Signal Processing 108 (2015) 451–458

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Signal recovery for jointly sparse vectors with different sensing matrices ☆

Li Li [a], Xiaolin Huang [b,*], Johan A.K. Suykens [b]

[a] Department of Automation, Tsinghua University, Beijing 100084, PR China
[b] KU Leuven, Department of Electrical Engineering (ESAT-STADIUS), B-3001 Leuven, Belgium

ARTICLE INFO

Article history:
Received 28 May 2014
Received in revised form
6 October 2014
Accepted 8 October 2014
Available online 22 October 2014

Keywords:
Sparse representation
Compressed sensing
Re-weighted algorithm
Multiple measurement vector

ABSTRACT

In this paper, we study a sparse multiple measurement vector problem in which we need to recover a set of jointly sparse vectors from incomplete measurements. Most related studies assumed that all these measurements correspond to the same compressed sensing matrix. Differently, we allow that the measurements come from different sensing matrices. To deal with different matrices, we establish an algorithm via applying block coordinate descent and Majorization–Minimization techniques. The numerical examples demonstrate the effectiveness of this new algorithm, which allows us to design different matrices for better recovery performance.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

With the advent of compressive sensing [1,2] and sparse regression (e.g., *lasso* [3]), finding sparse solutions becomes a vital task in many areas of signal processing and statistics. The key problem in this field is about solving the under-determined linear system of equations [4]:

$$A\boldsymbol{x} = \boldsymbol{b},$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^n$ are the variables. The compressive sensing matrix $A \in \mathbb{R}^{m\times n}$ and the measurement $\boldsymbol{b} \in \mathbb{R}^m$ are known. Particularly, we have $m < n$. Through this paper, we denote scalars with lower-case letters (e.g., x), vectors with lower-case bold-faced letters (e.g., $\boldsymbol{x}$), and matrices with uppercase letters (e.g., A).

This system contains fewer equations than unknown variables and usually has infinitely many solutions. It is obviously impossible to identify the desired solution without additional information. A fruitful regularization constraint for this problem is to recover a sparse or compressible solution. A plain explanation for this requirement is that we want to seek the simplest model fitting the data. In many instances, this requirement gives the most concise yet explainable model.

Mathematically speaking, such a requirement leads to the following constrained ℓ_0 minimization problem:

$$(P_0) \quad \min_{\boldsymbol{x}} \quad \|\boldsymbol{x}\|_0 \quad \text{s.t. } A\boldsymbol{x} = \boldsymbol{b},$$

where $\|\boldsymbol{x}\|_0$ denotes the number of non-zero entries of vector $\boldsymbol{x}$. Some greedy algorithms, e.g., *matching pursuit* algorithm [5,6], were designed to directly attack Problem (P_0). However, Problem (P_0) is generally difficult to solve, since a thorough combinatorial search often consumes intractable time. A widely used alternative is the following constrained ℓ_1 minimization problem [7–9]:

$$(P_1) \quad \min_{\boldsymbol{x}} \quad \|\boldsymbol{x}\|_1 \quad \text{s.t. } A\boldsymbol{x} = \boldsymbol{b},$$

☆ This work is supported in part by National Natural Science Foundation of China – 51278280 and 61473165 and Project Supported by Tsinghua University (20131089307). Xiaolin Huang and Johan Suykens acknowledge support from KU Leuven, The Flemish government, FWO, The Belgian federal science policy office and The European Research Council (CoE EF/05/006, GOA MANET, IUAP DYSCO, FWO G.0377.12, BIL with Tsinghua University, ERC AdG A-DATADRIVE-B).

* Corresponding author.
E-mail addresses: li-li@mail.tsinghua.edu.cn (L. Li), huangxl06@mails.tsinghua.edu.cn (X. Huang), johan.suykens@esat.kuleuven.be (J.A.K. Suykens).

http://dx.doi.org/10.1016/j.sigpro.2014.10.010
0165-1684/© 2014 Elsevier B.V. All rights reserved.

where $\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i|$. Problem (P_1) is convex and easy to solve. As one milestone in this field, it has been pointed out in [9] that the solutions to both Problems (P_0) and (P_1) are unique and with very high probability the same, if the vector is sparse enough. Because of its calculation convenience and feasibility to recover sparsity solution, ℓ_1 norm is widely used as a sparsity-promoting in many applications.

An important extension of the sparse recovery problem is the sparse multiple measurement vector (MMV) problems [19–24], in which we may have some incomplete linear measurements $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_K$ that correspond to different sparse solutions $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K \in \mathbb{R}^n$ as

$$A_k \boldsymbol{x}_k = \boldsymbol{b}_k,$$

where $\boldsymbol{b}_k \in \mathbb{R}^{m_k}$, $A_k \in \mathbb{R}^{m_k \times n}$, $k = 1, \ldots, K$.

The additional sparsity constraint for MMV problems is *jointly sparse*. That is, all $\boldsymbol{x}_k$ have the same sparsity structure. The sparsity structure of $\boldsymbol{x}$ is described by its support set as

$$\mathrm{supp}(\boldsymbol{x}) = \{i : x_i \neq 0\}.$$

Then the jointly sparse assumption means that

$$\mathrm{supp}(\boldsymbol{x}_1) = \mathrm{supp}(\boldsymbol{x}_2) = \ldots = \mathrm{supp}(\boldsymbol{x}_K). \tag{1}$$

In other words, the indices of the non-zero entries of all $\boldsymbol{x}_k$ are exactly the same.

In MMV problems, a signal corresponds to several $\boldsymbol{x}_k$ that have different values but the same sparsity structure. Recovering this signal appears in distributed or time-varying systems and has been recently discussed by [25–27]. These methods further assume that all the measurements correspond to the same compressive matrix, that is, $A_1 = A_2 = \cdots = A_K$. In this case, the rank aware pursuit algorithms have been established. The basic idea is to combine the measurements as $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_K]$, $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_K]$ and then to agglomerate the compressive sensing relations into one linear equation set $A\boldsymbol{X} = \boldsymbol{B}$. Then finding the jointly sparse solutions equals to finding a $\boldsymbol{X}$ with a small rank.

The above rank aware pursuit algorithm requires that all the A_k are the same. However, in real applications, one may meet different compressive sensing matrices. For example, in a distributed system, there are different sensors for one object [31]. Similar phenomena appear in time-varying system. Besides, different A_k provide more information and hence its sparse recovery performance could be better than one single sensing matrix. Under the Bayesian linear regression analysis framework, different compressive sensing matrices have been studied [28–30]. But in that field, joint sparsity has not been considered, i.e., $\boldsymbol{x}_k$ are not required to be jointly sparse.

In this paper, we will discuss the problem of signal recovery by appropriately combining jointly sparse measurements with different compressive sensing matrices. To fully utilize the joint sparsity, one new re-weighted ℓ_1 minimization algorithm will be designed to recover the vectors when they are not sparse enough to be recovered by conventional ℓ_1 minimization. The new algorithm consists of a sequence of weighted ℓ_1 minimization problems for each vector, where the weights of every vector used for the next iteration are computed based upon all the obtained solutions. This algorithm can be viewed as a mixture of a block coordinate descent algorithm and a Majorization–Minimization algorithm.

