - $\boldsymbol{L}_{k+1} = \mathcal{S}_*(\boldsymbol{M} - \boldsymbol{S}_k + \frac{1}{\rho_k}\boldsymbol{Y}_k, \frac{1}{\rho_k})$
  - $\boldsymbol{S}_{k+1} = \mathcal{S}_{2,1}(\boldsymbol{M} - \boldsymbol{L}_{k+1} + \frac{1}{\rho_k}\boldsymbol{Y}_k, \frac{1}{\rho_k})$
  - $\boldsymbol{Y}_{k+1} = \boldsymbol{Y}_k + \rho_k(\boldsymbol{L}_{k+1} + \boldsymbol{S}_{k+1} - \boldsymbol{M})$
  - increase ρ_k,

with convergence condition $\sum_k \rho_k^{-1} = +\infty$ [33]. Now, replacing $\| \, \boldsymbol{S} \, \|_{2,1}$ by $\| \, \boldsymbol{S} \, \|_{2,0}$ in the Lagrangian leads to replace the update of $\boldsymbol{S}_k$ by one loop of GHTA algorithm. With the notations of GHTA, we have $w^{(k)} = vec(\boldsymbol{S}_k)$, $\boldsymbol{A} = \boldsymbol{I}_K$ and $y = vec(\boldsymbol{M} - \boldsymbol{L}_{k+1} + \frac{1}{\rho_k}\boldsymbol{Y}_k)$. Note that this looks quite similar

to applying directly proximal gradient to $\| \cdot \|_{2,0}$:

$$\begin{aligned} &\arg\min_{\boldsymbol{X}} \lambda \parallel \boldsymbol{X} \parallel_{2,0} + \tfrac{1}{2} \parallel \boldsymbol{B} - \boldsymbol{X} \parallel_F^2 \\ &= \arg\min_{\boldsymbol{X}} \lambda \textstyle\sum_{n=1:N} \mathbb{1}_{\|\boldsymbol{X}_n\|>0} + \tfrac{1}{2} \parallel \boldsymbol{B} - X \parallel_F^2 \\ &= \left[\hat{\boldsymbol{X}}_1, \ldots, \hat{\boldsymbol{X}}_N\right], \end{aligned} \tag{26}$$

with $\hat{\boldsymbol{X}}_n = \boldsymbol{B}_n \times \mathbb{1}_{\|\boldsymbol{B}_n\|>\lambda}$. But instead of updating columns of $\boldsymbol{X}$ independently, in GHTA we consider a Gauss-Seidel like approach and each column update benefits from possible earlier update of other columns. Alternatively, the GHTA-GM approach discussed in the next section where exact block minimization is replaced by gradient descent can also be considered.

5. Extensions

5.1. Extension to non-isometric block matrices

Extension of GHTA to the case of full rank block matrices that are not isometric, that is, $\boldsymbol{A}_n^T \boldsymbol{A}_n \neq \boldsymbol{I}_{2m_n}$ is rather straightforward. Indeed, letting $\boldsymbol{A}_n = \boldsymbol{Q}_n \boldsymbol{R}_n$ denote the QR decomposition of $\boldsymbol{A}_n$ and $\boldsymbol{v}_n = \boldsymbol{R}_n \boldsymbol{w}_n$, where $\boldsymbol{w}_n$ is the n-th block of vector variable $\boldsymbol{w}$, the objective (11) rewrites

$$\mathcal{L}_{QR}(\boldsymbol{v}) = ||\boldsymbol{y} - \sum_{n=1}^{N} \boldsymbol{Q}_n \boldsymbol{v}_n||_2^2 + \alpha ||\boldsymbol{v}||_{2,0}. \tag{27}$$

Note that $\mathcal{L}_{QR}(\boldsymbol{v})$ can be minimized applying GHTA as described in algorithm (1) and that

$$\mathcal{L}_{QR}([\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N]) = \mathcal{L}([\boldsymbol{R}_1 \boldsymbol{w}_n, \ldots, \boldsymbol{R}_N \boldsymbol{w}_N]) \tag{28}$$

since $||\boldsymbol{v}_n||_2 = ||\boldsymbol{R}_n \boldsymbol{w}_n||_2$ and, as $\boldsymbol{R}_n$ is invertible, the block $\boldsymbol{v}_n$ is nonzero if and only if the block $\boldsymbol{w}_n$ is nonzero too. Then we derive algorithm QR-GHTA described in Algorithm 2. It is clear that QR-GHTA shares the same convergence guarantee as GHTA.

Algorithm 2 Generalized GHTA for non isometric blocks: QR-GHTA

1: Input: $\boldsymbol{A}, \boldsymbol{y}$, α, sizes $\{m_n\}_{n=1:N}$ of the blocks in v and w.

2: Initialization: $\boldsymbol{v}^1 = [\boldsymbol{v}_1^1, \boldsymbol{v}_2^1, \ldots, \boldsymbol{v}_N^1]$

3: Compute QR decompositions $\boldsymbol{A}_n = \boldsymbol{Q}_n \boldsymbol{R}_n$, $1 : N$

4: Apply GHTA (or GHTA-GM described below) to

$$\mathcal{L}_{QR}(\boldsymbol{v}) = ||\boldsymbol{y} - \sum_{n=1}^{N} \boldsymbol{Q}_n \boldsymbol{v}_n||_2^2 + \alpha ||\boldsymbol{v}||_{2,0}$$

$\rightarrow [\boldsymbol{v}_1^*, \boldsymbol{v}_2^*, \ldots, \boldsymbol{v}_N^*]$

5: Return: $\boldsymbol{w}^* = \left[\boldsymbol{R}_1^{-1}\boldsymbol{v}_1^*, \boldsymbol{R}_2^{-1}\boldsymbol{v}_2^*, ..., \boldsymbol{R}_N^{-1}\boldsymbol{v}_N^*\right]$.

