On the convergence of FOCUSS algorithm for sparse representation

Zhaozhi He, Shengli Xie, Senior Member, IEEE, and Andrzej Cichocki, Senior Member, IEEE

Abstract—FOCal Underdetermined System Solver (FOCUSS) is a powerful tool for sparse representation and underdetermined inverse problems, which is extremely easy to implement. In this paper, we provide a comprehensive convergence analysis on the FOCUSS algorithm towards establishing a systematic convergence theory for it. First, we give a rigorous derivation for this algorithm exploiting the auxiliary function. Then, we prove its convergence. In particular, we systematically analyze its convergence rate for different sparsity parameter $p$ and demonstrate its convergence rate by numerical experiments.

Index Terms—FOCUS algorithm, convergence, convergence rate, Auxiliary function, compressive sensing, superlinear convergence, linear convergence, global convergence theorem

I. INTRODUCTION

The problem of finding sparse solutions to underdetermined linear problems from limited data arises in many applications, including compressive sensing/compressive sampling [1]–[4], biomagnetic imaging problem [5], spectral estimation, direction-of-arrival (DOA), signal reconstruction, [6]–[8], and so on. Mathematically, this problem is to solve the following combinatorial optimization problem [9]–[12]:

$$\min_{x} \|s\|_0 \quad \text{subject to } x = As,$$

where $\|s\|_0$ denotes the number of nonzero components in $s$, $x = (x_1, \cdots, x_m)^T \in \mathbb{R}^m$ is an observed vector, $A = [a_1, \cdots, a_n] \in \mathbb{R}^{m \times n}$ is a known basis matrix $(m < n)$, $s = (s_1, \cdots, s_n)^T \in \mathbb{R}^n$ is an unknown vector which represents $n$ sparse sources or hidden sparse components, and $m$ is the number of observations. The objective is to estimate the sources $s$ such that $s$ is as sparse as possible in the sense that most of components of $s$ are zeros or approximate to zeros [5]–[9], [13]–[22].

In general, it is very difficult to directly solve the combinatorial problem (1) if its dimension is high. Measuring the sparsity by $\ell_p$ diversity, instead we usually consider its approximate optimization problem [5]–[9], [14], [17], [21]–[24]:

$$\begin{align*}
    \min_{s} \hat{F}(s) &= \min_{s} \sum_{i=1}^{n} |s_i|^p \\
    \text{subject to: } x &= As,
\end{align*}$$

where $0 < p < 2$. Much attention has been paid to this problem and many algorithms have been developed for it, especially for the special case $p = 1$ [9]–[11], [25], for example, linear programming (LP) [10], [16], [18], [20], [26], [27], basis pursuit (BP) [18], various greedy algorithms (e.g., shortest path decomposition [26], [28], $\ell_p$-BP with $0 < p < 1$ [29], MP and OMP [12], [30], [31]), least squares methods with $\ell_1$ regularization (e.g., PDCO-LSQR [32], Homotopy [33], TNIIPM [2], etc) and FOCUSS algorithm(s) [7], [19], [34]–[36].

Among them, LP and BP are time-consuming, the accuracy of MP and OMP, which are fast but just can achieve an approximate/rough solution to (2), is usually worse than the others however, $\ell_p$-BP [29] is $NP$-hard and requires a lot of storage space. So the LP, BP and $\ell_p$-BP are not suitable for large scale problems. The least squares methods with $\ell_1$ regularization can be used to solve large scale problem potentially; but, the regularization parameters for imposing the sparseness constraint must be given in advance subjectively. Generally speaking, it is not easy to properly set the optimal sparseness regularization parameters. In contrast, the FOCUSS algorithms, developed originally by Gorodnitsky, Rao et al [5]–[8], [14], [17], [21], are not only very efficient in finding a precise solution for the $\ell_p$-spars representation problem (2) but also have no regularization parameters to set, which are extremely easy to implement. Moreover, they are advantageous in terms of the computational complexity, and they are suitable even for large scale problems [37].

The standard FOCUSS algorithm can be addressed as follows:

$$s^{(t+1)} = \Pi^{-1}(s^{(t)}) \cdot A^T \cdot [A \cdot \Pi^{-1}(s^{(t)}) \cdot A^T]^{-1} \cdot x,$$

where $t = 0, 1, \cdots, +\infty$ and

$$\Pi(s^{(t)}) = \begin{bmatrix}
    |s_1^{(t)}|^{-p+2} & 0 & \cdots & 0 \\
    0 & |s_2^{(t)}|^{-p+2} & \cdots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
    0 & 0 & \cdots & |s_n^{(t)}|^{-p+2}
\end{bmatrix}.$$
Or, it can be equivalently implemented by three steps [7] as
\[
W^{(t+1)} = \text{diag} \left( |s_1^{(t)}|^{1-\frac{p}{2}}, \ldots , |s_n^{(t)}|^{1-\frac{p}{2}} \right)
\]
\[
q^{(t+1)} = [A^{(t+1)}]^T x, \text{ where } A^{(t+1)} = AW^{(t+1)}
\]
\[
s^{(t+1)} = W^{(t+1)} q^{(t+1)}.
\]
For simplicity, the overall procedure for finding a sparse solution \(s^{(*)}\) by FOCUSS can be notated as
\[
s^{(*)} = \text{FOCUSS}(x, A, s^{(0)}, \text{num\_iter}),
\]
where \(s^{(0)}\) is an initialization and \(\text{num\_iter} \) is the pre-specified number of iterations.

Rao et al [7] proved by the generalized Hölder inequality that given \(s^{(0)} \neq 0\), the cost function \(\tilde{F}(s)\) is monotonically nonincreasing on the sequence \(\{\tilde{F}(s^{(t)})\}_{t=0}^{\infty}\) obtained by (3). Furthermore, based on the global convergence theorem (GCT), they proved that the limit of any convergent subsequence of \(\{s^{(t)}\}_{t=0}^{\infty}\) is a stationary point of (4).

Following Gorodnitsky, Rao et al’s pioneering works [5–8], [14], [17], [21], in this paper, we further strengthen the FOCUSS algorithm theoretically and develop much stronger convergence results towards establishing a systematic convergence theory for it, in which the auxiliary function plays an essential role [38].

The rest of this paper is organized as follows. Section III states some mild assumptions. A rigorous derivation of the FOCUSS algorithm is given in Section IV. The convergence of FOCUSS algorithm is proved in Section V. Section VI discusses how the FOCUSS algorithm is related with the Newton method for \(l_p\)-optimization problem (2). The convergence rate of FOCUSS is investigated in Section VII. The discusions and conclusions are given in VIII and IX respectively.

II. SOME ASSUMPTIONS

The FOCUSS iterative formula (3) can be rewritten as
\[
s^{(t+1)} = \Pi^{-1}(s^{(t)}) \cdot c^{(t)} = \text{diag}(c^{(t)}) \cdot \left[ |s_1^{(t)}|^{2-p} \right. \left. \vdots \right. |s_n^{(t)}|^{2-p} \right]\]
where \(c^{(t)} = A^T \cdot [A \cdot \Pi^{-1}(s^{(t)}) \cdot A^T]^{-1} \cdot x\). Noting that \(\Pi^{-1}(s^{(t)}) = \text{diag}(|s_1^{(t)}|^{2-p}, \ldots , |s_n^{(t)}|^{2-p})\) is a diagonal matrix, \(\forall i\), we have \(s_i^{(t+1)} = 0\) if \(s_i^{(t)} = 0\), i.e., \(s^* = 0\) is a stationary point of (3) regardless of whether it is an optimal solution or not. Thus, given \(s^{(0)} = 0\), the FOCUSS algorithm is convergent and converges to zero. In addition, if \(x = 0\), the FOCUSS algorithm (3) will directly find the sparsest solution \(s^*\). Without loss of generality, in this paper we investigate the convergence issues of the FOCUSS algorithm under the following assumptions:

Assumption 1. \(x \neq 0\);

Assumption 2. For \(A = [a_1, \cdots , a_m] \in \mathbb{R}^{m \times n}\) \((m < n)\), any \(m\) columns of \(A\) are linearly independent;

Assumption 3. The initializations \(s^{(0)} = [s_1^{(0)}, \cdots , s_n^{(0)}]^T\) used for the FOCUSS algorithm are entrywisely/strictly nonzero, i.e., \(s_i^{(0)} \neq 0\) for \(i = 1, \cdots , n\).