To address the benefits of solving such interlaced minimization problems, the rest of this paper is arranged as follows. Section 2 constructs a new ℓ_0 minimization problem for simultaneously recovering multiple vectors with the same sparsity structure. To solve the new ℓ_0 minimization problem, Section 3 briefly introduces the block coordinate descent algorithm and the Majorization–Minimization algorithm, which are the foundation techniques of the established method. Section 4 first reviews the classical re-weighted ℓ_1 minimization algorithm for a single set of measurements and then proposes a new re-weighted ℓ_1 minimization algorithm for multiple sets of measurements. Section 5 provides numerical tests to evaluate the effectiveness of the proposed algorithm. Finally, Section 6 concludes the paper.

2. New ℓ_0 minimization problem for multiple measurements

We first consider the case involving two sets of measurements. The general cases with more sets of measurements will be discussed in the end of this section. Suppose that there are two sets of measurements $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ corresponding to two vectors $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^n$ via compressive sensing matrix A_1 and A_2 respectively as

$$A_1 \boldsymbol{x}_1 = \boldsymbol{b}_1, \quad A_2 \boldsymbol{x}_2 = \boldsymbol{b}_2, \tag{2}$$

where we assume that $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ have the same sparsity structure. Any two solutions satisfying constraints (2) are called feasible solutions in the follows. Similar to [30], we do not require the compressive sensing matrices are the same, i.e., we allow $A_1 \neq A_2$ and $\boldsymbol{b}_1 \neq \boldsymbol{b}_2$. They may even have different dimensions, i.e., $m_1 \neq m_2$.

Apparently, the sparsest solutions can be obtained via the following constrained ℓ_0 minimization problem:

$$(P_0') \quad \min_{\boldsymbol{x}_1, \boldsymbol{x}_2} \quad \|\boldsymbol{x}_1\|_0 + \|\boldsymbol{x}_2\|_0 \tag{3}$$

$$\text{s.t.} \quad A_1 \boldsymbol{x}_1 = \boldsymbol{b}_1, \quad A_2 \boldsymbol{x}_2 = \boldsymbol{b}_2. \tag{4}$$

Since $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are separable in both the objective function (3) and constraints (4), Problem (P_0') is indeed equivalent to two separated Problem (P_0) for $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, respectively.

For any vector $\boldsymbol{z} \in \mathbb{R}^n$, we construct a diagonal matrix $D(\boldsymbol{z}, \tau)$, where the ith diagonal element d_{ii} of matrix $D(\boldsymbol{z}, \tau)$ is defined as

$$d_{ii}(\boldsymbol{z}, \tau) = \begin{cases} 1 & \text{for } |z_i| \leq \tau, \\ 0 & \text{otherwise}, \end{cases} \quad i = 1, 2, \ldots, n, \tag{5}$$

where $\tau \geq 0$ is a thresholding scalar. Clearly, if $\tau \to 0$, this function labels the entries of $\boldsymbol{z}$ that are near 0. When $\tau = 0$, it labels the entries of $\boldsymbol{z}$ that equal to 0. Thus, $D(\boldsymbol{x}_2, 0)\boldsymbol{x}_1$ only keeps the entries of $\boldsymbol{x}_1$ where the corresponding entries of $\boldsymbol{x}_2$ are 0. Similarly, $D(\boldsymbol{x}_1, 0)\boldsymbol{x}_2$ only keeps the entries of $\boldsymbol{x}_2$ where the corresponding entries of $\boldsymbol{x}_1$ are 0.

As a result, only two feasible solutions $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ with the same sparsity structure can be the optimal solutions of

the following constrained ℓ_0 minimization problem:

$$(P_0^c) \quad \min_{\boldsymbol{x}_1,\boldsymbol{x}_2} \quad \|D(\boldsymbol{x}_2,0)\boldsymbol{x}_1\|_0+\|D(\boldsymbol{x}_1,0)\boldsymbol{x}_2\|_0$$
$$\text{s.t.} \quad A_1\boldsymbol{x}_1=\boldsymbol{b}_1, \quad A_2\boldsymbol{x}_2=\boldsymbol{b}_2,$$

since the objective function (P_0^c) reaches the lowest possible value 0 for feasible solutions $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ satisfying (1).

The objective function (P_0^c) only forces on finding two feasible solutions with the same sparsity structure but does not urge to find the sparsest solutions. To recover two sparse vectors with the same sparsity structure, we further pursue the sparsity of $\boldsymbol{x}_1,\boldsymbol{x}_2$ and establish the constrained ℓ_0 minimization problem below:

$$(P''_0) \quad \min_{\boldsymbol{x}_1,\boldsymbol{x}_2} \quad \|\boldsymbol{x}_1\|_0+\|\boldsymbol{x}_2\|_0 + \|D(\boldsymbol{x}_2,0)\boldsymbol{x}_1\|_0+\|D(\boldsymbol{x}_1,0)\boldsymbol{x}_2\|_0$$
$$\text{s.t.} \quad A_1\boldsymbol{x}_1=\boldsymbol{b}_1, \quad A_2\boldsymbol{x}_2=\boldsymbol{b}_2,$$

where the first two items are for sparse solutions and the last two items make the solutions have the same sparsity structure.

Moreover, we have the following theorem stating the link between Problem (P'_0) and (P''_0).

Theorem 1. *If the sparsest feasible solutions of Problem (P''_0) have the same sparse structure, the globally optimal solutions to Problem (P''_0) are the globally optimal solutions to Problem (P'_0); and vice versa.*

Proof. Suppose two vectors $\overline{\boldsymbol{x}}_1$ and $\overline{\boldsymbol{x}}_2$ are the sparsest feasible solutions. Thus, $\overline{\boldsymbol{x}}_1$ and $\overline{\boldsymbol{x}}_2$ are optimal to Problem (P'_0). Meanwhile, these vectors will have the same sparse structure by assumption. So, they satisfy (1),

$$\|D(\overline{\boldsymbol{x}}_2,0)\overline{\boldsymbol{x}}_1\|_0=\|D(\overline{\boldsymbol{x}}_1,0)\overline{\boldsymbol{x}}_2\|_0=0,$$

which follows that $\overline{\boldsymbol{x}}_1$ and $\overline{\boldsymbol{x}}_2$ are globally optimal to Problem (P''_0).

Similarly, we can prove that the globally optimal solutions to Problem (P''_0) are meanwhile globally optimal to Problem (P'_0). □

Since Problem (P''_0) provides additional constraints on the two vectors, solving this new ℓ_0 minimization problem may increase the probability of correct recovery.

