Iteratively Reweighted two-Stage LASSO for Block-Sparse Signal Recovery Under Finite-Alphabet Constraints

Malek Messai, Abdeldjalil Aissa El Bey, Karine Amis Cavalec, Frédéric Guilloud

Iteratively Reweighted two-Stage LASSO for Block-Sparse Signal Recovery Under Finite-Alphabet Constraints

Malek Messai[a], Abdeldjalil Aïssa-El-Bey[a], Karine Amis[a,*], Frédéric Guilloud[a]

[a]*IMT Atlantique, UMR CNRS 6285 Lab-STICC, UBL, F-29238 Brest, France*

Abstract

In this paper, we derive an efficient iterative algorithm for the recovery of block-sparse signals given the finite data alphabet and the non-zero block probability. The non-zero block number is supposed to be far smaller than the total block number (block-sparse). The key principle is the separation of the unknown signal vector into an unknown support vector $\boldsymbol{s}$ and an unknown data symbol vector $\boldsymbol{a}$. Both number ($\|s\|_0$) and positions ($s_i \in \{0,1\}$) of non-zero blocks are unknown. The proposed algorithms use an iterative two-stage LASSO procedure consisting in optimizing the recovery problem alternatively with respect to $\boldsymbol{a}$ and with respect to $\boldsymbol{s}$. The first algorithm resorts on ℓ_1-norm of the support vector and the second one applies reweighted ℓ_1-norm, which further improves the recovery performance. Performance of proposed algorithms is illustrated in the context of sporadic multiuser communications. Simulations show that the reweighted-ℓ_1 algorithm performs close to its lower bound (perfect knowledge of the support vector).

Key words: Block-sparsity recovery, iterative reweighting, ℓ_1-minimization, iterative recovery algorithms, LASSO, finite-alphabet.

1. Introduction

Compressed Sensing (CS) is a signal processing approach that enables exact recovery of sparse signals from highly incomplete sets of linear measurements [1, 2, 3]. In last decade, CS techniques have spread in various disciplines such as medical imaging, machine learning, computer science and many others. In many applications of CS such as measurement of gene expression levels [4], face recognition [5] or data clustering [6], the signal is block-sparse, i.e., its non-zero elements are grouped into blocks [7]. Many algo-

[*] Corresponding author

rithms have been proposed to recover sparse signals having a block structure. They can be classified into two popular kinds of algorithms. The first ones, referred to as greedy CS algorithms, lead to the lowest-complexity detectors from CS theory. A popular example of greedy CS algorithm is Group-OMP [8, 9] where non-zero blocks of symbols are iteratively detected. The second kind, referred to as the ℓ_1-minimization family, comprises for example algorithms based on ℓ_2/ℓ_1-*norm minimization* [10]. This family provides a better sparse recovery with a lower number of measurements than the greedy one, and so lower error rates can be achieved at the expense of an increased complexity.

In many cases, signals belong to a finite alphabet and this property can be exploited to enhance the recovery efficiency. For instance, CS under finite-alphabet constraints was proposed in [11] in the case of a binary alphabet and in [12] for any finite alphabet. In this paper, we consider the reconstruction of arbitrary finite-alphabet signals having a block-sparse structure. As already mentioned, [7] and [10] deal with block-sparse signal recovery without taking into account the finite-alphabet constraint. The main drawback is the sensitivity of these algorithms to a ℓ_2-norm bounding constraint parameter that depends on the noise level, which may not be available. In this paper, we derive an iterative algorithm from ℓ_1-*minimization* family that takes into account the finite alphabet constraint and whose single parameter depends only on sparsity parameters. The key principle is the separation of the unknown signal vector into an unknown sparse support vector $\boldsymbol{s}$ and an unknown data symbol vector $\boldsymbol{a}$. Both number ($\|s\|_0$) and positions ($s_i \in \{0,1\}$) of non-zero blocks are unknown. Proposed algorithms use an iterative two-stage LASSO procedure consisting in optimizing the recovery problem alternatively with respect to $\boldsymbol{a}$ and with respect to $\boldsymbol{s}$. The first algorithm resorts on ℓ_1-norm of the support vector and the second one applies reweighted ℓ_1-norm [13], which further improves the recovery performance.

The paper is organized as follows: in Section 2, we first introduce the signal model before describing the proposed iterative reweighted algorithm in Section 3. In Section 4, proposed algorithms are applied in the context of sporadic communications. Finally, conclusions and perspectives are given in Section 5.

Throughout this paper, notations x, $\boldsymbol{x}$ and $\boldsymbol{X}$ denote respectively a scalar value, a column vector and a matrix. $\boldsymbol{X}^T$ is the transpose of $\boldsymbol{X}$ and $\boldsymbol{X}^H$ its Hermitian matrix. $\mathbf{0}_N$ (resp. $\mathbf{1}_N$) is the null (resp. all-one) vector of length N. $\hat{x}$ stands for the estimated value of x in a continuous domain included in $\mathbb{R}$ or $\mathbb{C}$. $Q_{\mathcal{A}}(x)$ stands for the quantizer operator which maps x to its closest value in a discrete alphabet denoted by $\mathcal{A}$. $\tilde{x}$ denotes an estimate of x in $\mathcal{A}$.