III. A ROUGER DERIVATION FOR FOCUS ALGORITHM

The FOCUSS algorithm was derived and justified in [5–8], [14], [17], [21]. However, as will be explained in detail, the proof is not rigorous [37]. In this section, we propose a rigorous derivation for FOCUSS by constructing an auxiliary function, which sheds light on how the FOCUSS algorithm decreases the cost function during iterations.

A. The existing derivation for FOCUS algorithm

The Lagrange multiplier method was employed to solve problem (2) in [7], in which the Lagrange function is
\[
L(s, \alpha) = \tilde{F}(s) + \alpha^T (As - x),
\]
where \(\alpha\) is an \(m \times 1\) vector of the Lagrange multipliers. A necessary condition for the solution \(s^*\) to exist is that \((s^*, \alpha^*)\) is a stationary point of the Lagrange function, i.e.:
\[
\begin{align*}
\frac{\partial L(s, \alpha)}{\partial s} &= \frac{\partial \tilde{F}(s)}{\partial s} + A^T \alpha = 0
\end{align*}
\]
where \(\frac{\partial \tilde{F}(s)}{\partial s} = p \cdot \Pi(s) \cdot s\) [7], [21]. Solving the equations set (5), we can derive the FOCUSS equations as follows (see [7] and [37]):
\[
\begin{align*}
s &= \Pi^{-1}(s) \cdot A^T \cdot [A \cdot \Pi^{-1}(s) \cdot A^T]^{-1} \cdot x.
\end{align*}
\]
Replacing \(s\) with \(s^{(t)}\) on the right side of (6), we have the FOCUSS iterate (5).

However, as mentioned previously, theoretically the derivation above for (3) is not rigorous [37]. Note that the equations set (5) does not hold when \(0 < p < 1\) because some components of \(s\) can be zeros. To be precise, the matrix \(\Pi(s)\) does not exist in this case although the matrix \(\Pi^{-1}(s)\) does, because \(0^p \to \infty\). In order to solve this problem, next we propose a new derivation for (3) available for \(0 < p < 2\).

B. A new derivation for FOCUSS algorithm

Let’s start from the concept auxiliary function [38].

Definition 1. A function \(f(s_i|s^{(t)})\) with respect to \(s\) is said to be an auxiliary function to \(F(s)\) if \(f(s_i|s^{(t)}) \geq F(s)\) and \(f(s^{(t)}|s^{(t)}) = F(s^{(t)})\).

From Definition 1 carrying out some simple manipulations, we have the following lemma.

Lemma 1. Let
\[
F(s_i) = |s_i|^p
\]
and
\[
f(s_i|s^{(t)}) = \frac{p}{2} |s_i^{(t)}|^{p-2} \cdot s_i^2 + (1 - \frac{p}{2}) |s_i^{(t)}|^{p},
\]
where \(0 < p < 2\). Then \(f(s_i|s^{(t)})\) is an auxiliary function to \(F(s_i)\), i.e., \(\forall s_i\), \(f(s_i|s_i^{(t)}) \geq F(s_i)\), and \(f(s_i^{(t)}|s_i^{(t)}) = F(s_i^{(t)})\).

Proof: As showing in Fig 1 we can readily prove it by verifying two conditions of Definition 1.
The necessary condition for the solution of quadratic optimization problem \( F(s) \) with respect to \( F(s) = |s|^p \) such that \( f(s)|s(t)| \geq F(s) \) and \( f(s)|s(t)| = F(s) \),

\[
\begin{aligned}
\partial \tilde{F}(s) &= \sum_{i=1}^{n} F(s_i) \\
\tilde{f}(s|s(t)) &= \sum_{i=1}^{n} f(s_i|s(t)) ,
\end{aligned}
\]

\( \tilde{f}(s|s(t)) \) is also an auxiliary function to \( \tilde{F}(s) \). Instead of the \( \ell_p \) optimization problem \( \hat{F} \), now we consider its corresponding auxiliary optimization problem as follows:

\[
\begin{aligned}
\min_{s} \tilde{f}(s|s(t)) &= \min_{s} \sum_{i=1}^{n} \frac{p}{p-1} |s_i|^{p-1} \cdot (1 - \frac{p}{s_i}) \cdot |s_i|^p \\
\text{subject to : } x = As
\end{aligned}
\]

(9)

**Theorem 1.** If \( s(t) \) is entrywisely nonzero, i.e., \( |s_i(t)| > 0 \), the FOCUSS iterate \( s(t+1) \) obtained by (3) is the globally optimal solution of quadratic optimization problem (9), which satisfies \( As(t+1) = x \) and

\[
\tilde{f}(s(t)|s(t+1)) \leq \tilde{f}(s(t)|s(t)) = \tilde{F}(s(t)).
\]

(10)

**Proof:** For problem (9), we can construct the following Lagrange function

\[
\tilde{L}(s, \alpha) = \tilde{f}(s|s(t)) + \alpha^T(As - x).
\]

(11)

The necessary condition for the solution \( s^* \) to exist is that \( (s^*, \alpha^*) \) is a stationary point of the Lagrange function, i.e.,

\[
\begin{aligned}
\partial \tilde{L}(s, \alpha) &= \partial \tilde{f}(s|s(t)) + A^T \alpha = 0 \\
\partial \tilde{L}(s, \alpha) &= As - x = 0
\end{aligned}
\]

(12)

where \( \partial \tilde{f}(s|s(t)) = p \cdot \Pi(s(t)) \cdot s \). Noting that the vector \( s(t) \) is entrywisely nonzero, we can compute \( \Pi(s(t)) \). Thus, we can further obtain

\[
\tilde{f}(s|s(t)) = \sum_{i=1}^{n} f(s_i|s(t)) = \sum_{i=1}^{n} \frac{p}{p-1} \cdot \Pi^{-1}(s(t)) \cdot A^T \alpha = 0
\]

(13)

Substituting (13) into (12), we can immediately obtain the solution of problem (9) as follows:

\[
s = \Pi^{-1}(s(t)) \cdot A^T \cdot [A \cdot \Pi^{-1}(s(t)) \cdot A^T]^{-1} \cdot x.
\]

Thus, letting

\[
s(t+1) = \Pi^{-1}(s(t)) \cdot A^T \cdot [A \cdot \Pi^{-1}(s(t)) \cdot A^T]^{-1} \cdot x,
\]

we have \( As(t+1) = x \) and

\[
\tilde{f}(s(t+1)|s(t)) \leq \tilde{f}(s(t)|s(t)) = \tilde{F}(s(t)).
\]

(14)

On the other side, \( \tilde{f}(s|s(t)) \) is an auxiliary function to \( \tilde{F}(s) \). So \( \tilde{f}(s(t)|s(t)) = \tilde{F}(s(t)) \). Therefore, we have \( \tilde{f}(s(t+1)|s(t)) \leq \tilde{F}(s(t)) \).

**Theorem 2.** Given an entrywisely nonzero initialization \( s(0) \), the iterative sequence \( \{ \tilde{F}(s(t)) \} \to \infty \) obtained by (3) is convergent.