For multiple sets case, (P''_0) is extended to the constrained ℓ_0 minimization problem below:

$$(P'''_0) \quad \min_{\boldsymbol{x}_k} \quad \sum_{k=1}^{K}\|\boldsymbol{x}_k\|_0+\sum_{k=1}^{K}\sum_{l=1,l\neq k}^{K}\|D(\boldsymbol{x}_l,0)\boldsymbol{x}_k\|_0$$
$$\text{s.t.} \quad A_k\boldsymbol{x}_k=\boldsymbol{b}_k, \quad k=1,\ldots,K.$$

Since Problems (P''_0) and (P'''_0) are hard to solve, we will formulate the corresponding constrained ℓ_1 problems and solve them instead in Section 4.

3. Two optimization methods

To solve the proposed ℓ_0 minimization problems, we will apply two frequently-used optimization methods. Before presenting the main algorithm of this paper, we will first introduce these methods briefly.

3.1. Block coordinate descent algorithm

Coordinate descent algorithm has a long history in optimization studies. Usually, this algorithm works through a sequence of one-dimensional optimizations, whereby at each iteration one coordinate of the variable is adjusted to optimize the objective function, while the other coordinates are held fixed. It requires few calculation memory and is highly parallelizable [32,33]. Because of its conceptual and algorithmic simplicity, coordinate descent method has also been applied to sparsity constrained optimization problem and compressive sensing; see, e.g., [34–36].

The block coordinate descent algorithm can be viewed as a natural extension to the coordinate descent algorithm. This algorithm solves the optimization problems with multiple variables that can be partitioned into several blocks. In each iteration, we focus on updating a single block of variables only and keep the remaining blocks fixed. A survey of block coordinate descent algorithms in semidefinite programming can be found in [37].

3.2. Majorization–minimization algorithm

Majorization–Minimization algorithm works by first finding a surrogate function that majorizes the objective function and then optimizing this surrogate function until a local optimum is reached; see, e.g., [38,39]. It can be used in a wide range of problems, including statistics and machine learning.

Consider a minimization problem, say

$$\min_{\boldsymbol{v}} \quad f(\boldsymbol{v}) \quad \text{s.t. } \boldsymbol{v}\in C,$$

where C is a convex set. Suppose $\boldsymbol{v}^{(j)}$ is a fixed value of $\boldsymbol{v}$ and $g(\boldsymbol{v}|\boldsymbol{v}^{(j)})$ denotes a real-valued function that depends on $\boldsymbol{v}^{(j)}$. The surrogate function $g(\boldsymbol{v}|\boldsymbol{v}^{(j)})$ is said to be the majorize function of $f(\boldsymbol{v})$ at the point $\boldsymbol{v}^{(j)}$ if

$$g(\boldsymbol{v}|\boldsymbol{v}^{(j)})\geq f(\boldsymbol{v}) \quad \text{for all } \boldsymbol{v} \quad \text{and} \quad g(\boldsymbol{v}^{(j)}|\boldsymbol{v}^{(j)})=f(\boldsymbol{v}^{(j)}).$$

In other words, $g(\boldsymbol{v}|\boldsymbol{v}^{(j)})$ lies above $f(\boldsymbol{v})$ and is tangent to it at the point $\boldsymbol{v}^{(j)}$. We can minimize the surrogate function $g(\boldsymbol{v}|\boldsymbol{v}^{(j)})$ rather than the actual function $f(\boldsymbol{v})$. For a concave function $f(\boldsymbol{v})$, one possible choice of the surrogate function is

$$g(\boldsymbol{v}|\boldsymbol{v}^{(j)})=f(\boldsymbol{v}^{(j)})+\nabla f(\boldsymbol{v}^{(j)})^T(\boldsymbol{v}-\boldsymbol{v}^{(j)}).$$

Starting with a feasible solution $\boldsymbol{v}^{(0)}\in C$, we can iteratively solve the following simple function to force the given objective function $f(\boldsymbol{v})$ downhill

$$\boldsymbol{v}^{(j+1)}=\arg\min_{\boldsymbol{v}\in C} f(\boldsymbol{v}^{(j)})+\nabla f(\boldsymbol{v}^{(j)})^T(\boldsymbol{v}-\boldsymbol{v}^{(j)}) \tag{6}$$

until convergence. Each iteration involves a convex optimization problem that can be easily calculated.

Further discussions on surrogate optimization methods can be found in [40,41].

4. New re-weighted algorithm for multiple measurements

4.1. Classical re-weighted algorithm

Let us recall the classical re-weighted algorithm for the following comparison. The starting point behind any re-weighted algorithm [15–18] is the one-to-one correspondence between the optimal solution of Problem (P_0) and the optimal solution of the following optimization problem:

$$(P_{\log}) \quad \min_{\boldsymbol{x}} \ \sum_i \log|x_i| \quad \text{s.t. } A\boldsymbol{x}=\boldsymbol{b}.$$

This is because $\|\boldsymbol{x}\|_0=\lim_{p\to 0}\sum_i|x_i|^p$ and $\lim_{p\to 0}1/p\sum_i(|x_i|^p-1)=\sum_i\log|x_i|$ [42,17]. To avoid calculating $\log|x_i|$ when $x_i=0$, one often considers the following problem instead

$$(P'_{\log}) \quad \min_{\boldsymbol{x}} \ \sum_i \log(|x_i|+\varepsilon) \quad \text{s.t. } A\boldsymbol{x}=\boldsymbol{b},$$

where ε is a small positive constant.

As demonstrated in [16], Problem $(P'_{\log})$ is equivalent to the optimization problem below:

$$(P''_{\log}) \quad \min_{\boldsymbol{x},\boldsymbol{u}} \ \sum_i \log(u_i+\varepsilon) \quad \text{s.t. } A\boldsymbol{x}=\boldsymbol{b}, |x_i|\le u_i.$$

The key idea of the classical re-weighted algorithm is to iteratively solve Problem $(P''_{\log})$ via a special Majorization–Minimization algorithm like (6).

Taking the partial derivative of this surrogate function, we get the optimization problem for the jth iteration as

$$\left(\boldsymbol{x}^{(j+1)},\boldsymbol{u}^{(j+1)}\right)=\arg\min_{\boldsymbol{x},\boldsymbol{u}}\sum_i\frac{u_i}{u_i^{(j)}+\varepsilon}$$
$$\text{s.t.} \quad A\boldsymbol{x}=\boldsymbol{b}, \quad |x_i|\le u_i.$$

Via eliminating $\boldsymbol{u}$, the above problem is equivalent to

$$\boldsymbol{x}^{(j+1)}=\arg\min_{\boldsymbol{x}}\sum_i\frac{|x_i|}{|x_i^{(j)}|+\varepsilon} \quad \text{s.t. } A\boldsymbol{x}=\boldsymbol{b}. \tag{7}$$

This is a constrained ℓ_1 minimization problem that can be quickly solved by many techniques [15–18].