$\Re\{.\}$ and $\Im\{.\}$ denote, respectively, the real and the imaginary part of the object in brackets. $\otimes$ and $\odot$ denote the Kronecker and the Khatri-Rao products, respectively. Finally, $\alpha \leq \boldsymbol{x} \leq \beta$ means that each component x_i of $\boldsymbol{x}$ is bounded by α and β. When $\mathcal{A}$ is a set and n an integer, then $\mathcal{A}^n$ stands for the n-fold Cartesian product of $\mathcal{A}$. In an iterative algorithm seeking for the optimal value of $\boldsymbol{x}$, $\boldsymbol{x}^{(i)}$ denotes the value of $\boldsymbol{x}$ at the i-th iteration.

2. System Model

The basic CS model describes the linear connection between a sparse vector $\boldsymbol{x}$ and its measurement $\boldsymbol{y}$ by

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{\eta}, \tag{1}$$

where $\boldsymbol{\eta}$ denotes a zero mean complex Gaussian noise vector with covariance matrix $\sigma_\eta^2 \boldsymbol{I}$. $\boldsymbol{H} \in \mathbb{C}^{m \times n}$ is the underdetermined mixing matrix with fewer rows than columns ($m < n$). $\boldsymbol{x}$ is a block-sparse vector $\boldsymbol{x} = [\boldsymbol{x}_1^T, \boldsymbol{x}_2^T, \ldots, \boldsymbol{x}_K^T]^T$ composed of K length-N blocks $\boldsymbol{x}_i$ i.e., $n = KN$. The number of non-zero blocks in $\boldsymbol{x}$ is a random variable, denoted by K_a. When the k-th block is different from $\boldsymbol{0}_N$, it takes its value in finite alphabet $\mathcal{A}^N$. Thus, $\boldsymbol{x}_k$ belong to zero-augmented alphabet $\mathcal{A}^N \cup \{\boldsymbol{0}_N\}$. We denote by p_a the probability that a block is different from $\boldsymbol{0}_N$.

The original mathematical model which defines the sparsity condition in the noisy case is given by

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} ||\boldsymbol{x}||_0 \quad \text{s. t.} \quad ||\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}||_2 \leq \epsilon. \tag{2}$$

The optimization problem (2) is an NP-hard combinatorial problem [14]. A computationally tractable alternative is to relax the problem by performing ℓ_1-norm minimization in place of ℓ_0-norm minimization; the former can be solved in polynomial time using linear programming techniques [15, 16]. Then the problem is given by

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} ||\boldsymbol{x}||_1 \quad \text{s. t.} \quad ||\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}||_2 \leq \epsilon. \tag{3}$$

However solving (2) or (3) leads to a sparse vector that is not necessarily block-sparse, even when the true vector $\boldsymbol{x}$ is block-sparse [7]. In [7], the block-sparsity is taken into account by replacing the ℓ_0-norm used in (2): the non-zero blocks can be counted by combining an indicator function with ℓ_2-norm to detect energy which gives the following problem formulation:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \sum_{k=1}^{K} I_{(0,+\infty)}(||\boldsymbol{x}_k||_2) \quad \text{s.t.} \quad ||\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}||_2 \leq \epsilon, \tag{4}$$

where $I_A(.)$ is the indicator function of subset A. Greedy algorithms such as GOMP are used to solve (4). Another solution to reconstruct block-sparse signals is referred to as "ℓ_2/ℓ_1-norm minimization" [10]. Instead of using ℓ_1-norm as in (3), it combines ℓ_1-norm to measure the block sparsity and ℓ_2-norm to detect energy. By defining the energy vector $\boldsymbol{\xi}_{\boldsymbol{x}} = [\xi_{\boldsymbol{x}_1}, \ldots, \xi_{\boldsymbol{x}_K}]$ where $\xi_{\boldsymbol{x}_i} = ||\boldsymbol{x}_i||_2$ for $i = 1, \ldots, K$, the ℓ_2/ℓ_1-norm block-sparse signal recovery criterion is given by

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} ||\boldsymbol{\xi}_{\boldsymbol{x}}||_1 \quad \text{s. t.} \quad ||\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}||_2 \leq \epsilon,$$

which is equivalent to

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \sum_{k=1}^{K} ||\boldsymbol{x}_k||_2 \quad \text{s. t.} \quad ||\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}||_2 \leq \epsilon. \tag{5}$$

In all the approaches mentioned above, the upper bound ϵ depends on the noise deviation and on the number of observations. The optimization of ϵ is critical and requires a good knowledge of the noise standard deviation σ_η which may be unavailable. The resulting lack of accuracy is mitigated in Section 3 by reformulating the problem in such a way that its resolution doesn't need the noise deviation estimation. On the contrary, it exploits known information such as the source data alphabet and the probability that a block is null.