**Proof:** Since \( \tilde{f}(s|s(t)) \) is an auxiliary function to \( \tilde{F}(s) \), \( \forall s \), we have

\[
\tilde{f}(s|s(t)) \geq \tilde{F}(s) \implies \tilde{F}(s(t)) \leq \tilde{f}(s(t)|s(t)).
\]

Combing (10) with (14), we have the following iterative inequalities

\[
0 \leq \tilde{F}(s(t)) \leq \tilde{f}(s(t)|s(t)) \leq \tilde{f}(s(t)|s(t+1)) \leq \tilde{F}(s(t+1)) \leq \tilde{f}(s(t+1)|s(t+1)) \leq \tilde{F}(s(t+1)),
\]

i.e., the sequence \( \{ \tilde{F}(s(t)) \} \to \infty \) is monotonically non-increasing and bounded. Accordingly, \( \{ \tilde{F}(s(t)) \} \to \infty \) is convergent.

**C. An extended FOCUSS algorithm for the more general sparsity measure**

It is worth noting that the FOCUSS algorithm (3) can be straightforwardly extended for a general sparse representation problem as follows:

\[
\min_{s} \sum_{i=1}^{n} F(|s_i|) \quad \text{subject to : } x = As
\]

(15)
where $F(s)$ is a more general sparsity measure than $\ell_p$-norm such that $F''(s) < 2$ for $s > 0$, e.g., $F(|s|) = \ln |s|$ and $F(|s|) = -|s|^p$ ($p < 0$). However, in this case $\Pi(s)$ is replaced with $\Pi(s) = \text{diag} \left[ \frac{F'(s_1)}{2|s_1|}, \ldots, \frac{F'(s_n)}{2|s_n|} \right]$ in (3). And we can analogously prove it by constructing the following optimization problem

$$\min_{s} \sum_{i=1}^{n} f(s_i|s_i^{(t)})$$

subject to: $x = As$

where the auxiliary function is

$$f(s_i|s_i^{(t)}) = \frac{F'(s_i^{(t)})}{2s_i^{(t)}} \cdot s_i^2 + \frac{F'(s_i^{(t)})}{2} \cdot s_i^{(t)}.$$ 

IV. THE CONVERGENCE OF FOCUSS ALGORITHM

Let’s begin with a lemma.

**Lemma 2.** Let $A = [a_1| \cdots |a_n] \in \mathbb{R}^{m \times n}$ such that $m \leq n$. If $0 < p \leq 1, \text{ the solution } s^* = [s_1^*, \cdots, s_n^*]^T$ of $\ell_p$ problem (2) has $k \leq m$ nonzero components, i.e., $\# \{s^* \} \leq m$, where $\# \{ \cdot \}$ denotes the number of nonzero components of a set.

**Proof:** For $p = 1$ and $0 < p < 1$, this lemma had been proven in [26] and [29], respectively.

Without loss of generality, suppose the nonzero components of $s$ are $s_{n_1}, \cdots, s_{n_m}$. Denote $\Omega_N = \{n_1, \cdots, n_m\} \subset \{1, \cdots, n\}$ and $\Omega_O = \{1, \cdots, n\} \backslash \Omega_N$ as the subscripts of nonzero and zero components, respectively. $A_N = [a_{n_1}| \cdots |a_{n_m}]$, $A_O$ is the remaining matrix after removing the columns of $A_N$ from $A$. By analogy, $s_N = [s_{n_1}| \cdots |s_{n_m}^T]$ and $s_O = [0| \cdots |0]$ are the corresponding nonzero and zero parts of $s$, respectively.

**Theorem 3.** Let $A = [a_1| \cdots |a_n] \in \mathbb{R}^{m \times n}$ satisfying Assumption 2. For $1 < p < 2$ and $x \neq 0$, the $\ell_p$ problem (2) is non-concave and its solution $s^* = [s_1^*, \cdots, s_n^*]^T$ is such that $\# \{i : s_i^* = 0\} \leq m - 1$, i.e., $\# \{s^* \} \geq n - m + 1$.

**Proof:** Note that $A = [a_1| \cdots |a_n] \in \mathbb{R}^{m \times n}$ satisfies Assumption 2 and rank($A$) = $m$. Hence, there exists a matrix $C = [c_1| \cdots |c_{n-m}] \in \mathbb{R}^{m \times (n-m)}$ such that $\text{rank}(C) = n - m$ and $AC = 0_{m \times (n-m)}$, i.e., $O(A^T) = \text{span}(c_1, \cdots, c_{n-m})$. Given

$$s^0 = A^+ x = A^T (AA^T)^{-1} x,$$

we have $As^0 = x$. Then we can derive

$$A(s^0 + C\lambda) = x,$$

where $\lambda \in \mathbb{R}^{n-m}$. Subsequently, the equalities constrained problem (2) is equivalent to the following unconstrained optimization problem

$$\min_{\lambda} F(\lambda) = \min_{\lambda} \sum_{i=1}^{n} |s_i|^p,$$

where $s = (s_1, \cdots, s_n)^T = s^0 + C\lambda$. Then we can obtain

$$\frac{\partial F}{\partial \lambda} = p \cdot C^T \cdot \begin{bmatrix} |s_1|^{p-1} \cdot \text{sign}(s_1) \\ \vdots \\ |s_n|^{p-1} \cdot \text{sign}(s_n) \end{bmatrix}$$

and

$$H_\lambda = \frac{\partial^2 F}{\partial \lambda^2} = p(p-1) \cdot C^T \cdot \begin{bmatrix} |s_1|^{p-2} & 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & |s_n|^{p-2} \end{bmatrix} \cdot C.$$

For $1 < p < 2$, $H_\lambda$ is nonnegative-definite.

Suppose $s^*$ is the solution of problem (2), then it is a stationary point of $L(s, \alpha)$ with respect to $s$, i.e.,

$$\frac{\partial L(s^*, \alpha)}{\partial s} = \frac{\partial F(s^*)}{\partial s} + A^T \alpha^* = 0$$

$$= p \cdot \begin{bmatrix} |s_1|^{p-1} \cdot \text{sign}(s_1) \\ |s_2|^{p-1} \cdot \text{sign}(s_2) \\ \vdots \\ |s_n|^{p-1} \cdot \text{sign}(s_n) \end{bmatrix} + A^T \alpha^* = 0,$$

where $L(s, \alpha)$ is given in (4) and

$$\alpha^* = [A \cdot \Pi^{-1}(s^*) \cdot A^T]^{-1} \cdot x \neq 0.$$ (19)

Now we prove it by contradiction. Suppose $\# \{i : s_i^* = 0\} = \# \{a_1, \cdots, a_k\} = k > m-1$, i.e., $s_{a_1} = 0, \cdots, s_{a_k} = 0$,

Then we have

$$p \cdot \begin{bmatrix} |s_{a_1}|^{p-1} \cdot \text{sign}(s_{a_1}) \\ |s_{a_2}|^{p-1} \cdot \text{sign}(s_{a_2}) \\ \vdots \\ |s_{a_k}|^{p-1} \cdot \text{sign}(s_{a_k}) \end{bmatrix} + \begin{bmatrix} a_{a_1}^T \\ a_{a_2}^T \\ \vdots \\ a_{a_k}^T \end{bmatrix} \alpha^* = 0 \implies \begin{bmatrix} a_{a_1}^T \\ a_{a_2}^T \\ \vdots \\ a_{a_k}^T \end{bmatrix} = 0,$$

where $A_O = [a_1| \cdots |a_k] \in \mathbb{R}^{m \times k}$ ($k \geq m$) and rank($A_O$) = $m$. From $A_O \cdot \alpha^* = 0$, we can derive $\alpha^* = (A_O \cdot A_O^T)^{-1} A_O \cdot \alpha^* = 0$, which contradicts (19). Hence, we have $\# \{i : s_i^* = 0\} \leq m - 1$, equivalently, $\# \{s^* \} \geq n - m + 1$.

Denote

$$g_i(s) = \Pi_i^{-1}(s) \cdot A^T \cdot [A \cdot \Pi^{-1}(s) \cdot A^T]^{-1} \cdot x,$$

where $i = 1, \cdots, n$, and then

$$g(s) = [g_1(s) \cdots g_n(s)]^T = G(s) \cdot s.$$

Thus, the FOCUSS iterate (3) can be simply written as $s^{(t+1)} = g(s^{(t)})$.