5.2. Acceleration via gradient and momentum

It has been shown in the literature that majorization-minimization approaches combined with Nesterov momentum is an efficient approach for gradient based optimization. In particular, as discussed in [14], applying such approach as an alternative to exact iterative block optimization can prove more efficient, in terms of convergence speed. In [14], the authors consider a criterion intended to ensure both block sparsity and intra-block sparsity, leading to block update criterion in the form

$$\frac{1}{2n} \parallel \boldsymbol{r_{-n}} - \boldsymbol{A_n w_n} \parallel_2^2 + (1-\mu)\alpha \parallel \boldsymbol{w_n} \parallel_2 + \mu\alpha \parallel \boldsymbol{w_n} \parallel_1 \qquad (29)$$

In this work, we are not considering sparsity inside blocks but only among blocks. Thus, we assume that $\mu = 0$. Then, adapting the minimization of (29) proposed in [14] to the case where $\parallel \boldsymbol{w_n} \parallel_2$ is replaced by $\parallel \boldsymbol{w_n} \parallel_2^0$ yields algorithm GHTA-GM (GHTA with gradient and momentum) described in Algorithm 3. The description is given for isometric block matrices $\boldsymbol{A_n}$, but

of course it can be used with QR-GHTA where this assumption is relaxed. In the algorithm, $\beta \in (0,1)$ is a parameter for backtracking iterations [34], and K is the dimension of $\underline{\boldsymbol{y}}$ (as specified in section 2).

Algorithm 3 GHTA with gradient and momentum (GHTA-GM) for isometric blocks

Input: $\boldsymbol{A}, \boldsymbol{y}, \alpha$, β, sizes $\{m_n\}_{n=1:N}$ of the blocks in w.
Initialization: $\boldsymbol{w}^1 = [\boldsymbol{w}_1^1, \boldsymbol{w}_2^1, ..., \boldsymbol{w}_N^1]$, $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}^1$.
for $k = 1, 2, \ldots$ **do**
    Let $(n_1, \ldots, n_N)$ a random permutation of $(1, \ldots, N)$
    for $n \in (n_1, \ldots, n_N)$ **do**
        for $l = 1, 2, \ldots$ **do**
            $t = 1$
            do
                $t = \beta t$
                $\tilde{\boldsymbol{w}}_{\boldsymbol{n}}^{k+1} = H_{\alpha t}\left(\boldsymbol{w}_{\boldsymbol{n}}^k - t\left[\frac{-1}{2K}\boldsymbol{A}_{\boldsymbol{n}}(\boldsymbol{r} + \boldsymbol{A}_{\boldsymbol{n}}^T\boldsymbol{w}_{\boldsymbol{n}}^k)\right]\right)$
                $\tilde{\boldsymbol{r}} = \boldsymbol{r} - \boldsymbol{A}_{\boldsymbol{n}}(\tilde{\boldsymbol{w}}_{\boldsymbol{n}}^{k+1} - \boldsymbol{w}_{\boldsymbol{n}}^k)$
            while $\frac{1}{2} \parallel \tilde{\boldsymbol{r}} \parallel^2 > \frac{1}{2} \parallel \boldsymbol{r} \parallel^2 - \boldsymbol{r}^T\boldsymbol{A}_{\boldsymbol{n}}(\tilde{\boldsymbol{w}}_{\boldsymbol{n}}^{k+1} - \boldsymbol{w}_{\boldsymbol{n}}^k) + \frac{K}{t} \parallel \tilde{\boldsymbol{w}}_{\boldsymbol{n}}^{k+1} - \boldsymbol{w}_{\boldsymbol{n}}^k \parallel_2^2$
            $\boldsymbol{w}_{\boldsymbol{n}}^{k+1} = \boldsymbol{w}_{\boldsymbol{n}}^k + \frac{l}{l+3}(\tilde{\boldsymbol{w}}_{\boldsymbol{n}}^{k+1} - \boldsymbol{w}_{\boldsymbol{n}}^k)$
            $\boldsymbol{r} = \tilde{\boldsymbol{r}} + \boldsymbol{A}_n(\tilde{\boldsymbol{w}}_{\boldsymbol{n}}^{k+1} - \boldsymbol{w}_n^{k+1})$
        end for
    end for
end for
Return: $\boldsymbol{w}^* = [\boldsymbol{w}_1^*, \boldsymbol{w}_2^*, ..., \boldsymbol{w}_N^*]$.

6. Simulation results

With a view to assess performance of the proposed method, we compare performance obtained when minimizing

$$F_{q2}(\boldsymbol{w}) = \frac{1}{2}||\boldsymbol{y} - \sum_{n=1}^{N} \boldsymbol{A}_n \boldsymbol{w}_n||_2^2 + \alpha||\boldsymbol{w}||_{2,q}\,, \tag{30}$$

using different approaches. For $q = 0$, we used our proposed GHTA or GHTA-GM algorithms. For $q = 1$, we considered group Lasso algorithm. We also considered the IRLS approach in [28]. In [28], $q = 0.1$ is the minimum tested value and we consider it for comparison with GHTA.

Then we discuss the benefits of GOMP against such iterative strategies and show that despite its benefits it can fail in some situations such as for the array processing problem that we address next.

6.1. Comparison with group Lasso and IRLS

For the simulation, we considered a case where data have length $K = 400$ and $\boldsymbol{w}$ has length 180 and contains 18 blocks of length 10. Among the blocks 6 of them are nonzero. These parameters are similar to those chosen in [14]. The matrix $\boldsymbol{A}$ is a random isometric matrix. Simulation results have been obtained from 100 Monte Carlo simulations.

Figure 1 presents the evolution of NMSE (Normalized Mean Squared Error) at convergence, where NMSE is defined by

$$\text{NMSE} = \frac{\| \boldsymbol{w}_{\text{true}} - \boldsymbol{w}_{\text{estimated}} \|_2^2}{\| \boldsymbol{w}_{\text{true}} \|_2^2} \tag{31}$$

as a function of the regularization parameter α, as it increases from 0, that is, from the MMSE solution (MMSE $= \min_{\boldsymbol{w}} \frac{1}{2}||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}||_2^2$)).