3. Proposed Algorithms

Greedy algorithms such as GOMP are the least complex detectors from CS theory, but they suffer from error propagation. A false non-zero detection at a given iteration can cause the data and the non-zero detection of all subsequently blocks to be incorrect. The error propagation increases with the number of non-zero blocks K_a and thus with the probability p_a.

In order to bypass the GOMP defaults, we aim to define a criterion and an algorithm which exploits the finite alphabet constraint and the block-sparsity structure of the signal. Therefore, the main objective is the definition of the constraints from $\boldsymbol{x}$ properties.

$\boldsymbol{x}$ is block-sparse and each block $\boldsymbol{x}_k$ equals $\mathbf{0}_N$ or belongs to $\mathcal{A}^N$. The probability that $\boldsymbol{x}_k$ differs from $\mathbf{0}_N$ is p_a.

By exploiting the independence between support (number and positions of non-zero blocks) and data values, we introduce the length-K binary support vector $\boldsymbol{s}$ and the block-data vector $\boldsymbol{a}$. $s_k = 1$ if the k-th block is non-zero and $s_k = 0$ otherwise. $\boldsymbol{a} = [\boldsymbol{a}_1^T, \boldsymbol{a}_2^T, \dots, \boldsymbol{a}_K^T]^T$ with $\boldsymbol{a}_k \in \mathcal{A}^N$ for $k = 1, \dots, K$.

We can then write $\boldsymbol{x}_k = s_k \boldsymbol{a}_k$ that is to say $\boldsymbol{x} = \boldsymbol{D_s a}$ where $\boldsymbol{D_s}$ is a $KN \times KN$ matrix given by $\boldsymbol{D_s} = \operatorname{diag}(\boldsymbol{s}) \otimes \boldsymbol{I}_N$.

Given the constraints, the detection problem can be written as

$$\begin{aligned} \underset{\boldsymbol{s},\boldsymbol{a}}{\arg\min} \quad & ||\boldsymbol{y} - \boldsymbol{H D_s a}||_2 \\ \text{s.t.} \quad & \boldsymbol{s} \in \{0,1\}^K \\ & \boldsymbol{a} \in \mathcal{A}^{KN} \end{aligned} \tag{6}$$

Problem (6) is NP-hard and requires an exhaustive search. In order to get a polynomial complexity, we relax the constraints and we use an iterative two-stage LASSO algorithm which alternates between support detection given $\boldsymbol{a}$ and signal estimation given $\boldsymbol{s}$. The constraint involving $\boldsymbol{s}$ is relaxed by using ℓ_1-norm as well as lower and upper bounds, whereas real and imaginary parts of $\boldsymbol{a}$ are bounded by extreme values of the finite-alphabet. Then problem (6) is relaxed by:

$$\begin{aligned} \underset{\boldsymbol{s},\boldsymbol{a}}{\arg\min} \quad & ||\boldsymbol{y} - \boldsymbol{H D_s a}||_2 \\ \text{s.t.} \quad & \|\boldsymbol{s}\|_1 \leq \delta \\ & 0 \leq \boldsymbol{s} \leq 1 \\ & \min \Re\{\mathcal{A}\} \leq \Re\left(\boldsymbol{a}\right) \leq \max \Re\{\mathcal{A}\} \\ & \min \Im\{\mathcal{A}\} \leq \Im\left(\boldsymbol{a}\right) \leq \max \Im\{\mathcal{A}\} \end{aligned} \tag{7}$$

δ is introduced so as to force the sparsity of $\boldsymbol{s}$. Its value can be fixed such that

$$\Pr(K_a \leq \delta) = \Pr\left(\sum_{k=1}^{K} s_k \leq \delta\right) = P_c, \tag{8}$$

where P_c is defined as the target confidence probability. As $K_a = \sum_{k=1}^{K} s_k$ follows the binomial distribution with parameters K and p_a, the probability of having a non-zero

Algorithm 1 Iterative two-stage LASSO

```
Initialize:
    $\hat{s}^{(0)} = \mathbf{1}_K$
for i =1 ... $N_{\text{iter}}$ do
    Stage (1) $\begin{cases} \hat{\boldsymbol{a}}^{(i)} = \arg\min_{\boldsymbol{a}} ||\boldsymbol{y} - \boldsymbol{H}\boldsymbol{D}_{\hat{\boldsymbol{s}}^{(i-1)}}\boldsymbol{a}||_2 \\ \text{s. t.} \begin{cases} \min \Re\{\mathcal{A}\} \leq \Re(\boldsymbol{a}) \leq \max \Re\{\mathcal{A}\} \\ \min \Im\{\mathcal{A}\} \leq \Im(\boldsymbol{a}) \leq \max \Im\{\mathcal{A}\} \end{cases} \end{cases}$
    Stage (2) $\begin{cases} \hat{\boldsymbol{s}}^{(i)} = \arg\min_{\boldsymbol{s}} ||\boldsymbol{y} - \boldsymbol{H}\boldsymbol{M}_{\hat{\boldsymbol{a}}^{(i)}}\boldsymbol{s}||_2 \\ \text{s. t.} \begin{cases} ||\boldsymbol{s}||_1 \leq \delta \\ 0 \leq \boldsymbol{s} \leq 1 \end{cases} \end{cases}$
end for
```