The re-weighted algorithms were named after the inversely proportional weighting coefficient $(|x_i^{(j)}|+\varepsilon)^{-1}$ shown in the objective function of Problem (7). The intuitive explanation for this re-weighted trick is to force a potential solution $\boldsymbol{x}$ to shrink on the indices where $|x_i|$ is small. The constant ε is introduced to guarantee iteration stability and to ensure that a zero-valued component x_i in $\boldsymbol{x}$ does not strictly prohibit a nonzero x_i in the next iteration. However, ε is usually set as a very small value at the beginning of iterations. If the ith entry $x_i^{(j)}$ becomes about zero after the jth iteration, the corresponding weighting coefficient will then become a very large number satisfying $1/\varepsilon\gg 0$. This generally prevents us to reconsider the possibility to include ith entry $x_i^{(j)}$ as a part of the non-zero solution after the jth iteration.

To solve this problem, researchers often choose a gradually decreasing thresholding sequence $\varepsilon^{(j)}>0$ in practice [43,17]. Using this trick, we can write the classical iterative re-weighted algorithm as the following.

Algorithm 1. Classical iterative re-weighted algorithm.

Step 1. Set $W^{(0)}\in\mathbb{R}^{n\times n}$ as an identity matrix (that is, $w_{ii}=1$, $i=1,\ldots,n$). Set the iteration count j to zero.

Step 2. Solve the following ℓ_1 minimization problem with respect to the diagonal weighting matrix $W^{(j)}\in\mathbb{R}^{n\times n}$

$$\boldsymbol{x}^{(j+1)}=\arg\min_{\boldsymbol{x}}\|W^{(j)}\boldsymbol{x}\|_1 \quad \text{s.t. } A\boldsymbol{x}=\boldsymbol{b}.$$

Step 3. Update the diagonal weighting matrix as

$$w_{ii}^{(j+1)}=\frac{1}{|x_i^{(j+1)}|+\varepsilon^{(j+1)}},$$

where $\varepsilon^{(j+1)}>0$ is a gradually decreasing thresholding sequence.

Step 4. Terminate if $\boldsymbol{x}^{(j)}$ converges or we reach the maximum iterations $j_{\max}$. Otherwise, increase j and go back to Step 2.

4.2. New re-weighted algorithm for two measurements

Similar to the derivations for the classical re-weighted algorithm, we will establish a re-weighted algorithm for Problem (P''_0) proposed in Section 2. The general cases with more sets of measurements have the similar properties and will be discussed in the end of this subsection.

Problem (P''_0) can be turned to an equivalent form as

$$(P^{+}_{\log}) \quad \min_{\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{u}_1,\boldsymbol{u}_2} \ h(\boldsymbol{u}_1,\boldsymbol{u}_2)$$
$$\text{s.t.} \quad A_1\boldsymbol{x}_1=\boldsymbol{b}_1, \quad A_2\boldsymbol{x}_2=\boldsymbol{b}_2$$
$$|x_{1,i}|\le u_{1,i}, \quad |x_{2,i}|\le u_{2,i}, \tag{8}$$

where the objective function is defined as

$$h(\boldsymbol{u}_1,\boldsymbol{u}_2)=\sum_i\big(\log(u_{1,i}+\varepsilon)+\log(u_{2,i}+\varepsilon)$$
$$+\log\left[\theta(u_{2,i},0)\cdot u_{1,i}+\varepsilon\right]+\log\left[\theta(u_{1,i},0)\cdot u_{2,i}+\varepsilon\right]\big),$$

where $\theta(u_i,\tau)$ is defined according to (5) as

$$\theta(u_i,\tau)=\begin{cases}0 & \text{for } u_i\ge\tau,\\ 1 & \text{for } 0\le u_i<\tau.\end{cases} \tag{9}$$

Since $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are interlaced in the objective function of Problem $(P^{+}_{\log})$, we use the block coordinate descent algorithm to solve this problem in this paper. More precisely, in each iteration, we will first take $\boldsymbol{x}_2$ and $\boldsymbol{u}_2$ as constants and optimize the objective function $h(\boldsymbol{u}_1,\boldsymbol{u}_2)$ only with respect to $\boldsymbol{x}_1$ and $\boldsymbol{u}_1$. Under this assumption, the objective function $h(\boldsymbol{u}_1,\boldsymbol{u}_2)$ can be simplified into

$$h(\boldsymbol{u}_1)=\sum_i\big(\log(u_{1,i}+\varepsilon)+\log\left[\theta(u_{2,i},0)\cdot u_{1,i}+\varepsilon\right]$$
$$+\log\left[\theta(u_{1,i},0)\cdot u_{2,i}+\varepsilon\right]\big). \tag{10}$$

However, this objective function is neither convex nor concave. The first two items of (10), i.e., $\log(u_{1,i}+\varepsilon)$ and $\log\left[\theta(u_{2,i},0)\cdot u_{1,i}+\varepsilon\right]$, mainly contribute to the value of objective function. Based on this observation, in this paper, we choose the following inaccurate surrogate function:

$$h'(\boldsymbol{u}_1)=\sum_i\big(\log(u_{1,i}+\varepsilon)+\log\left[\theta(u_{2,i},0)\cdot u_{1,i}+\varepsilon\right]\big).$$

Numerical examples given in Section 4 will illustrate that the inaccurate Majorization–Minimization algorithm works well in our problem.

Taking the partial derivative of $h'(\boldsymbol{u}_1)$ in (9), we have the optimization problem of $\boldsymbol{x}_1$ in the jth iteration as

$$\min_{\boldsymbol{x}_1,\boldsymbol{u}_1} \sum_i \left(\frac{u_{1,i}}{u_{1,i}^{(j)}+\varepsilon}+\frac{\theta(u_{2,i}^{(j)},0)\cdot u_{1,i}}{\theta(u_{2,i}^{(j)},0)\cdot u_{1,i}^{(j)}+\varepsilon}\right)$$
$$\text{s.t.}\quad A_1\boldsymbol{x}_1=\boldsymbol{b}_1,\quad |x_{1,i}|\le u_{1,i}.$$

Via eliminating $\boldsymbol{u}_1$, the sub-problem of $\boldsymbol{x}_1$ in the jth iteration can be rewritten as

$$\min_{\boldsymbol{x}_1} \sum_i \left(\frac{1}{|x_{1,i}^{(j)}|+\varepsilon}+\frac{\theta(|x_{2,i}^{(j)}|,0)}{\theta(|x_{2,i}^{(j)}|,0)\cdot|x_{1,i}^{(j)}|+\varepsilon}\right)|x_{1,i}|$$
$$\text{s.t.}\quad A_1\boldsymbol{x}_1=\boldsymbol{b}_1.$$

This is a constrained ℓ_1 minimization problem and can be solved easily with many existing methods [4,10–18].

After obtaining a minimum of this sub-problem, we then take $\boldsymbol{x}_1$ and $\boldsymbol{u}_1$ as constants and optimize the objective function $h(\boldsymbol{u}_1,\boldsymbol{u}_2)$ only with respect to $\boldsymbol{x}_2$ and $\boldsymbol{u}_2$. Formulating this subproblem into a constrained ℓ_1 minimization problem is similar to the above procedure and is thus omitted.