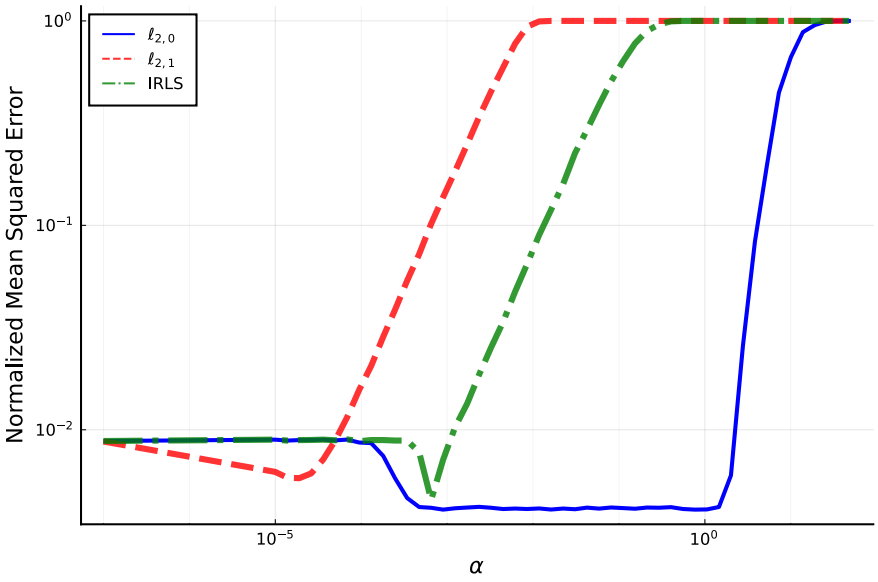

Figure 1: Normalized mean squared error versus regularization parameter α for SNR=40dB

Figure 2 presents the detection performance both in terms of number of detected blocks and in terms of simultaneous correct detection of zero and nonzero blocks for all blocks in $\boldsymbol{w}$.

From these figures it clearly appears that GHTA does not seem to suffer from possible convergence to local minima. It is also interesting to notice that GHTA achieves lower NMSE and shows better robustness against the choice of α than group Lasso or IRLS. In particular, Figure 2 shows that zero and nonzero blocks are well detected in range about $[10^{-3}, 1]$ with GHTA but only about $[10^{-4}, 10^{-2}]$ with group Lasso and $[10^{-3}, 10^{-1}]$ with IRLS. In addition, over these respective intervals the NMSE remains small and constant for GHTA while it increases with α in group Lasso or IRLS. This illustrates the presence of the bias introduced by the $\ell_{2,1}$ or even the $\ell_{2,0.1}$

penalization terms, that increases with α, while no bias is introduced with the $\ell_{2,0}$ penalization in respective intervals of α where desired sparsity indices are recovered by the algorithm.

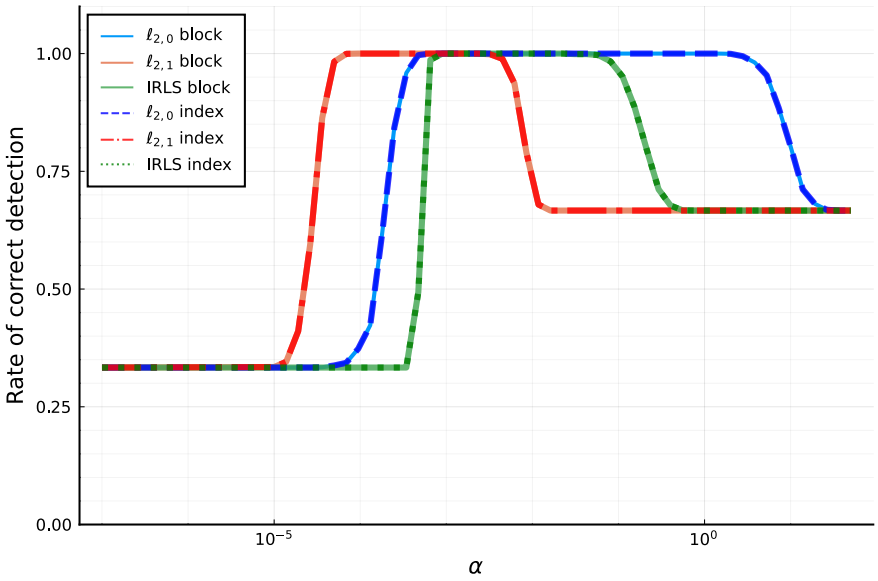

Figure 2: Rate of correct detection versus regularization parameter α for SNR=40dB

Figures 3 and 4 show NMSE and block detection performance at lower SNR (SNR=10dB). The intervals where correct detection of zero and nonzero blocks is achieved is reduced compared to the results at high SNR. However, the same conclusions hold for comparison of the effects of $\ell_{2,1}$, $\ell_{2,0.1}$ and $\ell_{2,0}$ penalties.

Figure 5 shows the influence of SNR on NMSE performance. α is optimized for each method and each SNR. The bias introduced by $\ell_{2,1}$ and even $\ell_{2,0.1}$ penalties clearly appears and limits performance of group Lasso and IRLS while performance increases regularly with SNR for $\ell_{2,0}$ penalty.

In our simulations, we have checked that GHTA and GHTA-GM exhibit

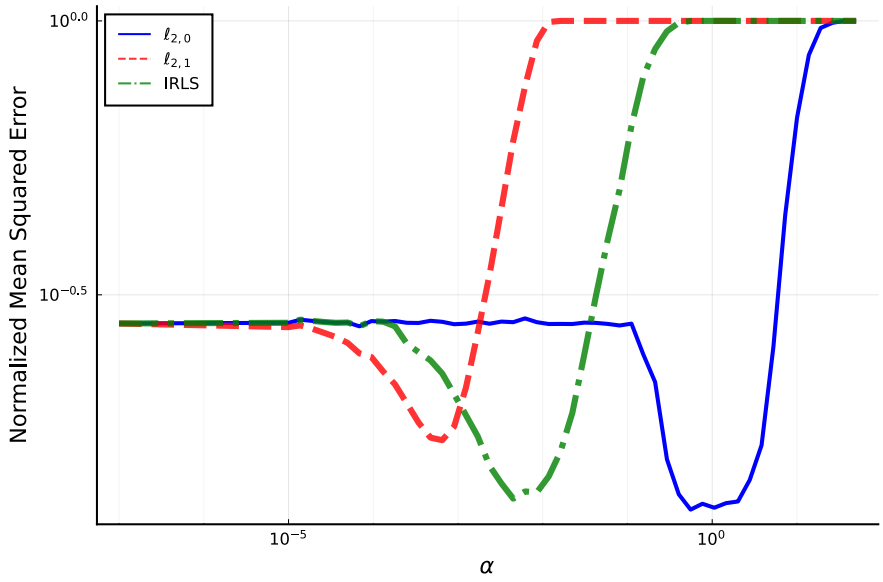

Figure 3: Normalized mean squared error versus regularization parameter α for SNR=10dB

similar performance in terms of NMSE or detection rate. The difference between both algorithms lies in the different convergence speed that can be observed. Indeed, Figures 6 and 7 show that GHTA-GM achieves faster convergence than GHTA both at high and low SNR. Optimum values have been chosen from parameters α. As expected from the discussion above GHTA and GHTA-GM achieve same asymptotic error, while that of group Lasso remains higher.