**Lemma 3.** For $0 < p < 2$, supposing $A$ and $x$ are such that Assumption 1 and Assumption 2 the FOCUSS iterative formulas (3) has at most a finite number of stationary points or fixed points $s^*$ in the sense that they satisfy $s^* = g(s^*)$.

**Proof:** Suppose $A$ and $x$ are such that Assumption 1 and Assumption 2.

For $0 < p \leq 1$, following the proof of Lemma 2 in [29] and [26], for every possible stationary point $s^*$, we can derive $\# \{s^* \} \leq m$ where $s^* = [s_N^* \cdots s_O^*] = [s_N^* 0]$. 
there exist at most a finite number of stationary points.

For $1 < p < 2$ and an arbitrary possible stationary point $s^*$, $\#\{s^*\} = \#\{s_N^*\}$. Noting that in (IV) the Hessian matrix $H_{XX}(s_N^*)$ is strictly positive-definite with respect to $s_N^*$, which indicates that the $\ell_p$ optimization problem (2) has a unique solution with respect to $s_N^*$. Since $s^* = \left[ \begin{array}{c} s_N^* \\ s_0^* \end{array} \right]$, we can derive that the FOCUSS iterative formulas (3) has at most a finite number of stationary points for $1 < p < 2$.

In summary, the FOCUSS iterative formulas (3) has at most a finite number of stationary points for $0 < p < 2$.

**Theorem 4.** For $0 < p < 2$, supposing $A$, $x$ and $s(0)$ are such that Assumption [7] Assumption [2], the FOCUSS algorithm converges, i.e., the iterative sequence $\{s(t)\}_{t=0}^{\infty}$ obtained by (3) is convergent.

**Proof:**

Given $0 < p < 2$, the function

$$q(z) = \frac{p}{2} \cdot \frac{1}{z^{2-p}} + (1 - \frac{p}{2}) \cdot z^p - 1 \geq 0$$

for $z > 0$ and it reaches its unique minimum 0 in $z \in (0, +\infty)$ at $z^* = 1$, i.e.,

$$q(z^*) = 0 \text{ if and only if } z^* = 1$$

because

$$q'(z) = -\frac{p(2 - p)}{2} \cdot \frac{1}{z^{3-p}} - (1 - 2z)$$

$$\Delta(t) \triangleq f(s(t+1)|s(t)) - F(s(t+1))$$

$$= \sum_{i=1}^{n} f(s_i(t+1)|s_i(t)) - \sum_{i=1}^{n} F(s_i(t+1))$$

$$= \sum_{i=1}^{n} \left[ f(s_i(t+1)|s_i(t)) - F(s_i(t+1)) \right]$$

$$= \sum_{i=1}^{n} \Delta_i(t),$$

where

$$\Delta_i(t) = f(s_i(t+1)|s_i(t)) - F(s_i(t+1))$$

$$= \frac{p}{2} \cdot |s_i(t)|^{p-2} \cdot |s_i(t+1)|^2 + (1 - \frac{p}{2}) \cdot |s_i(t)|^p - |s_i(t+1)|^p$$

$$= q \left( \frac{|s_i(t)|}{|s_i(t)|} \right) \cdot |s_i(t+1)|^p$$

From Lemma [11], $f(s_i|s_i(t))$ is an auxiliary function to $F(s_i)$. Hence, we have

$$\Delta_i(t) \geq 0 \implies 0 \leq \Delta_i(t) \leq \sum_{i=1}^{n} \Delta_i(t) = \Delta(t).$$

Also, from (15), we have

$$0 \leq |\Delta(t)| = \overline{F}(s(t)) - \overline{F}(s(t+1)).$$

From Theorem [2], $\{\overline{F}(s(t))\}_{t=0}^{\infty}$ is convergent, i.e.,

$$\overline{F}(s(t)) - \overline{F}(s(t+1)) \rightarrow 0 \text{ as } t \rightarrow +\infty.$$  \hspace{1cm} (23)

Combining (22) with (23), we have

$$0 \leq |\Delta(t)| \rightarrow 0$$

Then, we have two options: $q \left( \frac{|s_i(t)|}{|s_i(t+1)|} \right) \rightarrow 0$ or $|s_i(t+1)|^p \rightarrow 0$. Additionally, $\{s_i(t+1)\}_{t=0}^{\infty}$ is bounded because $\{\overline{F}(s(t))\}_{t=0}^{\infty}$ is convergent (see Theorem 3). Next, we separately prove that $\{s_i(t)\}_{t=0}^{\infty}$ is convergent in both cases.

1) In the first case, from (20), we can derive

$$q \left( \frac{|s_i(t)|}{|s_i(t+1)|} \right) \rightarrow 0 \implies \frac{|s_i(t)|}{|s_i(t+1)|} \rightarrow 1$$

So $\{s_i(t)\}_{t=0}^{\infty}$ is convergent in this case.

2) In the second case, as $t \rightarrow +\infty$ we have

$$|s_i(t+1)|^p \rightarrow 0 \implies |s_i(t+1)| \rightarrow 0 \implies |s_i(t+1) - s_i(t)| \rightarrow 0,$$

i.e., also $|s_i(t+1) - s_i(t)| \rightarrow 0$ when $|s_i(t+1)| \rightarrow 0$.

As a consequence, in both cases $|s_i(t+1) - s_i(t)| \rightarrow 0$ as $t \rightarrow +\infty$, $i = 1, \ldots, n$. Then, as $t \rightarrow +\infty$ we have

$$|s(t+1) - s(t)| \rightarrow 0 \iff s(t+1) = g(s(t)) \rightarrow s(t),$$

which indicates that the FOCUSS sequence $\{s(t)\}_{t=0}^{\infty}$ obtained by (3) goes to one of the stationary points or fixed points of (3) as $t \rightarrow +\infty$.

In addition, by Lemma [3] there are only a finite number of fixed points for (3).

For the same reason as in [39], $\{s(t)\}_{t=0}^{\infty}$ obtained by (3) must converge to one of stationary points. Therefore, $\{s(t)\}_{t=0}^{\infty}$ is convergent.

**V. The Relation Between the FOCUSS Method and the Newton Method for $\ell_p$ Optimization Problem**

**Theorem 5.** The FOCUSS formula (3) is a quasi-Newton (but not exact Newton) algorithm for minimizing the Lagrange function $L(s, \alpha)$ in (4) with the following quasi-Hessian matrix

$$H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} = \begin{bmatrix} 0 & A^T \cdot \Pi \cdot A \end{bmatrix}.$$
\[ H^{-1} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}^{-1} = \begin{bmatrix} I & 0 \\ -H_{22} H_{21} & I \end{bmatrix} \left( \begin{bmatrix} H_{11} - H_{12} H_{22} H_{21} & 0 \\ 0 & H_{22} \end{bmatrix}^{-1} \right) \begin{bmatrix} I & -H_{12} H_{22} \\ 0 & I \end{bmatrix} \]
\[ \tilde{H}^{-1} = \frac{-p(p-1) \cdot (AP^{-1} A^T)^{-1}}{\Pi^{-1} A^T (AP^{-1} A^T)^{-1}} \left( \frac{1}{p} \Pi^{-1} A^T (AP^{-1} A^T)^{-1} \Pi^{-1} + \frac{1}{p} \Pi^{-1} \right) \cdot \frac{(AP^{-1} A^T)^{-1} AP^{-1}}{p(p-1) \Pi^{-1}} . \]

Proof: From [5], we have
\[ \frac{\partial L}{\partial \alpha} = \begin{bmatrix} As - x \\ p \cdot \Pi(s) \cdot s + A^T \alpha \end{bmatrix} = H \cdot \begin{bmatrix} \alpha \\ s \end{bmatrix} - \begin{bmatrix} x \\ 0 \end{bmatrix} . \]