We can see that the core of the new algorithm is using the solution $\boldsymbol{x}_2^{(j)}$ to update the weighting matrix for the solution $\boldsymbol{x}_1^{(j)}$ to enhance sparsity and vice versa for the solution $\boldsymbol{x}_2^{(j)}$. Via this weighting technique, we can force two solutions to have the same or at least very similar sparsity structure at the end of iterations.

Further numerical tests showed that the above block coordinate descent+Majorization–Minimization algorithm can converge to a local minima even when the last item of (10), i.e., $\log[\theta(u_{1,i},0)\cdot u_{2,i}+\varepsilon]$, is neglected. This helps to boil down the minimization problem (8) to simplify re-weighted ℓ_1 minimization problems. All the calculating methods for the classical ℓ_1 minimization problem can then be adopted here.

For empirical calculating, we choose a small positive scalar $\tau>0$ to determine whether an entry of $\boldsymbol{x}_1$ or $\boldsymbol{x}_2$ reaches zero. In summary, we write this new iterative re-weighted algorithm for two sets of measurements below.

Algorithm 2. New re-weighted algorithm for two vectors:

Step 1. Set $W_1^{(0)}$ and $W_2^{(0)}$ as two identity matrices. Set the iteration count j to zero.

Step 2. Solve the following ℓ_1 minimization problem for $\boldsymbol{x}_1$

$$\boldsymbol{x}_1^{(j+1)}=\arg\min_{\boldsymbol{x}_1}\|W_1^{(j)}\boldsymbol{x}_1\|_1\quad \text{s.t. } A_1\boldsymbol{x}_1=\boldsymbol{b}_1,$$

and update the diagonal weighting matrix W_2 as

$$w_{2,ii}^{|j|}=\frac{1}{|x_{2,i}^{|j|}|+\varepsilon^{|j|}}+\frac{\theta(|x_{1,i}^{|j+1|}|,\tau)}{\theta(|x_{1,i}^{|j+1|}|,\tau)\cdot|x_{2,i}^{|j|}|+\varepsilon^{|j|}},$$

where $\varepsilon^{(j+1)}>0$ is a gradually decreasing thresholding sequence. $\tau>0$ is a pre-selected thresholding scalar.

Step 3. Solve the following ℓ_1 minimization problem for $\boldsymbol{x}_2$:

$$\boldsymbol{x}_2^{(j+1)}=\arg\min_{\boldsymbol{x}_2}\|W_2^{(j)}\boldsymbol{x}_2\|_1\quad \text{s.t. } A_2\boldsymbol{x}_2=\boldsymbol{b}_2,$$

and update the diagonal weighting matrix W_1 as

$$w_{1,ii}^{|j+1|}=\frac{1}{|x_{1,i}^{|j+1|}|+\varepsilon^{|j+1|}}+\frac{\theta\left(|x_{2,i}^{|j+1|}|,\tau\right)}{\theta\left(|x_{2,i}^{|j+1|}|,\tau\right)\cdot|x_{1,i}^{|j+1|}|+\varepsilon^{|j+1|}}.$$

Step 4. Terminate either if both $\boldsymbol{x}_1^{(j)}$ and $\boldsymbol{x}_2^{(j)}$ converge, or we find a sparse enough feasible solution, or we reach the maximum iterations $j_{\max}$. Otherwise, increase j and go back to Step 2.

Similarly, we establish the new iterative re-weighted algorithm for more than two sets of measurements below.

Algorithm 3. New re-weighted algorithm for more vectors:

Step 1. Set $W_k^{(0)}$, $k=1,\ldots,K$ as K identity matrices. Set the iteration count j to zero. Set all the initial estimations of $\boldsymbol{x}_i$ as n dimensional vectors with all the entries equal to 1.

Step 2. Sequentially solve the following ℓ_1 minimization problems for $\boldsymbol{x}_k$, $k=1,\ldots,K-1$

$$\boldsymbol{x}_k^{(j+1)}=\arg\min_{\boldsymbol{x}_k}\|W_k^{(j)}\boldsymbol{x}_k\|_1\quad \text{s.t. } A_k\boldsymbol{x}_k=\boldsymbol{b}_k,$$

and update the diagonal weighting matrix W_{k+1} as

$$w_{k+1,ii}^{(j)}=\frac{1}{|x_{k+1,i}^{(j)}|+\varepsilon^{(j)}}+\left\{\sum_{l=1}^{k}\frac{\theta(|x_{1,i}^{(j+1)}|,\tau)}{\theta(|x_{1,i}^{(j+1)}|,\tau)\cdot|x_{k+1,i}^{(j)}|+\varepsilon^{(j)}}+\sum_{l=k+1}^{K}\frac{\theta(|x_{1,i}^{(j)}|,\tau)}{\theta(|x_{1,i}^{(j)}|,\tau)\cdot|x_{k+1,i}^{(j)}|+\varepsilon^{(j)}}\right\},$$

where $\varepsilon^{(j)}>0$ is a gradually decreasing thresholding sequence. $\tau>0$ is a pre-selected thresholding scalar.

Step 3. Solve the following ℓ_1 minimization problem for $\boldsymbol{x}_K$

$$\boldsymbol{x}_K^{(j+1)}=\arg\min_{\boldsymbol{x}_K}\|W_K^{(j)}\boldsymbol{x}_K\|_1\quad \text{s.t. } A_K\boldsymbol{x}_K=\boldsymbol{b}_K,$$

and update the diagonal weighting matrix W_1 as

$$w_{1,ii}^{|j+1|}=\frac{1}{|x_{1,i}^{|j+1|}|+\varepsilon^{|j+1|}}+\sum_{k=2}^{K}\frac{\theta(|x_{k,i}^{|j+1|}|,\tau)}{\theta(|x_{k,i}^{|j+1|}|,\tau)\cdot|x_{1,i}^{|j+1|}|+\varepsilon^{|j+1|}}.$$

Step 4. Terminate either if all $\boldsymbol{x}_k^{(j)}$, $k=1,\ldots,K$ converge, or we find a sparse enough feasible solution, or we reach the maximum iterations $j_{\max}$. Otherwise, increase j and go back to Step 2.

5. Numerical experiments

In this section, we evaluate the proposed re-weighted algorithm by numerical experiments. In our experiments, the elements of matrices A_k were first drawn independently from a normal distribution $\mathcal{N}(0,1)$. Then, the columns of A_k are normalized such that $\|A_k\|_2=1$. These matrices are known to satisfy the restricted isometry property with optimal bounds with very high probability [9]. Similar to [28–30], the non-zero entries of vectors $\boldsymbol{x}_k$ are Gaussian $\mathcal{N}(0,1)$ i.i.d. random variables.

Fig. 1 shows the (experimentally determined) probability of successful recovery of sparse 256-dimensional vectors $\boldsymbol{x}_k$, with respect to the sparsity value (number of non-zero entries), for the classical and new re-weighted algorithms. In this experiment, we claim the signal being successfully recovered when the ℓ_∞ distance between the real signal and the recovered one is smaller than 10^{-3}. For a group of jointly sparse vectors, we claim this group being successfully recovered, when all the signals are successfully recovered.