For a comparison with $\ell_{2,0.1}$ approach, note that only few iterations are involved in IRLS. However, for $\boldsymbol{A} \in \mathrm{R}^{K\times L}$, GHTA or group Lasso involve $\mathcal{O}(N_{outer}KL)$ operations where N_{outer} is the number of outer loops, while IRLS requires $\mathcal{O}(N_{IRLS}L^3)$ operations where N_{IRLS} corresponds to the number of matrix inversions in IRLS algorithm. In [28] the maximum number

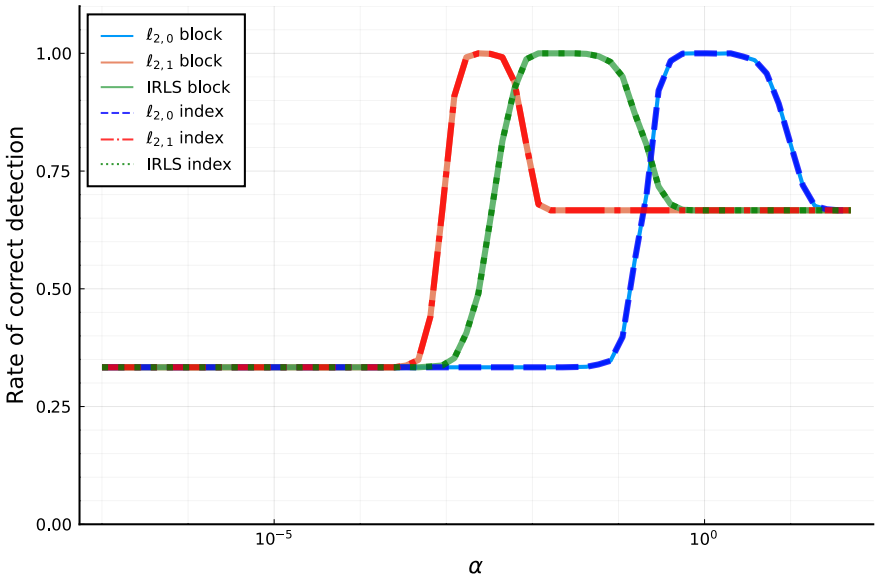

Figure 4: Rate of correct detection versus regularization parameter α for SNR=10dB

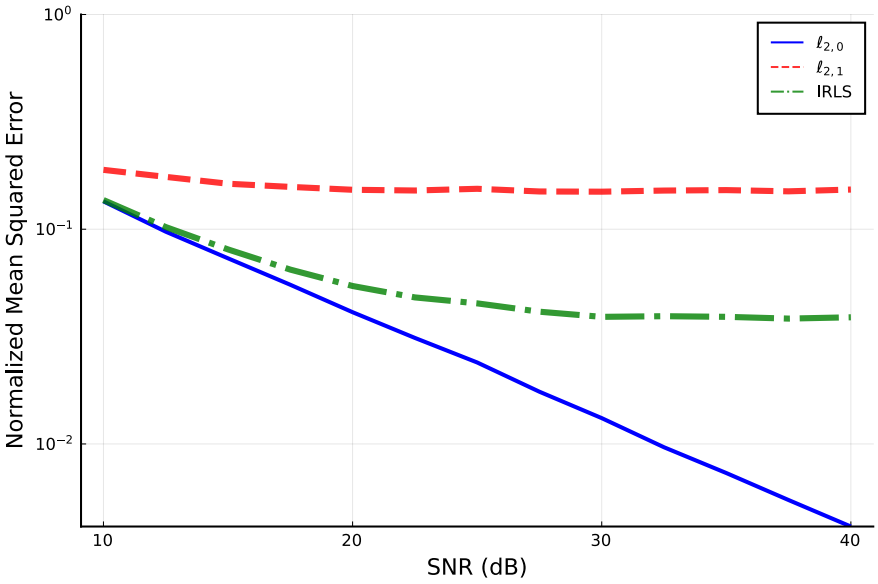

Figure 5: Normalized mean squared error versus SNR

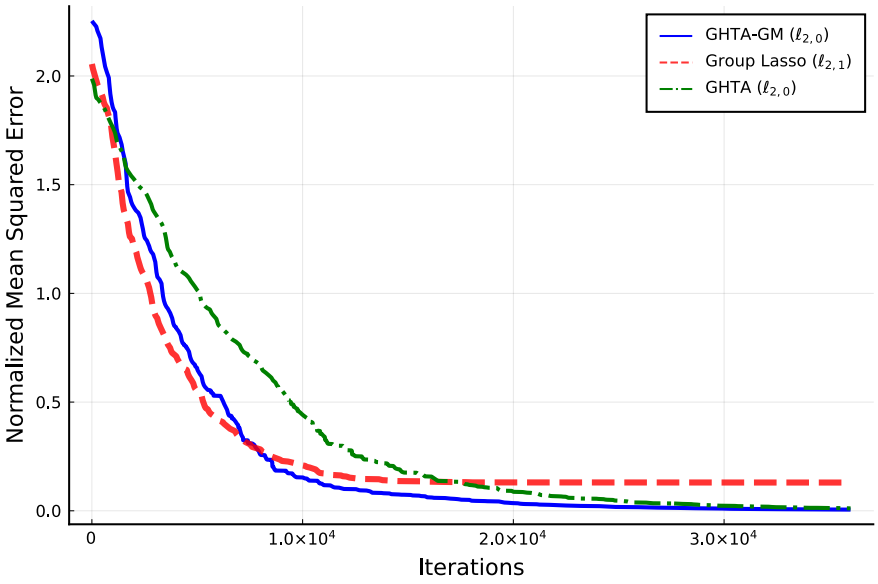

Figure 6: Normalized mean squared error versus inner loop iterations for SNR=40dB