From [40], we can derive (26). By the quasi-Newton iterative formula, we have
\[ \begin{bmatrix} \alpha^{(t+1)} \\ s^{(t+1)} \end{bmatrix} = \begin{bmatrix} \alpha^{(t)} \\ s^{(t)} \end{bmatrix} - H^{-1}(\alpha^{(t)}, s^{(t)}) \cdot \begin{bmatrix} \frac{\partial L(\alpha^{(t)})}{\partial \alpha^{(t)}} \\ \frac{\partial L(s^{(t)})}{\partial s^{(t)}} \end{bmatrix} \]
\[ = H^{-1}(\alpha^{(t)}, s^{(t)}) \cdot \begin{bmatrix} x \\ 0 \end{bmatrix} \]
\[ = \left[ -p \cdot (A \cdot \Pi^{-1}(s^{(t)}) \cdot A^T)^{-1} \cdot x \\ \Pi^{-1}(s^{(t)}) \cdot A^T \cdot [A \cdot \Pi^{-1}(s^{(t)}) \cdot A^T]^{-1} \cdot x \right] , \]

thus, we have
\[ s^{(t+1)} = \Pi^{-1}(s^{(t)}) \cdot A^T \cdot [A \cdot \Pi^{-1}(s^{(t)}) \cdot A^T]^{-1} \cdot x . \]

Hence, the FOCUSS algorithm is a quasi-Newton method. The proof is completed.

However, the FOCUSS algorithm is NOT an exact Newton method because \( \tilde{H} \) is just a quasi-Hessian rather than exactly the Hessian matrix except \( p = 2 \), noticing that \( \tilde{H} = \tilde{H} \) only at \( p = 2 \), where \( \tilde{H} \) is the exact Hessian matrix given by
\[ \tilde{H} = \left[ \begin{array}{cc} \frac{\partial^2 L}{\partial \alpha^2} & \frac{\partial^2 L}{\partial \alpha \partial s} \\ \frac{\partial^2 L}{\partial \alpha \partial s} & \frac{\partial^2 L}{\partial s^2} \end{array} \right] = \begin{bmatrix} 0 & A^T \\ A \cdot (p-1) \cdot \Pi \end{bmatrix} , \]

which is different from \( H \). In the same manner as in (26), we obtain \( \tilde{H}^{-1} \) in (27). Then, we have the Newton method for Lagrange function \( L(s, \alpha) \) as follows:
\[ \begin{bmatrix} \alpha^{(t+1)} \\ s^{(t+1)} \end{bmatrix} = \begin{bmatrix} \alpha^{(t)} \\ s^{(t)} \end{bmatrix} - \tilde{H}^{-1}(\alpha^{(t)}, s^{(t)}) \cdot \begin{bmatrix} \frac{\partial L(\alpha^{(t)})}{\partial \alpha^{(t)}} \\ \frac{\partial L(s^{(t)})}{\partial s^{(t)}} \end{bmatrix} \]
\[ = (I - \tilde{H}^{-1} H) \cdot \begin{bmatrix} \alpha^{(t)} \\ s^{(t)} \end{bmatrix} + \tilde{H}^{-1}(\alpha^{(t)}, s^{(t)}) \cdot \begin{bmatrix} x \\ 0 \end{bmatrix} \]
\[ = \left[ -p \cdot (A \cdot \Pi^{-1}(s^{(t)}) \cdot A^T)^{-1} \cdot x \\ \frac{1}{p} \Pi^{-1}(s^{(t)}) A^T (A \cdot \Pi^{-1}(s^{(t)}) A^T)^{-1} \cdot x + (1 - \frac{1}{p}) s^{(t)} \right] , \]
i.e., the Newton iteration for minimizing Lagrange function \( L(s, \alpha) \) in (4) is
\[ s^{(t+1)} = \frac{1}{p-1} \Pi^{-1}(s^{(t)}) A^T (A \cdot \Pi^{-1}(s^{(t)}) A^T)^{-1} x + \left( 1 - \frac{1}{p} \right) s^{(t)} , \]

which differs from FOCUSS (3). Unfortunately, the numerical experiments show that the Newton method does not work well. This probably might be due to the non-positive definiteness of Hessian matrix \( \tilde{H} \).

VI. Convergence Rate of FOCUSS Algorithm

One of the key measures of the performance of an iterative algorithm is its rate of convergence [41], [42]. We discuss the convergence rate of FOCUSS algorithm in this section, which is simply shown in Table I.

Suppose that the sequence \( \{s^{(t)}\}_{t=0}^{+\infty} \) converges to \( s^(*) \). We say that the convergence is linear if there exists a constant \( \mu \in (0, 1) \) such that
\[ \frac{\|s^{(t+1)} - s^(*)\|}{\|s^{(t)} - s^(*)\|} \leq \mu \text{ as } t \to +\infty. \]

The sequence \( \{s^{(t)}\}_{t=0}^{+\infty} \) is said to converge superlinearly if
\[ \lim_{t \to +\infty} \frac{\|s^{(t+1)} - s^(*)\|}{\|s^{(t)} - s^(*)\|} = 0. \]

One says that it converges sublinearly if it converges, but
\[ \lim_{t \to +\infty} \frac{\|s^{(t+1)} - s^(*)\|}{\|s^{(t)} - s^(*)\|} = 1. \]

More generally, we say that its order of convergence is \( r \) (\( r > 1 \)) if
\[ \frac{\|s^{(t+1)} - s^(*)\|}{\|s^{(t)} - s^(*)\|^r} \leq \mu \text{ as } t \to +\infty, \]

where \( \mu > 0 \) is not necessarily less than 1.

It is well known that many quasi-Newton methods converge superlinearly, whereas a Newton method converges quadratically. In contrast, the steepest descent algorithms converge only at linear rate [42]. In general, the speed of convergence depends on \( r \) and (more weakly) on \( \mu \), e.g., a quadratically convergent sequence will always eventually converge faster than a linearly one [42].
A. The superlinear convergence of FOCUSS for $0 < p < 1$

**Lemma 4.** For $0 < p < 1$, denote

$$h_j(s^{(\ast)}) \triangleq |s_j^{(\ast)}|^{1-p} \cdot \text{sign}(s_j^{(\ast)}) \cdot a_j^T \cdot (A \cdot \Pi^{-1}(s^{(\ast)}) \cdot A^T)^{-1} \cdot x,$$

where $\text{sign}(\cdot)$ is a sign function. Suppose the FOCUSS algorithm (3) converges to $s^{(\ast)}$, then $h(s^{(\ast)}) = [h_1(s^{(\ast)}), \ldots, h_n(s^{(\ast)})]^T$ is a $(0,1)$-vector given by

$$h_j(s^{(\ast)}) = \begin{cases} 1, & s_j^{(\ast)} \neq 0 \\ 0, & s_j^{(\ast)} = 0 \end{cases}$$

where $j = 1, \ldots, n$.

**Proof:** First of all, for $0 < p < 1$, from (31), if $s_j^{(\ast)} = 0$ we can immediately derive $h_j(s^{(\ast)}) = 0$.

In addition, since $s^{(\ast)}$ is a stationary point of FOCUSS algorithm, i.e.,

$$s^{(\ast)} = \Pi^{-1} \cdot A^T \cdot (A \cdot \Pi^{-1}(s^{(\ast)}) \cdot A^T)^{-1} \cdot x,$$

which can be equivalently rewritten as follows:

$$\text{diag}(s^{(\ast)}) \cdot 1 = \text{diag}(s^{(\ast)})$$

$$\times \text{diag} \left[ |s_1^{(\ast)}|^{1-p} \cdot \text{sign}(s_1^{(\ast)}), \ldots, |s_n^{(\ast)}|^{1-p} \cdot \text{sign}(s_n^{(\ast)}) \right]$$

$$\times A^T \cdot (A \cdot \Pi^{-1}(s^{(\ast)}) \cdot A^T)^{-1} \cdot x$$

$$= \text{diag}(s^{(\ast)}) \cdot \left[ h_1(s^{(\ast)}), \ldots, h_n(s^{(\ast)}) \right]^T \implies h_j(s^{(\ast)}) = 1 \text{ if } s_j^{(\ast)} \neq 0.$$

In summary, we have

$$h_j(s^{(\ast)}) = \begin{cases} 1, & s_j^{(\ast)} \neq 0 \\ 0, & s_j^{(\ast)} = 0 \end{cases}$$

where $j = 1, \ldots, n$.