The measurements $\boldsymbol{b}$ are 32-dimensional vectors. That is, we have $m=32$, $n=256$. For each sparsity value shown in the plot, we generated 100 pairs of sparse vectors, within each pair, the two vectors have the same sparse structure. ℓ_1 minimization (P_1) and Algorithm I is then applied to recover the signal for all the 200 sparse vectors. Suppose we have p sparse vectors successfully recovered, then the ratio is $p/200$. Meanwhile, we apply Algorithm II to recover the signals for all the 100 pairs of sparse vectors.

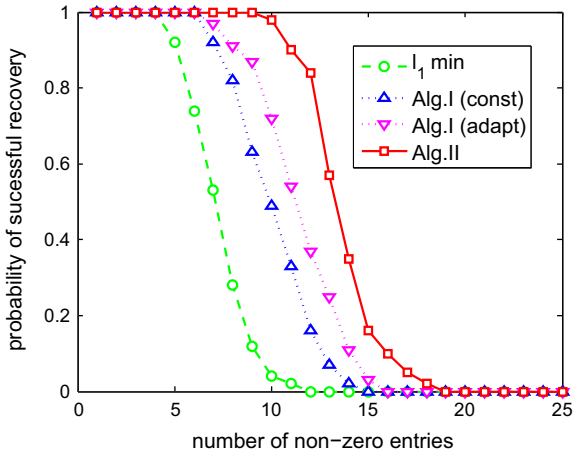


Fig. 1. The (experimentally determined) probability of successful recovery for the classical and new re-weighted algorithms vs. the number of non-zero entries. In figure, ℓ_1-min refers to directly solving Problem (P_1) described in Section 1; Algorithm 1 (const) refers to Algorithm I described in Section 4.1 with constant thresholding scalar $\varepsilon = 0.05$. Algorithm 1 (adap) refers to Algorithm 1 with adaptive thresholding scalar sequence $\varepsilon^{(j)} = \max\{0.5 - 0.05 \cdot j, 0.05\}$. Algorithm 2 refers to Algorithm 2 proposed in Section 4.2 with two sets of measurements and adaptive thresholding scalar sequence $\varepsilon^{(j)} = \max\{0.5 - 0.05 \cdot j, 0.05\}$.

Suppose p pairs of vectors are successfully recovered, then the ratio is $p/100$. The compressive sensing matrices $A \in \mathbb{R}^{m \times n}$ are randomly generated in different attempts. Although we allow different dimensions of the compressive sensing matrices A in the new algorithm, we keep A unchanged for all $\boldsymbol{x}_k$ during one attempt, so as to provide a fair comparison between the classical re-weighted algorithm and the new re-weighted algorithm.

In the first experiment, couple of vectors are jointly sparse, i.e., each couple of vectors have the same support set but have different values. Then one can apply Algorithm II to recover these two vectors together. Meanwhile, the classical recovery algorithms are applicable to recover the vectors one by one. The thresholding scalar is set as $\tau = 0.05$ through the whole tests. The maximum iterations are set as $j_{\max} = 20$ for all algorithms in Fig. 1.

In agreement with [15–18], we can see that the classical iterative re-weighted algorithm leads to notably better recovery performance than directly solving original constrained ℓ_1 minimization Problem (P_1). Furthermore, in accordance with [43,17], we find that the classical iterative re-weighted algorithm with an adaptive thresholding scalar outperforms the one with a constant thresholding scalar.

Moreover, Fig. 1 indicates that the new iterative re-weighted algorithm further improves the probability of successful recovery, by combining multiple sets of measurements. Fig. 2 illustrates the decrease of the 0-norm values of the estimated vectors when applying Algorithm 2. It also demonstrates that the interlaced penalty function (4) forces two solutions to match with each other in sparsity structure. This helps us enhance sparsity solutions when multiple sets of measurements are available.

Fig. 3 compares the proposed re-weighted algorithm and *Rank Awareness Order Recursive Matching Pursuit* (RA-ORMP) algorithm proposed in [25] for MMV. For each

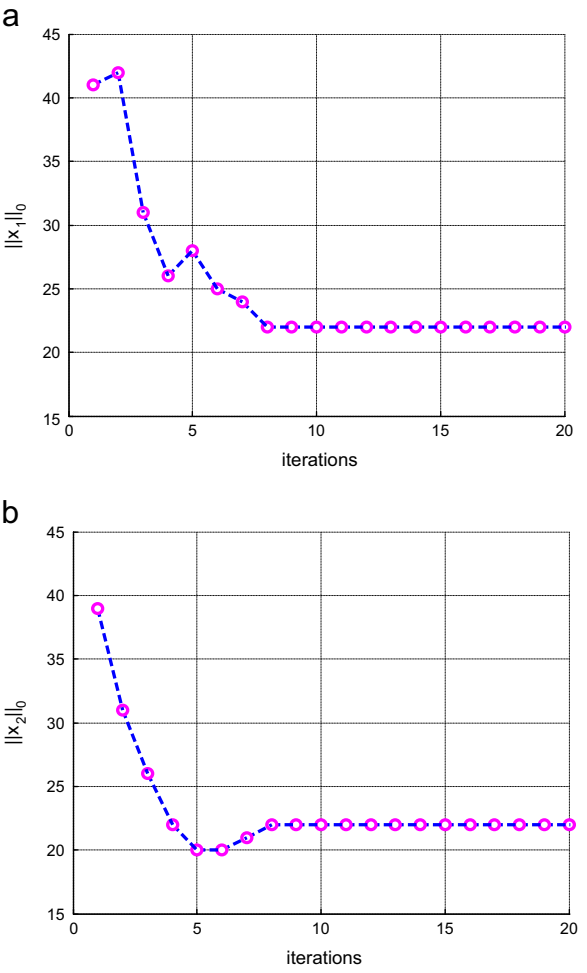


Fig. 2. An illustration on the convergence process of Algorithm 2. (a) $\|\boldsymbol{x}_1\|_0$ in each iteration; (b) $\|\boldsymbol{x}_2\|_0$ in each iteration. Finally, $\|\boldsymbol{x}_1\|_0$ and $\|\boldsymbol{x}_2\|_0$ converge to the same value, implying that the two solutions have the same sparsity structure.

sparsity value shown in the plot, 100 attempts were made to determine the probability of successful recovery. The data dimensions are still $m=32$, $n=256$. In this experiment, the compressive sensing matrices A are randomly generated in different attempts. We keep A unchanged for all $\boldsymbol{x}_k$ during one attempt. The generation details for A and $\boldsymbol{x}$ are the same as those in Fig. 1.

Fig. 3 shows that the recovery rate curves of both the new re-weighted algorithm and RA-ORMP algorithm keep shifting rightward, when more measurements are available. The shift gap of the recovery rate curve for the RA-ORMP algorithm is larger than that for the new re-weighted algorithm.