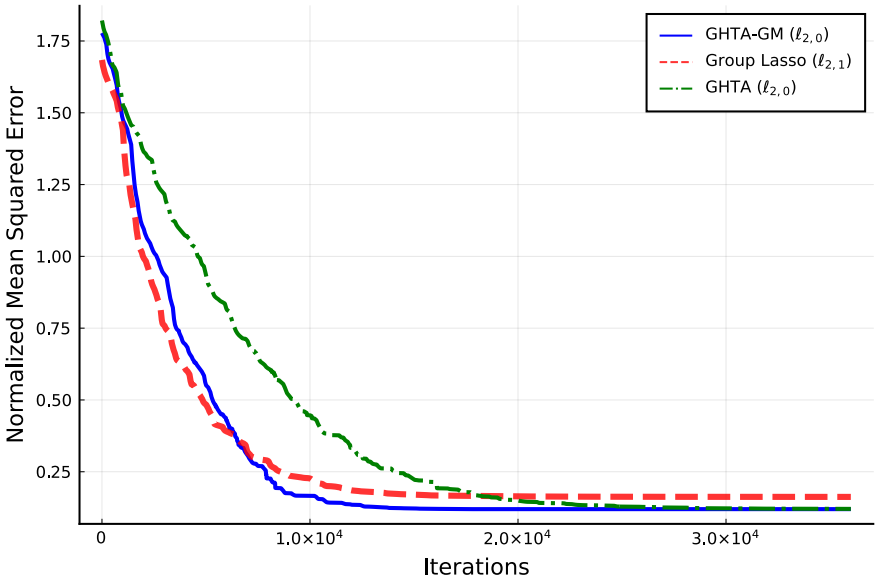

Figure 7: Normalized mean squared error versus inner loop iterations for SNR=10dB

Algorithm / (K, L)	GHTA ($\ell_{2,0}$)		IRLS		GOMP	
	10dB	40dB	10dB	40dB	10dB	40dB
(200,90)	21.6	0.2	128.1	0.2	0.2	0.2
(400,180)	107.5	0.6	341.2	0.6	1.3	1.3
(800,360)	306.7	2.3	1276.0	2.3	11.3	11.1

Table 1: Execution time in (ms), for 10dB and 40dB respectively, to achieve NMSE=0.1 for GHTA, IRLS and GOMP algorithms, and several values for (K, L).

of iterations is set to $N_{IRLS}^{max} = 500$ (so that $N_{IRLS} \leq N_{IRLS}^{max}$), and we set the maximum number of iterations for GHTA and group Lasso so that the total maximum number of operations is about the same: $N_{outer} \leq N_{outer}^{max} = N_{IRLS}^{max} L^2/K$. In practice, for optimized α, in our example we observed rather similar complexity for all algorithms.

As far as speed of execution is concerned we give elapsed times in Table 1 for a few values of (K, L) where the time needed to achieve NMSE equal to 0.1 is presented. Results are given for SNR=10dB and SNR=40dB. As Lasso is not able to reach this precision due to bias introduced by the $\ell_{2,1}$ penalty term we considered only GHTA and IRLS. It appears that GHTA is faster than IRLS, and depending on configurations it can be up to six times faster. To complete comparison, we added the GOMP algorithm [15], the group extension of OMP (Orthogonal Matching Pursuit) [7]. This is a very fast greedy algorithm and for some configurations it can achieve targeted NMSE about 100 faster than GHTA.

If we apply GOMP with the examples presented in this subsection we find similar performance in terms of block detection and normalized mean

squared error as those obtained by GHTA on Figures 1 to 4 and we did not presented these curves on the figures for clarity.

Thus, GHTA or IRLS may appear useless since they are computationally demanding because they often require many iterations. However, we are going to show that there are situations were the GOMP always converges to particular local optima that are not of interest for the problem at hand unlike GHTA.

6.2. Comparison with Group Orthogonal Matching Pursuit (GOMP)

In this section we compare GHTA with GOMP algorithm [15] on a particular problem related to sensor selection for array processing. First, we show how it can be described by a linear model where block sparsity is searched for.

For the sake of simplicity, we consider a standard configuration where we assume a linear array of sensors. We assume that signals of interest have wavelength λ and that K sensors are positioned on a grid of N regularly spaced virtual positions, with spacing $\lambda/2$. To select the positions of sensors on the grid, we look for optimal beamformers in a range of regularly spaced directions $\theta_i \in \{-\theta_{\max}, \ldots, \theta_{\max}\}$ for $i = 1 : M$. The beamformers are designed so that for the i-th beamformer we want to observe unit gain in some direction θ_i and zero gain in other directions. Letting $\phi_i = \pi \sin(\theta_i)$ denote the spatial pulsation associated with direction θ_i (directions are measured from the normal to the array direction), we define the matrix $\boldsymbol{D} \in \mathbb{C}^{M \times N}$ with entries $\boldsymbol{D}_{ab} = e^{i(b-1)\phi_a}$. Letting $\boldsymbol{D}_a$ denote column a of $\boldsymbol{D}$ and $\boldsymbol{y} = vec(\boldsymbol{I}_M)$, where $vec(.)$ is the vectorization operator, the linear model associated with

the problem writes

$$\boldsymbol{y} = [\boldsymbol{I}_M \otimes \boldsymbol{D}_1, \ldots, \boldsymbol{I}_M \otimes \boldsymbol{D}_N]\boldsymbol{w}. \tag{32}$$

where $\otimes$ is the Kronecker product and $\boldsymbol{w} \in \mathbb{C}^{NM}$ consists of N blocks, each representing a virtual sensor and the m-th entry of block n represents the beamformer coefficient on sensor n when scanning angle θ_m. Note that we could refine this model and consider in particular a set of tested directions for each beamformer that differs from the set of directions of interest for the different beamformers but this is beyond the scope of the paper and the proposed approach avoids introducing many notations.