block number lower than δ in K trials is given by

$$\Pr(K_a \leq \delta) = \sum_{k=0}^{\lfloor \delta \rfloor} \binom{K}{k} p_a^k (1-p_a)^{K-k} = I_{1-p_a}(K - \lfloor \delta \rfloor, 1 + \lfloor \delta \rfloor) \tag{9}$$

with $\lfloor . \rfloor$ the greatest integer function and $I_x(\cdot, \cdot)$ the "regularized incomplete beta function". We obtain δ by solving (9) given a target confidence probability $P_c \triangleq \Pr(K_a \leq \delta)$. δ can also be interpreted as the maximum number of non-zero blocks that the algorithm will be able to detect. Its choice is as critical as the choice of ϵ in (5). But δ only depends on the probability p_a, while ϵ depends on the noise variance σ_η^2.

To solve (7), we assume as an initial condition that all blocks are non-zero: $\boldsymbol{s}^{(0)} = \mathbf{1}_K$. The proposed iterative process is referred to as "iterative two-stage LASSO" and is detailed in Algorithm 1 where $\boldsymbol{M}_{\boldsymbol{a}} = \boldsymbol{I}_K \odot \boldsymbol{a}$.

The algorithm output $\hat{\boldsymbol{s}}$ belongs to $[0, 1]$. To reconstruct $\boldsymbol{x}$, we need a binary support vector and we have two options to estimate it from $\hat{\boldsymbol{s}}$. The first one, without any prior, applies on $\hat{\boldsymbol{s}}$ a threshold fixed to 0.5. The second one exploits the knowledge of the number of non-zero blocks by selecting the K_a highest values of $\hat{\boldsymbol{s}}$.

Note that the alphabet constraint consists in a strong relaxation by the introduction of a box constraint on the alphabet. In practice, the recovery performance of (7) will improve as long as the cardinality of real and imaginary parts of $\mathcal{A}$ is small.

An interesting solution referred to as *reweighted ℓ_1-minimization* was proposed in [13] as an intermediate between ℓ_0 and ℓ_1 minimizations: it performs closer to ℓ_0 minimization while remaining convex as ℓ_1-norm based criteria. Therefore, based on [13], we propose in this paper the "Iterative Reweighted two-stage LASSO algorithm" where reweighted ℓ_1-norm minimization is applied in second stage for support detection. It is described in Algorithm 2 ($i-$th iteration). $\boldsymbol{W}$ is a diagonal matrix with diagonal entries $w_1, w_2, \ldots, w_K$. Initial weights are set to $w_k^{(0)} = 1,\ k = \{1, 2, \ldots, K\}$.

Algorithm 2 Reweighted Iterative two-stage LASSO

```
Initialize:
    $\hat{s}^{(0)} = \mathbf{1}_K$
for i =1 ... $N_{\text{iter}}$ do
    Stage (1) $\begin{cases} \hat{\boldsymbol{a}}^{(i)} = \arg\min_{\boldsymbol{a}} ||\boldsymbol{y} - \boldsymbol{H}\boldsymbol{D}_{\hat{\boldsymbol{s}}^{(i-1)}}\boldsymbol{a}||_2 \\ \text{s. t.} \begin{cases} \min \Re\{\mathcal{A}\} \leq \Re(\boldsymbol{a}) \leq \max \Re\{\mathcal{A}\} \\ \min \Im\{\mathcal{A}\} \leq \Im(\boldsymbol{a}) \leq \max \Im\{\mathcal{A}\} \end{cases} \end{cases}$
    Stage (2) $\begin{cases} \hat{\boldsymbol{s}}^{(i)} = \arg\min_{\boldsymbol{s}} ||\boldsymbol{y} - \boldsymbol{H}\boldsymbol{M}_{\hat{\boldsymbol{a}}^{(i)}}\boldsymbol{s}||_2 \\ \text{s. t} \begin{cases} ||\boldsymbol{W}^{(i)}\boldsymbol{s}||_1 = \sum_{k=1}^{K} w_k^{(i)}|s_k| \leq \delta \\ 0 \leq \boldsymbol{s} \leq 1 \end{cases} \end{cases}$
    Update $w_k^{(i+1)} = \frac{1}{|s_k^{(i)}|+\rho}$
end for
```

Non-zero weights $w_k^{(i+1)}$ are updated as in [13]. The parameter ρ ensures that a decision taken at one step can be questioned at next step i.e. a zero-valued component in $\boldsymbol{s}^{(i)}$ could turn into a non-zero estimate at next step. We consider the same optimization of ρ as in [13]:

$$\rho = \max\{|\boldsymbol{s}^{(i-1)}|_{(K_a+1)}, 10^{-3}\}, \tag{10}$$

where $|\boldsymbol{s}^{(i-1)}|_{(K_a+1)}$ represents the (K_a+1)-th element of the vector that orders $(|\boldsymbol{s}^{(i-1)}|)$ in decreasing order of magnitude. It requires the estimation of K_a, which can be obtained thanks to an energy detector. The purpose of ρ is to avoid the divergence of the norm due to high weight values caused by extremely low components of the support vector.