**Lemma 5.** Denote

$$G(s^{(\ast)}) = \Pi^{-1}(s^{(\ast)}) \cdot A^T \cdot (A \cdot \Pi^{-1}(s^{(\ast)}) \cdot A^T)^{-1} \cdot \Pi.$$ (32)

Suppose the FOCUSS algorithm (3) converges to $s^{(\ast)}$, if $0 < p < 1$, then

$$Q(s^{(\ast)}) = G(s^{(\ast)}) \cdot \text{diag}[h(s^{(\ast)})]$$

$$= G(s^{(\ast)}) \cdot \begin{bmatrix} h_1(s^{(\ast)}) & 0 & \cdots & 0 \\ 0 & h_2(s^{(\ast)}) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_n(s^{(\ast)}) \end{bmatrix}$$

is a diagonal $(0,1)$-matrix such that

$$Q_{ij}(s^{(\ast)}) = \begin{cases} 1, & i = j \text{ and } s_i^{(\ast)} \neq 0 \\ 0, & \text{otherwise} \end{cases}$$

**Proof:** For $0 < p < 1$, from Lemma 2 we have $\#\{s^{(\ast)}\} \leq m$. Without loss of generality, suppose the nonzero components of $s$ are $s_{n_1}, \ldots, s_{n_m}$, where $\Omega_N = \{n_1, \ldots, n_m\} \subset \{1, \ldots, n\}$. Then, we can get

$$[\Pi^{-1}(s^{(\ast)}) \cdot A^T]_{i,:} = \begin{cases} |s_i^{(\ast)}|^{2-p} \cdot a_i^T, & s_i^{(\ast)} \neq 0 \text{ or } i \in \Omega_N \\ 0^T, & \text{otherwise} \end{cases}$$

where $[\cdot]_{i,:}$ stands for the $i$th row of a matrix. From Lemma 4 we have

$$[A \cdot \text{diag}[h(s^{(\ast)})]]_{ :, j} = \begin{cases} a_j, & s_j^{(\ast)} \neq 0 \text{ or } j \in \Omega_N \\ 0, & \text{otherwise} \end{cases}$$

where $[\cdot]_{ :, j}$ denotes the $j$th column of a matrix. In addition,

$$[A \cdot \Pi^{-1}(s^{(\ast)}) \cdot A^T]^{-1} = [A_N \cdot \Pi^{-1}(s^{(\ast)}_N) \cdot A_N^T]^{-1}$$

$$(\Pi(s^{(\ast)}_N))^{-1} \cdot (A(s^{(\ast)}_N) \cdot A_N^{-1},$$

where $\Pi(s^{(\ast)}_N)$ is a positive-definite diagonal matrix given by

$$\Pi(s^{(\ast)}_N) = \begin{bmatrix} |s_{n_1}^{(\ast)}|^{p-2} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & |s_{n_m}^{(\ast)}|^{p-2} \end{bmatrix}.$$ 

Consequently, from (32), we can derive

$$Q_{ij}(s^{(\ast)}) = \begin{cases} h_i(s^{(\ast)}) = 1, & i = j \text{ and } s_i^{(\ast)} \neq 0 \\ 0, & \text{otherwise} \end{cases}$$

For simplicity, let us consider the simplest example for Lemma 2 $\Omega_N = \{n_1, \ldots, n_m\} = \{1, \ldots, m\}$, for which we can readily verify Lemma 5.

**Theorem 6.** Suppose $A$, $x$ and $s^{(0)}$ satisfy Assumption 2, Assumption 2 and the FOCUSS algorithm (3) converges to $s^{(\ast)}$. If $0 < p < 1$, its convergence rate is superlinear, i.e.,

$$\lim_{t\to\infty} \frac{\|s^{(t+1)} - s^{(\ast)}\|}{\|s^{(t)} - s^{(\ast)}\|} = 0.$$

**Proof:** From (40), we have

$$\frac{dM(s)}{ds} = -M(s) \cdot \frac{dM(s)}{ds} \cdot M^{-1}(s).$$

Letting $M^{-1}(s) = A \cdot \Pi^{-1}(s) \cdot A^T$, we have

$$\frac{dM^{-1}(s)}{ds} = \frac{d\Pi^{-1}(s)}{ds} \cdot a_i^T \cdot [A \cdot \Pi^{-1}(s) \cdot A^T]^{-1} \cdot x +$$

$$\Pi^{-1}(s) \cdot a_i^T \cdot \frac{dA_i}{ds} \cdot [A \cdot \Pi^{-1}(s) \cdot A^T]^{-1} \cdot x$$

$$= (2-p)\delta_{ij} \cdot h_j(s) +$$

$$\Pi^{-1}(s) \cdot a_i^T \cdot \frac{dA_i}{ds} \cdot [A \cdot \Pi^{-1}(s) \cdot A^T]^{-1} \cdot x$$

$$= (2-p)\delta_{ij} \cdot h_j(s) - \Pi^{-1}(s) \cdot a_i^T \cdot [A \cdot \Pi^{-1}(s) \cdot A^T]^{-1}$$

$$\times \frac{dA \cdot \Pi^{-1}(s) \cdot A^T}{ds} \cdot [A \cdot \Pi^{-1}(s) \cdot A^T]^{-1} \cdot x$$

$$= (2-p)\left[ \delta_{ij} \cdot h_j(s) - \Pi^{-1}(s) \cdot a_i^T \cdot [A \cdot \Pi^{-1}(s) \cdot A^T]^{-1} \cdot [0, \ldots, h_j(s), \ldots, 0]^T \right],$$

where the indicator function is given by

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$
Then we can derive
\[
\frac{\partial g(s)}{\partial s} = \left[ \frac{dg_j(s)}{ds} \right]_{n \times n}
\]
\[(2 - p) \times [\text{diag}(h_1(s), \ldots, h_n(s)) - G(s) \cdot \text{diag}(h_1(s), \ldots, h_n(s))] \]
\[= (2 - p) (\text{diag}[h(s)] - G(s) \cdot \text{diag}[h(s)]) \]
\[= (2 - p) (\text{diag}[h(s)] - Q(s)). \]

From Lemma 4 and Lemma 5, we can obtain
\[
\frac{\partial g(s^{(\ast)})}{\partial s} = 0_{n \times n}.
\]

By the mean-value theorem, we have
\[
s^{(t+1)} - s^{(\ast)} = g(s^{(t)}) - g(s^{(\ast)}) \\
\approx \frac{\partial g(s^{(\ast)})}{\partial s} \cdot [s^{(t)} - s^{(\ast)}] \\
= 0 \cdot [s^{(t)} - s^{(\ast)}] \]
\[(34)\]
as \(t \to +\infty\). Hence,
\[\lim_{t \to +\infty} \frac{\|s^{(t+1)} - s^{(\ast)}\|}{\|s^{(t)} - s^{(\ast)}\|} = 0.\]

To illustrate the Theorem 6 we give a numerical example in Fig.2 in which the convergence rate is illustrated by \(R^{(t)}\) defined as follows:
\[R^{(t)} = \frac{\|s^{(t+1)} - s^{(\ast)}\|}{\|s^{(t)} - s^{(\ast)}\|}.\]