Note also that the matrix $\boldsymbol{A} = [\boldsymbol{I}_M \otimes \boldsymbol{D}_1, \ldots, \boldsymbol{I}_M \otimes \boldsymbol{D}_N]$ can be very big: for instance considering 100 directions and 100 possible sensor positions yields a matrix $\boldsymbol{A}$ with 10^8 coefficients. Fortunately the matrix is sparse and matrix operations involved in GHTA or GOMP boil down to simpler operations involving dense matrices and vectors thanks to the properties of composition among Kronecker products and matrix/vector multiplications. For instance, at the end of its k-th iteration GOMP involves computation of the k currently active blocks of $\boldsymbol{w}$: $\boldsymbol{w}_{G_k} = (\boldsymbol{A}_{G_k}^H \boldsymbol{A}_{G_k})^{-1} \boldsymbol{A}_{G_k}^H \boldsymbol{y}$, where $G_k = \{g_1, \ldots, g_k\}$ is the group of selected indices and $\boldsymbol{A}_{G_k} = [\boldsymbol{I}_M \otimes \boldsymbol{D}_{g_1}, \ldots, \boldsymbol{I}_M \otimes \boldsymbol{D}_{g_k}]$ is the corresponding block sub-matrix of $\boldsymbol{A}$. Then, for $\boldsymbol{y} = vec(\boldsymbol{I}_M)$, $\boldsymbol{w}_{G_k}$ rewrites in a simple compact form: $\boldsymbol{w}_{G_k} = vec\left((\boldsymbol{D}_{G_k}^H (\boldsymbol{D}_{G_k}^H \boldsymbol{D}_{G_k})^{-1}\right)^*$, with $\boldsymbol{D}_{G_k} = [\boldsymbol{D}_{g_1}, \ldots, \boldsymbol{D}_{g_k}]$.

As an example we consider an array with sensors that have aperture $[-45°, 45°]$. For each angle $\theta_i = -45° + i$, with $i = 0° : 90°$, we consider a beamformer for this direction. We assume 64 regularly sampled positions where we want to put 32 sensors. Figure 8 shows the locations obtained

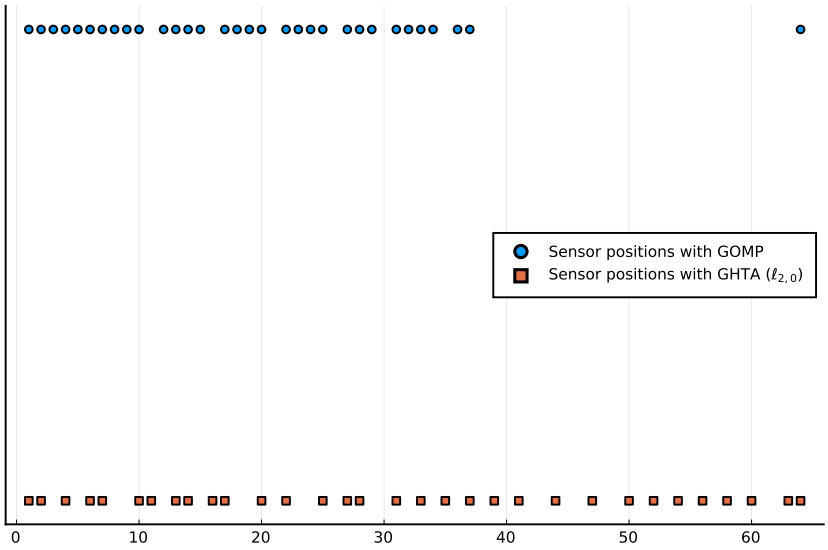

Figure 8: Sensor positions with GOMP and GHTA: 32 positions are selected for sensor locations among 64 possible regularly spaced positions.

for the arrays of sensors both for GOMP and GHTA. Here we have forced end locations to belong to the selected positions of sensors. We observe that GOMP yields a very compact antenna with most sensors at one end of the set of possible positions. As a result, the main lobe of the corresponding beamformers is about the same as for a compact antenna with 32 sensors with spacing $\lambda/2$, as shown in figure 9. On the contrary, we see on figure 10 that GHTA achieves more spread sensor locations, resulting in a mainlobe width similar to that of a full antenna with 64 sensors. Figure 11 shows a superposition of 0° beamformers for GOMP and GHTA. Note also that for GHTA, the norm of the error $\| \boldsymbol{y} - \boldsymbol{A}\boldsymbol{w} \|_2$ is generally lower than for GOMP by a factor about 20%. Of course, there is a price to pay for the high resolution of GHTA beamformers: a higher sidelobe level. This is visible for

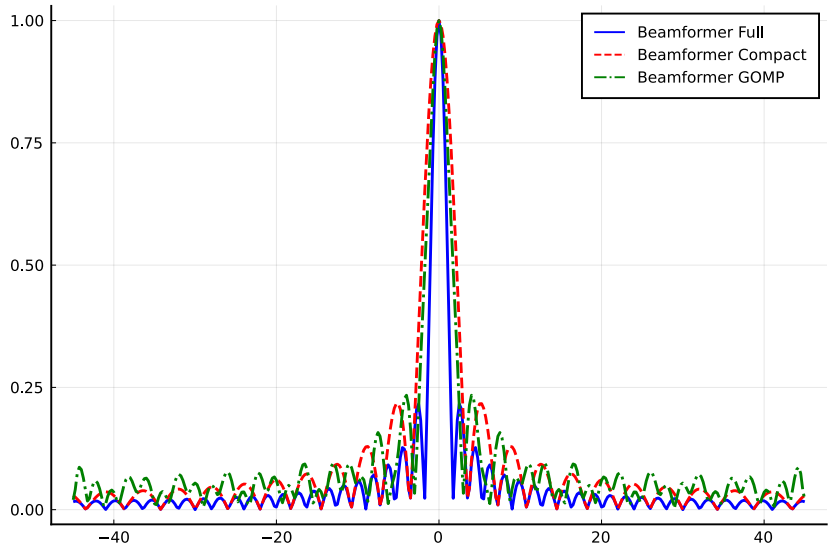

Figure 9: Optimized beamformer with GOMP for direction $\theta = 0°$ (green). Blue and red curves are obtained with the conventional beamformers for 64 and 32 regularly spaced sensors.

the 0° beamformer in Figure 10 and even more on some other beamformers as the one at about 20° in Figure 12 (a symetric figure is obtained for −20°). With these beamformers applications such as source localization can be considered, using for instance OMP based techniques (see e.g. [35]). We do develop on this subject furher here, but we believe that extensions of the approach proposed in this section to related subjects such as beamforming for wideband array processing would be worth being considered.