Let us note that Algorithm 2 only represents a variation of the one proposed in [13]. The cost function of (7) is continuously differentiable and constraint sets of Algorithms 1 and 2 are convex. The cost function remains convex with respect to one variable when the other is held constant. Being alternations of two convex optimizations with independent constraints, the convergence of Algorithms 1 and 2 is ensured by Propositions 2.5 and

3.9 in [17].

To give rough estimation of the complexity, we only kept the most complex computational operations and we neglected the others. For the GOMP algorithm, the overall complexity is dominated by the LS estimation step, maximum at last iteration and given by $\mathcal{O}((K_aN)^3)$. Hence, the complexity for the K_a iterations is upper-bounded by $K_a\,\mathcal{O}((K_aN)^3)$. For the ℓ_2/ℓ_1 algorithm, the overall complexity is given by $\mathcal{O}((KN)^3)$. Finally, for the proposed LASSO-based algorithm, the complexity of first stage is dominated by $\mathcal{O}((KN)^3)$, whereas for second stage, it is dominated by $\mathcal{O}(K^3)$. Therefore, its overall complexity is dominated by $N_{\text{iter}}\mathcal{O}((KN)^3)$. In next section, Algorithms 1 and 2 are applied in the context of sporadic communications.

4. Application to Sporadic Multiuser Communications

Multi-user detection (MUD) based on CS (CS-MUD) was recently proposed in [18, 19] in the context of sporadic machine type communications (MTC) to jointly detect user activity and user data. In sporadic MTC, K users can either send a small frame to an aggregation point (in that case, the user is said active) or remain idle. If more than one user is active, collisions occur at the aggregation point. We consider the linear model given in (1) to describe multi-user sporadic communications.

Recovery of sparse sequences requires a minimum number of observations [20] to estimate sparse vectors with overwhelming probability. A solution to increase the number of observations is to spread each symbol period using spreading sequences [21]. In that case, the spreading goal is obviously not orthogonal multiple access. In the following, we consider spreading sequences of length N_s smaller than the number of users which yields an underdetermined multiuser detection problem. Therefore, the system matrix $\boldsymbol{H}$ includes all the frequency-selective Rayleigh fading channels and the spreading sequences of each user. Hence, matrix $\boldsymbol{H} \in \mathbb{C}^{m\times n}$ with $n = KN$ and $m = N_sN$. The aggregation point has no knowledge whether the $k-$th user is active or not during a given time-slot. Therefore, user activity has to be jointly detected with transmitted data and algorithms proposed in Section 3 apply to this context. As in [18, 19], we consider that the number of active users K_a is perfectly estimated by an energy detector. Detection (either Algorithm 1 or Algorithm 2) is first carried out with δ fixed so that (8) is satisfied. The perfect knowledge of K_a is exploited first to fix ρ in (10) and secondly at the iterative process output by setting the K_a maximum values of $\hat{\boldsymbol{s}}$ to 1 and the remaining ones to 0. Then, we decide $\tilde{\boldsymbol{x}} = Q_{\mathcal{A}}\left(\boldsymbol{D}_{\hat{\boldsymbol{s}}}\hat{\boldsymbol{a}}\right)$.

In the simulations, we assume a low activation probability ($p_a < 0.5$). The activity of each user is randomly sorted according to p_a, such that $\mathbb{E}(K_a) = Kp_a$. The multipath fading channel is modeled by $L = 6$ independent Rayleigh-distributed taps. The total number of users is set to $K = 80$. Each pseudo-random sequence length is $N_s = 25$. An active user transmits $N = 30$ information symbols per frame which belong to alphabet $\mathcal{A}$ such that the variance of the symbols is equal to 1. The performance are given as a function of the signal-to-noise ratio (SNR) which equals

$$\text{SNR} = \frac{p_a K}{\sigma_\eta^2}.$$

Proposed algorithms are compared to the GOMP and to the ℓ_2/ℓ_1 *minimization* algorithm. Since no analytical solution is available [22] for problem (7), the convex optimization toolbox (CVX) [23] is used to execute the algorithms. "It. 2-stage LASSO" (resp. "Reweight. It. 2-stage LASSO") refers to Iterative two-stage LASSO algorithm (resp. Reweighted Iterative two-stage LASSO algorithm).

4.1. Symbol Error Rate

The symbol error rate (SER) is illustrated in Fig. 1 for BPSK modulation with $p_a = 15\%$ (Fig. 1a) and for 16-QAM modulation with $p_a = 10\%$ (Fig. 1b). Proposed algorithms are carried out with $P_c = 0.99$ and 40 iterations.

As benchmark, we propose two oracle algorithms which have perfect knowledge of the support vector $\boldsymbol{s}$.