Note that \(\|s^{(t)} - s^{(\ast)}\| \to 0\) as \(t \to +\infty\). Due to the limited machine accuracy of a computer, it is a difficult task to compute \(R^{(t)}\) in numerical analysis when \(\|s^{(t)} - s^{(\ast)}\| \to 0\). For this reason, we can see in Fig.2 that \(R^{(t)} \to 0\) first as expected in the Theorem 6 but then oscillates when the denominator is plotted. Both \(A \in \mathbb{R}^{125 \times 200}\) and \(x \in \mathbb{R}^{125 \times 1}\) are randomly generated in MATLAB 2010b. For \(p = 0.6, p = 0.7, p = 0.8\) and \(p = 0.95\), given the entrywise-nonzero initializations (i.e., \(|s^{(0)}| > 0\)) generated randomly, the FOCUSS algorithm consistently converges to the sparse solutions \(s^{(\ast)}\) superlinearly, which exactly satisfied \(\#\{s^{(\ast)}\} = m = 125\) and \(\lim_{t \to +\infty} \frac{\|s^{(t+1)} - s^{(\ast)}\|}{\|s^{(t)} - s^{(\ast)}\|} = 0.\)

B. Convergence rate analysis on FOCUSS for \(p = 1\)

**Theorem 7.** Suppose \(A, x\) and \(s^{(0)}\) satisfy Assumption 4 Assumption 3 and the FOCUSS algorithm converges to \(s^{(\ast)}\). For \(p = 1\), its convergence rate is first-order at most, i.e.,
\[\lim_{t \to +\infty} \frac{\|s^{(t+1)} - s^{(\ast)}\|}{\|s^{(t)} - s^{(\ast)}\|} = \lim_{t \to +\infty} \frac{\|I - G(s^{(\ast)}) \cdot \text{diag}[h(s^{(\ast))}] \cdot [s^{(t)} - s^{(\ast)}]\|}{\|s^{(t)} - s^{(\ast)}\|} \leq 1,
\[
\text{where } h(s^{(\ast)}) = \text{diag}[\text{sign}(s^{(\ast)})] \cdot A^T [A \cdot \Pi^{-1}(s^{(\ast)}) \cdot A^T]^{-1} \cdot x.
\]

And
\[\text{diag}[\text{sign}(s^{(\ast)})] = \begin{bmatrix}
\text{sign}(s_1^{(\ast)}) & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & \text{sign}(s_n^{(\ast)})
\end{bmatrix}.\]
Proof: In the Theorem 4, we have proved that for \( p = 1 \) the FOCUSS algorithm is convergent, given a strictly nonzero initialization \( s^{(0)} \), i.e., the sequence \( \{s^{(t)}\}_{t=0}^{\infty} \) obtained by FOCUSS with \( p = 1 \) is convergent. Hence, we have
\[
\lim_{t \to +\infty} \frac{|s^{(t+1)} - s^{(t)}|}{|s^{(t)} - s^{(t)}|} \leq 1,
\]
or else, \( \{s^{(t)}\}_{t=0}^{\infty} \) would be divergent. Accordingly, from (33) and (34), we have
\[
\begin{align*}
&\lim_{t \to +\infty} \|s^{(t+1)} - s^{(t)}\| = \frac{\|\frac{\partial g}{\partial s}(s^{(t)}) \cdot [s^{(t)} - s^{(t)}]\|}{\|s^{(t)} - s^{(t)}\|} \\
&= \lim_{t \to +\infty} \frac{\left\| \left[ I - G(s^{(t)}) \right] \cdot [s^{(t)} - s^{(t)}] \right\|}{\|s^{(t)} - s^{(t)}\|} \leq 1,
\end{align*}
\]
where \( h(s^{(t)}) \) is given by (35).

Remark 1. As mentioned in [26], for \( p = 1 \), the optimal solution of problem (2) satisfies \( \#s^{(t)} = m \) if \( A \) and \( x \) satisfy Assumption 2. However, due to the numerical inaccuracy of a computer, for \( p = 1 \), the FOCUSS algorithm usually derives an approximately optimal solution instead satisfying \( \#s^{(t)} > m \) but \( \#s^{(t)} \geq \epsilon > 0 \) is a very small positive number (e.g., \( \epsilon = 10^{-30} \)). For example, suppose the optimal solution is \( s^* = [e_1, e_2, 0, 0, 0]^T \); due to the numerical inaccuracy, the FOCUS algorithm with \( p = 1 \) can just converges to \( s_i = [e_1, e_2, e_3, e_4, e_5]^T \approx s^* \) (but \( s_i \neq s^* \)) so that \( \text{sign}(s_i) \neq \text{sign}(s^*) \). Thus, \( R^{(t)} \to R^* \neq R^* = 0 \) as \( t \to +\infty \).

Extensive experiments show that \( 0 < R^* < 1 \) for FOCUSS algorithm with \( p = 1 \) (see Fig. 2).

C. Convergence analysis on FOCUSS for \( 1 < p < 2 \)

Lemma 6. Suppose the FOCUSS algorithm (3) converges to \( s^{(t)} = [s_1^{(t)}, \ldots, s_n^{(t)}]^T \), i.e., \( \lim_{t \to +\infty} s^{(t)} = s^{(t)} \). For \( 1 < p < 2 \), if \( s_j^{(t)} = 0 \), we have
\[
\lim_{t \to +\infty} h_j(s^{(t)}) = 0,
\]
where \( h_j(s^{(t)}) \) is defined in (31), \( j = 1, \ldots, n \).

Proof: Note that the FOCUSS iterative sequence \( \{s^{(t)}\}_{t=0}^{\infty} \) is convergent, where \( s^{(t)} = [s_1^{(t)}, \ldots, s_n^{(t)}]^T \). Thus, \( s_j^{(t)} \) is also convergent. Suppose that it converge to \( s_j^{(t)} \). Then from (18) and (19), we have
\[
p \cdot |s_j^{(t)}|^{p-1} \cdot \text{sign}(s_j^{(t)}) + a_j^T \alpha^* = a_j^T \alpha^* = 0.
\]
So, if \( s_j^{(t)} = 0 \), we have
\[
a_j^T \alpha^* = a_j^T \cdot [A \cdot \Pi^{-1}(s^*) \cdot A^T]^{-1} \cdot x = 0.
\]
In addition, for \( 1 < p < 2 \), \( \lim_{t \to +\infty} s_j^{(t)} = s_j^{(t)} = 0 \Rightarrow \lim_{t \to +\infty} |s_j^{(t)}|^{1-p} = +\infty \). Hence,
\[
\begin{align*}
h_j(s^{(t)}) &= |s_j^{(t)}|^{1-p} \cdot \text{sign}(s_j^{(t)}) \cdot a_j^T [A \cdot \Pi^{-1}(s^*) \cdot A^T]^{-1} \cdot x \\
&= \frac{\text{sign}(s_j^{(t)}) \cdot a_j^T [A \cdot \Pi^{-1}(s^*) \cdot A^T]^{-1} \cdot x}{|s_j^{(t)}|^{1-p}} \\
&\to 0
\end{align*}
\]
as \( t \to +\infty \), i.e., \( h_j(s^{(t)}) \) is an indeterminate form with respect to \( s_j^{(t)} \) as \( t \to +\infty \). Following the proof of Theorem 5,
by L'Hôpital's rule, we can derive

\[ |h_j(s^{(*)})| = \lim_{t \to +\infty} \frac{|\text{sign}(s_j^{(t)}) \cdot a_j^T [A \cdot \Pi^{-1}(s^{(t)}) \cdot A^T]^{-1} x|}{|s_j^{(t)}|^{p-1}} \]

\[ \leq \lim_{t \to +\infty} \frac{|a_j^T [A \cdot \Pi^{-1}(s^{(t)}) \cdot A^T]^{-1} x|}{|s_j^{(t)}|^{p-1}} \]

\[ = \lim_{t \to +\infty} \left[ |a_j^T [A \cdot \Pi^{-1}(s^{(t)}) \cdot A^T]^{-1} x| \right]_{s_j^{(t)}}' \]

\[ = \frac{2-p}{p-1} \lim_{t \to +\infty} \frac{|s_j^{(t)}|^{p-1} \cdot a_j^T [A \cdot \Pi^{-1}(s^{(t)}) \cdot A^T]^{-1} a_j h_j(s^{(t)})}{|s_j^{(t)}|^{p-2}} \]

\[ = \frac{2-p}{p-1} \lim_{t \to +\infty} |s_j^{(t)}| \cdot |a_j^T [A \cdot \Pi^{-1}(s^{(t)}) \cdot A^T]^{-1} a_j| \cdot |h_j(s^{(t)})| \]

\[ = 0 \cdot |h_j(s^{(*)})|, \]

i.e., \( |h_j(s^{(*)})| \leq 0 \cdot |h_j(s^{(*)})| \), which indicates that \( h_j(s^{(*)}) = 0 \) or \( h_j(s^{(*)}) = +\infty \). As \( h_j(s^{(*)}) \) is bounded, we have \( \lim_{t \to +\infty} h_j(s^{(t)}) = 0 \).