7. Conclusion

In this work we have illustrated the possible benefits of using $\ell_{2,0}$ penalty terms in linear regression techniques when looking for block sparse solutions compared to popular $\ell_{2,1}$ penalization. We have proved the convergence of

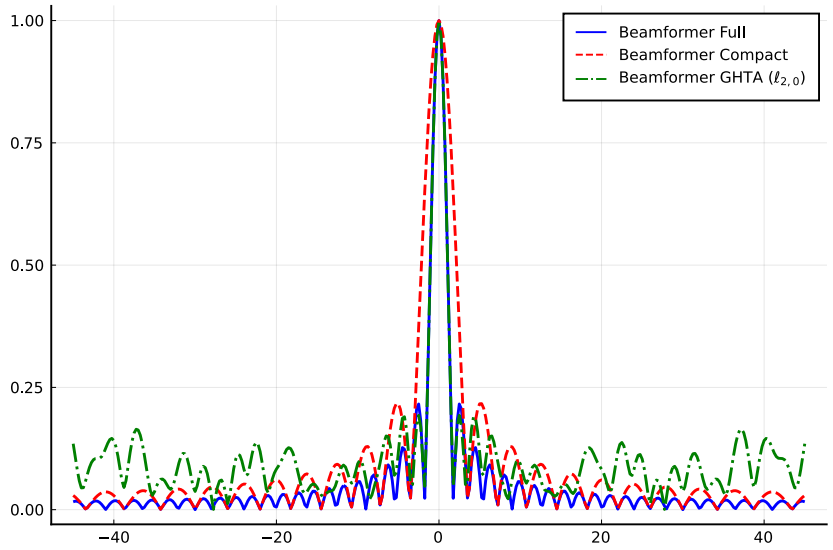

Figure 10: Optimized beamformer with GHTA for direction $\theta = 0°$ (green). Blue and red curves are obtained with the conventional beamformers for 64 and 32 regularly spaced sensors.

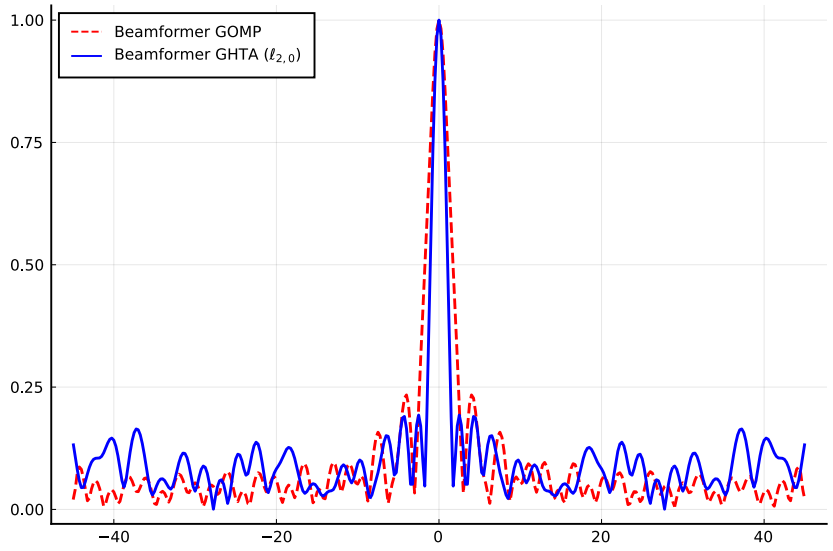

Figure 11: GOMP (red) and GHTA (blue) beamformers of Figures 9 and 10.

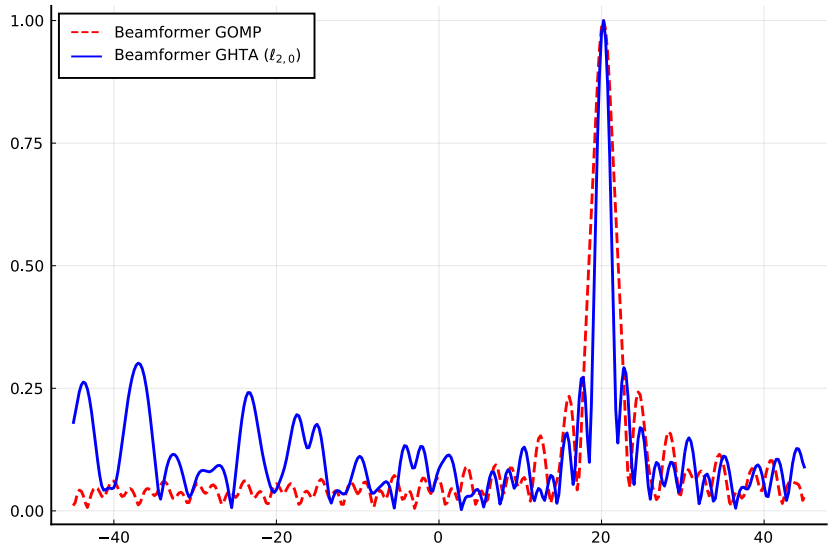

Figure 12: GOMP (red) and GHTA (blue) beamformers of Figures 9 and 10 at about 20°.

the block cyclic minimization for the proposed $\ell_{2,0}$ penalized criterion. We also checked the good robustness of the proposed GHTA approach against the choice of regularization parameters in terms of block detection performance and the absence of asymptotic bias it achieves. We also have shown how powerful and simpler can be the greedy GOMP approach that often achieves similar results but is much faster than iterative procedures. Nevertheless, we have illustrated the benefit of GHTA in a situation were GOMP is only able to supply an uninteresting local minimum.

References

[1] D. L. Donoho, Compressed sensing, IEEE Transactions on Information Theory 52 (4) (2006) 1289–1306.