For both of them, the observation vector equals $\boldsymbol{y} = \boldsymbol{H}_{\boldsymbol{s}}\boldsymbol{x}_{\boldsymbol{s}} + \boldsymbol{\eta}$, where $\boldsymbol{x}_{\boldsymbol{s}}$ combines all non-zero blocks $\boldsymbol{x}_k$ of $\boldsymbol{x}$ for which $s_k = 1$, and $\boldsymbol{H}_{\boldsymbol{s}}$ is the concatenation of the sub-matrices of $\boldsymbol{H}$ which have the same indices as the non-zero blocks. The first oracle referred to as oracle-LS is only based on least-squares estimation of data: $\hat{\boldsymbol{x}}_{\boldsymbol{s}} = \left(\boldsymbol{H}_{\boldsymbol{s}}^H \boldsymbol{H}_{\boldsymbol{s}}\right)^{-1} \boldsymbol{H}_{\boldsymbol{s}}^H \boldsymbol{y}$.

The second oracle referred to as oracle-BLS (bounded least squares) exploits the finite alphabet constraint thanks to bounded-variable convex constraints and estimates

$\boldsymbol{x}_s$ as follows:

$$\begin{cases} \hat{\boldsymbol{x}}_s = \arg\min\limits_{\boldsymbol{x_s}} ||\boldsymbol{y} - \boldsymbol{H_s x_s}||_2 \\ \text{s. t} \\ \min \Re\{\mathcal{A}\} \leq \Re\left(\boldsymbol{x_s}\right) \leq \max \Re\{\mathcal{A}\} \\ \min \Im\{\mathcal{A}\} \leq \Im\left(\boldsymbol{x_s}\right) \leq \max \Im\{\mathcal{A}\} \end{cases} \tag{11}$$

The oracle-LS bound is above the oracle-BLS bound since the latter considers the alphabet constraint. Improvement offered by proposed algorithms is significant. An important gain appears for SNR higher than 12 dB and 27 dB for BPSK and 16-QAM respectively. Iterative two-stage LASSO performs close to oracle-LS, while Reweighted Iterative two-stage LASSO outperforms oracle-LS and achieves a SER close to the oracle-BLS one.

4.2. Support Detection

The probability that an inactive user is detected as active and vice versa is expressed as $E[\|\boldsymbol{s} - \tilde{\boldsymbol{s}}\|_2^2] = \frac{1}{K}\sum_{i=1}^{K}|s_i - \tilde{s}_i|^2$ where $\tilde{\boldsymbol{s}}$ denotes the decided support given by $\tilde{\boldsymbol{s}} = Q_{\{0,1\}}(\hat{\boldsymbol{s}})$. This activity-detection error rate is illustrated in Fig. 2 for BPSK modulation with $p_a = 15\%$ (Fig. 2a) and for 16-QAM modulation with $p_a = 10\%$ (Fig. 2b).

We observe an error floor for the GOMP algorithm and also a lower one for the ℓ_2/ℓ_1 minimization algorithm. Proposed iterative two-stage LASSO algorithms outperform the other two with no error floor observed in the activity-detection error rate and with a better performance for the reweighted version.

5. Conclusion

In this paper, we proposed a new iterative ℓ_1-minimization algorithm that exploits the finite alphabet knowledge and iterates between data estimation and activity detection. Its parameter only depends on the activation probability. We introduced an improved version by reweighting ℓ_1-norm of the support vector. Both algorithms are evaluated in the context of sporadic multiuser communications. They perform better than GOMP and ℓ_2/ℓ_1-minimization algorithms without error floor observed in the activity-detection error-rate. In the future, we will investigate the robustness towards estimation of the active user number and to channel estimation (with the help of additional pilot symbols). Proposed algorithms would iterate between channel estimation and support detection.

References

[1] E. J. Candes, J. Romberg, T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, IEEE Transactions on Information Theory 52 (2) (2006) 489–509. doi:10.1109/TIT.2005.862083.

[2] J. A. Tropp, A. C. Gilbert, M. J. Strauss, Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit, Signal Processing 86 (3) (2006) 572–588. doi:10.1016/j.sigpro.2005.05.030.

[3] J. A. Tropp, Algorithms for simultaneous sparse approximation. Part II: Convex relaxation, Signal Processing 86 (3) (2006) 589 – 602. doi:10.1016/j.sigpro.2005.05.031.

[4] F. Parvaresh, H. Vikalo, S. Misra, B. Hassibi, Recovering sparse signals using sparse measurement matrices in compressed DNA microarrays, Journal of Selected Topics in Signal Processing 2 (3) (2008) 275–285. doi:10.1109/JSTSP.2008.924384.

[5] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 210–227. doi:10.1109/TPAMI.2008.79.

[6] E. Elhamifar, R. Vidal, Sparse subspace clustering: Algorithm, theory, and applications, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (11) (2013) 2765–2781. doi:10.1109/TPAMI.2013.57.

[7] Y. C. Eldar, P. Kuppinger, H. Bolcskei, Block-sparse signals: Uncertainty relations and efficient recovery, IEEE Transactions on Signal Processing 58 (6) (2010) 3042–3054. doi:10.1109/TSP.2010.2044837.

[8] G. Swirszcz, N. Abe, A. C. Lozano, Grouped orthogonal matching pursuit for variable selection and prediction, in: Advances in Neural Information Processing Systems 22, Curran Associates, Inc., 2009, pp. 1150–1158.