When only two sets of measurements are available, the new re-weighted algorithm gives better or equal recovery rates when the sparsity value $k < 15$; while the RA-ORMP algorithm gives slightly better recovery rates when the sparsity value $k \geq 15$. When four sets of measurements are available, the new re-weighted algorithm gives better or equal recovery rates when the sparsity value $k < 18$; while the RA-ORMP algorithm gives notably better recovery

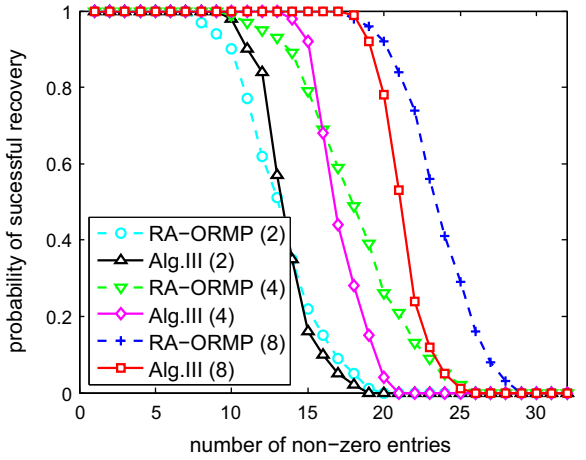


Fig. 3. The (experimentally determined) probability of successful recovery for the new re-weighted algorithm and RA-ORMP algorithm vs. the number of non-zero entries. In this figure, RA-ORMP (2), (4), (8) refer to RA-ORMP algorithm with two, four, and eight sets of measurements, respectively; Algorithm 3 (2), (4), (8) refer to Algorithm 3 proposed in Section 4.2 with two, four, and eight sets of measurements. Algorithm 3 (2) is actually the Algorithm 2, for which we use adaptive thresholding scalar sequence $\varepsilon^{(j)}=\max\{1-0.1\cdot j, 0.05\}$ and $j_{\max}=20$. For Algorithm 3 (4), $\varepsilon^{(j)}=\max\{2-0.1\cdot j, 0.05\}, j_{\max}=25$ and for Algorithm 3 (8), $\varepsilon^{(j)}=\max\{2-0.05\cdot j, 0.05\}, j_{\max}=40$.

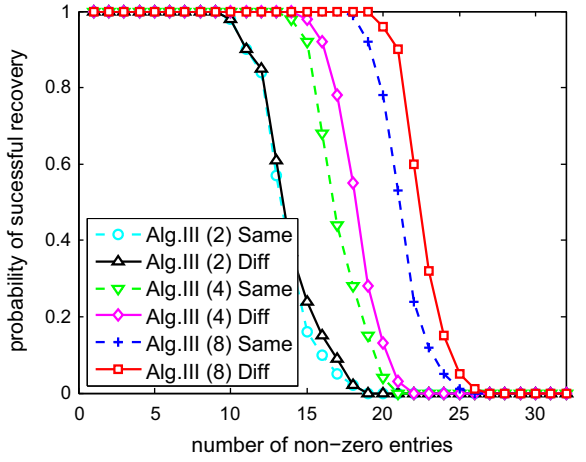


Fig. 4. The (experimentally determined) probability of successful recovery for the new re-weighted algorithm vs. the number of non-zero entries, when different compressive sensing matrices are allowed for different sets of measurements. Algorithm 3 (2), (4), (8) refer to Algorithm 3 proposed in Section 4.2 with two, four, and eight sets of measurements, respectively. Algorithm 3 (2) is actually the Algorithm 2, for which we use adaptive thresholding scalar sequence $\varepsilon^{(j)}=\max\{1-0.1\cdot j, 0.05\}$ and $j_{\max}=20$. For Algorithm 3 (4), $\varepsilon^{(j)}=\max\{2-0.1\cdot j, 0.05\}, j_{\max}=25$ and for Algorithm 3 (8), $\varepsilon^{(j)}=\max\{2-0.05\cdot j, 0.05\}, j_{\max}=40$. "Same" means that the same compressive sensing matrix is used in every attempt; while "Diff." means that we use different compressive sensing matrices in every attempt.

rates when the sparsity value $k\geq 18$. When more than eight sets of measurements are available, the performance of the RA-ORMP algorithm is significantly better than the new re-weighted algorithm. So, if all the measurements corresponds to the same compressive sensing matrix A, we

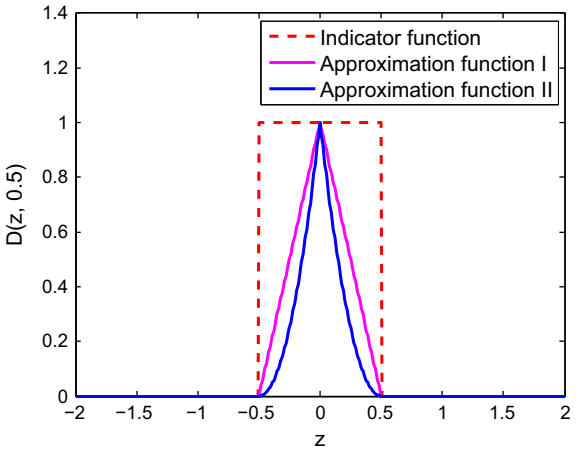


Fig. 5. An illustration of the indicator function and different approximation functions, where $\tau=0.5$.

suggest to adopt the RA-ORMP algorithms e.g., [25] when we have more than four sets of measurements.

However, it should be pointed out that the new re-weighted algorithms allow different matrices A_k, while the RA-ORMP algorithm can only deal with one unique compressive sensing matrix. Tests prove that the recovery rate is larger or at least equal for every sparsity value, if the sets of measurements corresponds to different matrix A_k. As a proof, Fig. 4 plots the recovery rate curves of the proposed re-weighted algorithm when different matrices A_k are used in one attempt. This observation implies that if we can design different measurement matrices A_k, it is possible to have better recovery rate than one single matrix.

One remarkable point is that we only apply the non-continuous indicator function (3) in the above tests. In fact, some smoothed approximations of this indicator function can be utilized; see Fig. 5 for some example. Tests show that the choice of indicator function does not significantly influence the probability of successful recovery, especially when τ is set to be a small value.

6. Conclusions

The main purpose of this paper is to investigate whether we can correctly recover jointly sparse vectors by combining multiple sets of measurements, when the compressive sensing matrices are different. Jointly sparsity assumption holds in many areas, such as sparse filtering and sparse regression. In this paper, we focus on the case of different measure matrices. Different measure matrices exist in distributed systems. Numerical experiments also imply that its recovery performance is better than one single matrix. That means one can design different measure matrices and improve the sparse recovery performance via the proposed algorithm.

By constructing a new constrained ℓ_0 minimization problem, we force the solutions to have the same or at least very similar sparsity structure. Then, a block coordinate descent algorithm and a Majorization–Minimization algorithm are applied to formulate a constrained ℓ_1 minimization problem that can be efficiently solved by

re-weighted procedure. The motivation of the proposed methods mainly depend on the simplicity, speed, and stability.