[2] M. F. Duarte, Y. C. Eldar, Structured compressed sensing: From theory

to applications, IEEE Transactions on Signal Processing 59 (9) (2011) 4053–4085.

[3] Shaobing Chen, D. Donoho, Basis pursuit, in: Proceedings of 28th Asilomar Conference on Signals, Systems and Computers, Vol. 1, 1994, pp. 41–44.

[4] I. F. Gorodnitsky, J. S. George, B. D. Rao, Neuromagnetic source imaging with focuss: a recursive weighted minimum norm algorithm, Electroencephalography and Clinical Neurophysiology 95 (4) (1995) 231–251.

[5] E. J. Candès, M. B. Wakin, S. P. Boyd, Enhancing sparsity by reweighted ℓ_1-minimization, Journal of Fourier Analysis and Applications 14 (5) (2008) 877–905.

[6] S. G. Mallat, Zhifeng Zhang, Matching pursuits with time-frequency dictionaries, IEEE Transactions on Signal Processing 41 (12) (1993) 3397–3415.

[7] J. Wang, S. Kwon, B. Shim, Generalized orthogonal matching pursuit, IEEE Transactions on Signal Processing 60 (12) (2012) 6202–6216.

[8] G. Marjanovic, V. Solo, ℓ_q sparsity penalized linear regression with cyclic descent, IEEE Transactions on Signal Processing 62 (6) (2014) 1464–1475.

[9] J. Zeng, Z. Peng, S. Lin, Z. Xu, A cyclic coordinate descent algorithm for lq regularization (2014). arXiv:1408.0578.

[10] Z. Zhang, Y. Xu, J. Yang, X. Li, D. Zhang, A survey of sparse representation: Algorithms and applications, IEEE Access 3 (2015) 490–530.

[11] E. Crespo Marques, N. Maciel, L. Naviner, H. Cai, J. Yang, A review of sparse recovery algorithms, IEEE Access 7 (2019) 1300–1322.

[12] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 210–227.

[13] E. Elhamifar, R. Vidal, Sparse subspace clustering: Algorithm, theory, and applications, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (11) (2013) 2765–2781.

[14] N. Simon, J. Friedman, T. Hastie, R. Tibshirani, A sparse-group lasso, Journal of Computational and Graphical Statistics 22 (2) (2013) 231–245.

[15] G. Swirszcz, N. Abe, A. C. Lozano, Grouped orthogonal matching pursuit for variable selection and prediction, in: Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, A. Culotta (Eds.), Advances in Neural Information Processing Systems, Vol. 22, Curran Associates, Inc., 2009.

[16] M. Messai, A. Aïssa-El-Bey, K. Amis, F. Guilloud, Iteratively reweighted two-stage lasso for block-sparse signal recovery under finite-alphabet constraints, Signal Processing 157 (2019) 73–77.

[17] C. Zhao, X. Mao, T. Zhang, C. Yu, A likelihood-based hyperparameter-free algorithm for robust block-sparse recovery, Signal Processing 161 (2019) 89–100.

[18] W. Wang, H. Zhao, A novel block-sparse proportionate NLMS algorithm based on the $\ell_{2,0}$ norm, Signal Processing 176 (2020) 107671.

[19] E. J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis?, J. ACM 58 (3) (jun 2011).

[20] Z. Zhou, X. Li, J. Wright, E. J. Candès, Y. Ma, Stable principal component pursuit, CoRR abs/1001.2363 (2010). arXiv:1001.2363.
URL `http://arxiv.org/abs/1001.2363`

[21] G. Tang, A. Nehorai, Robust principal component analysis based on low-rank and block-sparse matrix decomposition, in: 2011 45th Annual Conference on Information Sciences and Systems, 2011, pp. 1–5.

[22] H. Xu, C. Caramanis, S. Sanghavi, Robust PCA via outlier pursuit, IEEE Transactions on Information Theory 58 (5) (2012) 3047–3064.

[23] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, IEEE Trans. on Pattern Analysis and Machine Intelligence 35 (2013).

[24] P. Netrapalli, U. N. Niranjan, S. Sanghavi, A. Anandkumar, P. Jain, Non-convex robust PCA, Advances in Neural Information Processing Systems abs/1410.7660 (2014).

[25] F. Wen, R. Ying, P. Liu, T.-K. Truong, Nonconvex regularized robust PCA using the proximal block coordinate descent algorithm, IEEE Transactions on Signal Processing 67 (20) (2019) 5402–5416.

[26] P. Jain, N. Rao, I. Dhillon, Structured sparse regression via greedy hard-thresholding, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16, Curran Associates Inc., Red Hook, NY, USA, 2016, p. 1524–1532.

[27] C. Zou, K. I. Kou, Y. Wang, Y. Y. Tang, Quaternion block sparse representation for signal recovery and classification, Signal Processing 179 (2021) 107849.

[28] L. Xiaohu, Y. Wanzhen, H. Jincai, A. Xing, T. Xishan, Non-convex block-sparse compressed sensing with coherent tight frames, EURASIP Journal on Advances in Signal Processing 2020 (2020) 1–9.

[29] T. Bluemensath, M. Davies, Iterative thresholding for sparse approximations, The Journal of Fourier Analysis and Applications 14 (2008) 629—-654.

[30] M. S. Bazaraa, Nonlinear Programming: Theory and Algorithms, 3rd Edition, Wiley Publishing, 2013.

[31] R. Glowinski, J.-L. Lions, R. Trémolières, Analyse Numérique des Inéquations Variationnelles - Tome 1 : Théorie Générale et Premières Applications - Tome 2 : Applications aux phénomènes stationnaires et d'évolution, Dunod, 1976.

[32] W. Wu, Convergence of the randomized block Gauss-Seidel method, SIAM Undergraduate Research Online 11 (2018) 369–382.

[33] Z. Lin, R. Liu, Z. Su, Linearized alternating direction method with adaptive penalty for low rank representation, NIPS (2011).

[34] L. Armijo, Minimization of functions having lipschitz continuous first partial derivatives., Pacific J. Math. 19 (3) (1966) 1–3.

[35] G. Guan, Q. Wan, F. Adachi, Direction of arrival estimation using modified orthogonal matching pursuit algorithm, Int. Journal of the Physical Sciences 6(22) (2011) 5230–5234.