[9] Y. Fu, H. Li, Q. Zhang, J. Zou, Block-sparse recovery via redundant block OMP, Signal Processing 97 (2014) 162–171. doi:10.1016/j.sigpro.2013.10.030.

[10] M. Stojnic, F. Parvaresh, B. Hassibi, On the reconstruction of block-sparse signals with an optimal number of measurements, IEEE Transactions on Signal Processing 57 (8) (2009) 3075–3085. doi:10.1109/TSP.2009.2020754.

[11] Z. Tian, G. Leus, V. Lottici, Detection of sparse signals under finite-alphabet constraints, in: IEEE International Conference on Acoustics, Speech and Signal Processing, 2009, pp. 2349–2352. doi:10.1109/ICASSP.2009.4960092.

[12] A. Aïssa El Bey, D. Pastor, S. M. Aziz Sbai, Y. Fadlallah, Sparsity-based Recovery of Finite Alphabet Solutions to Underdetermined Linear Systems, IEEE Transactions on Information Theory 61 (4) (2015) 2008–2018. doi:10.1109/TIT.2015.2399914.

[13] 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. doi:10.1007/s00041-008-9045-x.

[14] B. K. Natarajan, Sparse approximate solutions to linear systems, SIAM Journal on Computing 24 (2) (1995) 227–234. doi:10.1137/S0097539792240406.

[15] D. L. Donoho, Compressed sensing, IEEE Transactions on Information Theory 52 (4) (2006) 1289–1306. doi:10.1109/TIT.2006.871582.

[16] E. J. Candès, J. K. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements, Communications on Pure and Applied Mathematics 59 (8) (2006) 1207–1223. doi:10.1002/cpa.20124.

[17] D. Bertsekas, J. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, Athena Scientific, 1997.

[18] C. Bockelmann, H. F. Schepker, A. Dekorsy, Compressive sensing based multiuser detection for machine-to-machine communication, Transactions on Emerging Telecommunications Technologies 24 (4) (2013) 389–400. doi:10.1002/ett.2633.

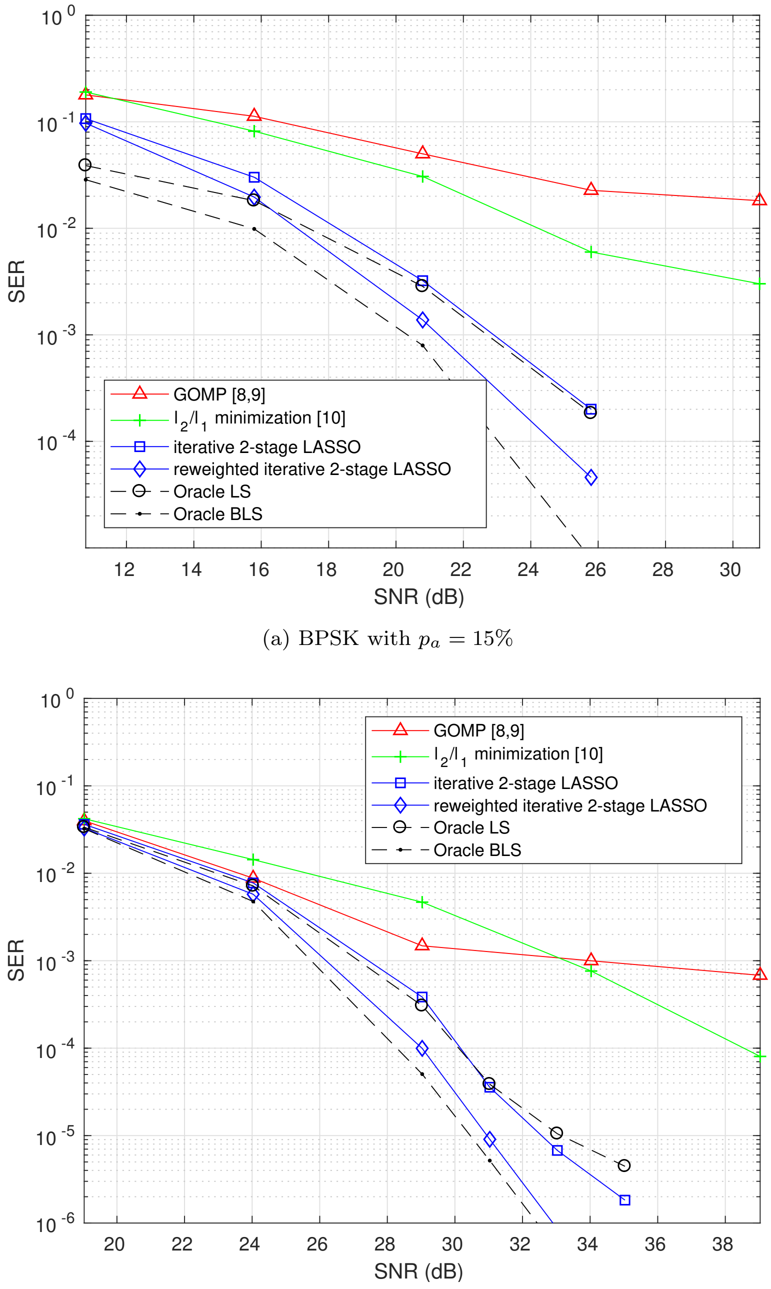
[19] H. F. Schepker, C. Bockelmann, A. Dekorsy, Efficient detectors for joint compressed sensing detection and channel decoding, IEEE Transactions on Communications 63 (6) (2015) 2249–2260. doi:10.1109/TCOMM.2015.2424414.