Numerical testing results show that the new iterative re-weighted algorithm can effectively recover sparse signals when different measurement matrices are involved. The performance sheds more lights on solving the sparsity-constrained problems when all the solutions are structurally related. The proposed algorithm helps us to design different measure matrices, which can improve the recovery performance.

References

[1] M. Elad, Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing, Springer, New York, 2010.
[2] Y.C. Eldar, G. Kutyniok, Compressed Sensing: Theory and Applications, Cambridge University Press, Cambridge, 2012.
[3] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B. Methodol. 58 (1996) 267–288.
[4] A.M. Bruckstein, D.L. Donoho, M. Elad, From sparse solutions of systems of equations to sparse modeling of signals and images, SIAM Rev. 51 (2009) 34–81.
[5] S. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries, IEEE Trans. Signal Process. 41 (1993) 3397–3415.
[6] D. Needell, R. Vershynin, Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit, IEEE J. Sel. Topics Signal Process. 4 (2010) 310–316.
[7] S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, SIAM J. Sci. Comput. 20 (1998) 33–61.
[8] D. Donoho, M. Elad, Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization, Proc. Natl. Acad. Sci. 100 (2003) 2197–2202.
[9] E.J. Candès, T. Tao, Decoding by linear programming, IEEE Trans. Inf. Theory 51 (2005) 4203–4215.
[10] M. Schmidt, G. Fung, R. Rosales, Optimization Methods for ℓ_1−regularization, Technical Report, University of British Columbia, 2009.
[11] A. Yang, A. Ganesh, S. Sastry, Y. Ma, Fast ℓ_1−minimization algorithms and an application in robust face recognition: a review, in: Proceedings of IEEE International Conference on Image Processing, 2010, pp. 1849–1852.
[12] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, Cambridge, 2004.
[13] D.L. Donoho, De-noising by soft thresholding, IEEE Trans. Inf. Theory 41 (1995) 613–627.
[14] I. Daubechies, M. Defrise, C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Commun. Pure Appl. Math. 57 (2004) 1413–1457.
[15] I. Gorodnitsky, B. Rao, Sparse signal reconstruction from limited data using FOCUSS: a reweighted minimum norm algorithm, IEEE Trans. Signal Process. 45 (1997) 600–616.
[16] E.J. Candès, M.B. Wakin, S.P. Boyd, Enhancing sparsity by reweighted ℓ_1 minimization, J. Fourier Anal. Appl. 14 (2008) 877–905.
[17] D. Wipf, S. Nagarajan, Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions, IEEE J. Sel. Topics Signal Process. 4 (2010) 317–329.
[18] Q. Lyu, Z. Lin, Y. She, C. Zhang, A comparison of typical l_p minimization algorithm, Neurocomputing 119 (2013) 413–424.
[19] S.F. Cotter, B.D. Rao, K. Engan, K. Kreutz-Delgado, Sparse solutions to linear inverse problems with multiple measurement vectors, IEEE Trans. Signal Process. 53 (2005) 2477–2488.
[20] J.A. Tropp, Algorithms for simultaneous sparse approximation, Part II: convex relaxation, Signal Process. 86 (2006) 589–602.
[21] J. Chen, X.M. Huo, Theoretical results on sparse representations of multiple-measurment vectors, IEEE Trans. Signal Process. 54 (2006) 4634–4643.
[22] G. Tang, A. Nehorai, Performance analysis for sparse support recovery, IEEE Trans. Inf. Theory 56 (2010) 1383–1399.
[23] J.M. Kim, O.K. Lee, J.C. Ye, Compressive MUSIC: revisiting the link between compressive sensing and array signal processing, IEEE Trans. Inf. Theory 58 (2012) 278–301.
[24] J. Ziniel, P. Schniter, Efficient high-dimensional inference in the multiple measurement vector problem, IEEE Trans. Signal Process. 61 (2013) 340–354.
[25] M. Davies, Y.C. Eldar, Rank awareness in joint sparse recovery, IEEE Trans. Inf. Theory 58 (2012) 1135–1146.
[26] K. Lee, Y. Bresler, M. Junge, Subspace methods for joint sparse recovery, IEEE Trans. Inf. Theory 58 (2012) 3613–3641.
[27] J. Blanchard, M.E. Davies, Recovery guarantees for rank aware pursuits, IEEE Signal Process. Lett. 19 (2012) 427–430.
[28] M.E. Tipping, Sparse Bayesian learning and the relevance vector machine, J. Mach. Learn. Res. 1 (2001) 211–244.
[29] S. Ji, Y. Xue, L. Carin, Bayesian compressive sensing, IEEE Trans. Signal Process. 56 (2008) 2346–2356.
[30] S. Ji, D. Dunson, L. Carin, Multitask compressive sensing, IEEE Trans. Signal Process. 57 (2009) 92–106.
[31] Q. Ling, Z. Wen, W. Yin, Decentralized jointly sparse optimization by reweighted minimization, IEEE Trans. Signal Process. 61 (2013) 1165–1170.
[32] Z.Q. Luo, P. Tseng, On the convergence of the coordinate descent method for convex differentiable minimization, J. Optim. Theory Appl. 72 (1992) 7–35.
[33] D.P. Bertsekas, Nonlinear Programming, 2nd ed. Athena Scientific, Belmont, MA, 1999.
[34] T.T. Wu, K. Lange, Coordinate descent algorithms for Lasso Penalized regression, Ann. Appl. Stat. 2 (2008) 224–244.
[35] Y. Li, S. Osher, Coordinate descent optimization for ℓ_1 minimization with applications to compressed sensing: a greedy algorithm, Inverse Probl. Imaging 3 (2009) 487–503.
[36] R. Mazumder, J.H. Friedman, T. Hastie, SparseNet: coordinate descent with nonconvex penalties, J. Am. Stat. Assoc. 106 (2011) 1125–1138.
[37] Z. Wen, D. Goldfarb, K. Scheinberg, Block coordinate descent methods for semidefinite programming, in: M.F. Anjos, J.B. Lasserre (Eds.), Handbook on Semidefinite, Cone and Polynomial Optimization: Theory, Algorithms, Software and Applications, Springer, New York, 2012.
[38] D.R. Hunter, K. Lange, A tutorial on MM algorithms, Am. Stat. 58 (2004) 30–37.
[39] K. Lange, Optimization, Sringer-Verlag, New York, 2004.
[40] Z. Zhang, J.T. Kwok, D.-Y. Yeung, Surrogate maximization/minimization algorithms and extensions, Mach. Learn. 69 (2007) 1–33.
[41] J. Mairal, Optimization with first-order surrogate functions, in: Proceedings of International Conference on Machine Learning, 2013, pp. 783–791.
[42] B. Rao, K. Engan, S.F. Cotter, J. Palmer, K. Kreutz-Delgado, Subset selection in noise based on diversity measure minimization, IEEE Trans. Signal Process. 51 (2003) 760–770.
[43] R. Chartrand, W. Yin, Iteratively reweighted algorithms for compressive sensing, in: Proceedings of International Conference on Acoustics, Speech, Signal Processing, 2008, pp. 3869–3872.