[20] E. J. Candès, T. Tao, Near-optimal signal recovery from random projections: Universal encoding strategies?, IEEE Transactions on Information Theory 52 (12) (2006) 5406–5425. doi:10.1109/TIT.2006.885507.

[21] H. Zhu, G. B. Giannakis, Exploiting sparse user activity in multiuser detection, IEEE Transactions on Communications 59 (2) (2011) 454–465. doi:10.1109/TCOMM.2011.121410.090570.

[22] P. B. Stark, R. Parker, Bounded-variable least-squares: an algorithm and applications, Computational Statistics 10 (2) (1995) 129–141.

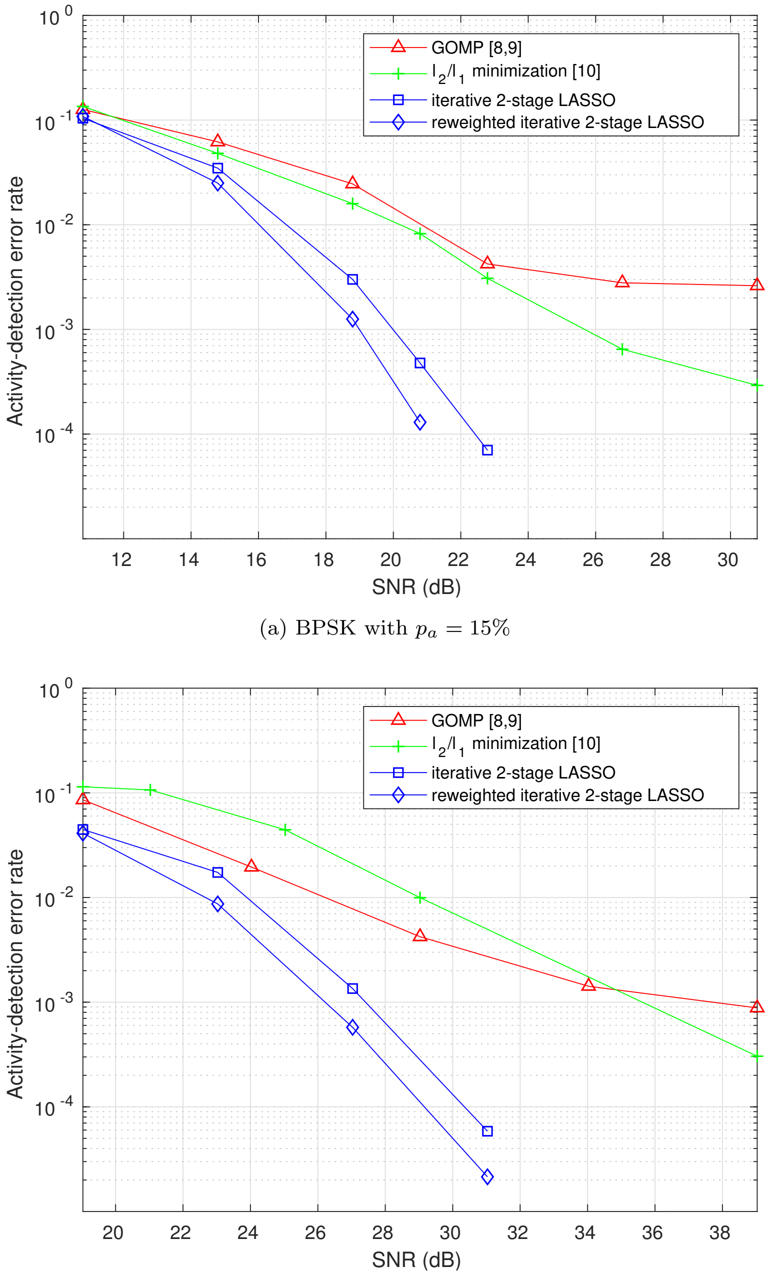
[23] M. Grant, S. Boyd, CVX: Matlab software for disciplined convex programming, version 2.1, `http://cvxr.com/cvx` (march 2014).



(a) BPSK with $p_a = 15\%$

(b) 16-QAM with $p_a = 10\%$

Figure 1: SER comparisons for BPSK and 16-QAM modulations with $K = 80$ users and $N_s = 25$.



(a) BPSK with $p_a = 15\%$

(b) 16-QAM with $p_a = 10\%$

Figure 2: Activity-detection error rate for BPSK and 16-QAM modulations with $K = 80$ users and $N_s = 25$.