**Theorem 8.** Suppose \( A, x \) and \( s^{(0)} \) satisfy Assumption [1] [Assumption 2] and the FOCUSS algorithm [3] converges to \( s^{(*)} \). If \( 1 < p < 2 \) and \( \# \{s^{(*)}\} = m \), it converges superlinearly, i.e.,

\[ \lim_{t \to +\infty} \frac{\|s^{(t+1)} - s^{(*)}\|}{\|s^{(t)} - s^{(*)}\|} = 0. \]

The Theorem [8] is simply demonstrated in Fig [4].

**Proof:** First, for \( 1 < p < 2 \) and \( \# \{s^{(*)}\} = m \), from Lemma [6] we have \( \lim_{t \to +\infty} h_j(s^{(t)}) = 0 \) if \( s_j^{(*)} = 0 \). In addition, in a similar way parallel to the proof Lemma [4] we can derive

\[ \lim_{t \to +\infty} h_j(s^{(t)}) = 1 \text{ if } s_j^{(*)} \neq 0. \]

We have

\[ \lim_{t \to +\infty} h_j(s^{(t)}) = \begin{cases} 1, & s_j^{(*)} \neq 0 \\ 0, & s_j^{(*)} = 0 \end{cases} \]

for \( 1 < p < 2 \) and \( j = 1, \ldots, n \). Then, in the same way as the proof of Theorem [6] we can derive \( \frac{\partial g(s^{(*)})}{\partial s} = 0_{n \times n} \) and

\[ s^{(t+1)} - s^{(*)} = g(s^{(t)}) - g(s^{(*)}) \approx \left[ \frac{\partial g(s^{(*)})}{\partial s} \right]^T, [s^{(t)} - s^{(*)}] \]

\[ = 0 \cdot [s^{(t)} - s^{(*)}] \]

as \( t \to +\infty \). Thus, for \( 1 < p < 2 \), we have

\[ \lim_{t \to +\infty} \frac{\|s^{(t+1)} - s^{(*)}\|}{\|s^{(t)} - s^{(*)}\|} = 0 \text{ if } \# \{s^{(*)}\} = m. \]

**Theorem 9.** Suppose \( A, x \) and \( s^{(0)} \) satisfy Assumption [1] [Assumption 2] and the FOCUSS algorithm [3] converges to \( s^{(*)} \). For \( 1 < p < 2 \), if \( \# \{s^{(*)}\} = n \), it converges linearly and

\[ \lim_{t \to +\infty} \frac{\|s^{(t+1)} - s^{(*)}\|}{\|s^{(t)} - s^{(*)}\|} = 2 - p < 1. \]

The Theorem [9] is demonstrated by an illustrative example in Fig [5].

**Proof:** For \( 1 < p < 2 \) and \( \# \{s^{(*)}\} = n \), the stationary point \( s^{(*)} \) is nonzero entrywisely, i.e., \( s_i^{(*)} \neq 0 \) for \( i = 1, \ldots, n \). Note that

\[ s^{(*)} = \Pi^{-1} \cdot A^T \cdot [A \cdot \Pi^{-1}(s^{(*)}) \cdot A^T]^{-1} \cdot x \]

can be rewritten as follows:

\[ \text{diag}(s^{(*)}) = \text{diag}(s^{(*)}) \]

\[ \times \text{diag} \left[ |s_1^{(*)}|^{(1-p)}, \ldots, |s_n^{(*)}|^{(1-p)} \cdot \text{sign}(s_1^{(*)}) \right] \]

\[ \times A^T \cdot [A \cdot \Pi^{-1}(s^{(*)}) \cdot A^T]^{-1} \cdot x \]

\[ = \text{diag}(s^{(*)}) \cdot \left[ h_1(s^{(*)}), \ldots, h_n(s^{(*)}) \right]^T = \text{diag}(s^{(*)}) \cdot h(s^{(*)}) \]
Fig. 5. An example demonstrating the Theorem 9, in which it is shown that \( \lim_{t \to +\infty} R(t) = 2 - p \) for \( 1 < p < 2 \) and \( \#(s^{(t)}) = n \). In this example, the datasets are the same as in Fig. For \( p = 1.1, p = 1.3, p = 1.5, p = 1.7, p = 1.9 \) and \( p = 1.95 \), given the entrywise-nonzero initializations \( s^{(0)} \) generated randomly, the FOCUSS algorithm uniformly converges with linear-convergence rate such that \( \#(s^{(t)}) = n = 200 \) and \( \lim_{t \to +\infty} \|s^{(t)} - s^{(0)}\| = 2 - p \).

Fig. 6. An example illustrating the Theorem 10 which shows that \( \lim_{t \to +\infty} R(t) = 2 - p \) for \( 1 < p < 2 \) and \( m < \#(s^{(t)}) < n \). The datasets, containing \( A \in \mathbb{R}^{13 \times 20} \) and \( x \in \mathbb{R}^{13 \times 1} \), were randomly generated by the Algorithm 2 in Appendix B.

\[
\Rightarrow \quad h(s^{(t)}) = 1_{n \times 1} \\
\Rightarrow \quad \text{diag}[h(s^{(t)})] = I_{n \times n}. \\
\]

(36)

On the other side, from Theorem 1, given a strictly nonzero initialization \( s^{(0)} \), we have \( As^{(t)} = x \) under the FOCUSS 4 for \( t = 1, \ldots, +\infty \). Then, from (32) and (36), we have

\[
G(s^{(t)}) \cdot \text{diag}[h(s^{(t)})] \cdot [s^{(t)} - s^{(t)}] \\
= G(s^{(t)}) \cdot [s^{(t)} - s^{(t)}] \\
= \Pi^{-1}(s^{(t)}) \cdot A^T \cdot [A \cdot \Pi^{-1}(s^{(t)}) \cdot A^T \cdot A \cdot [s^{(t)} - s^{(t)}]] \\
= \Pi^{-1}(s^{(t)}) \cdot A^T \cdot [A \cdot \Pi^{-1}(s^{(t)}) \cdot A^T] \cdot [x - x] \\
= 0.
\]

Then, from (33), (34) and (36), we can obtain

\[
e(t + 1) = s^{(t+1)} - s^{(t)} = g(s^{(t)}) - g(s^{(t)}) \\
\frac{\partial g(s^{(t)})}{\partial s} \cdot [s^{(t)} - s^{(t)}] \\
= (2 - p) \left( \text{diag}[h(s^{(t)})] - G(s^{(t)}) \cdot \text{diag}[h(s^{(t)})] \right) [s^{(t)} - s^{(t)}] \\
= (2 - p) \cdot \text{diag}[h(s^{(t)})] \cdot [s^{(t)} - s^{(t)}] \\
= (2 - p) \cdot e(t) \\
\Rightarrow \lim_{t \to +\infty} \frac{\|e^{(t+1)}\|}{\|e^{(t)}\|} = \lim_{t \to +\infty} \frac{\|s^{(t+1)} - s^{(t)}\|}{\|s^{(t)} - s^{(t)}\|} = 2 - p < 1
\]

for \( 1 < p < 2 \).

Then, from (33), (34) and (36), we can obtain

\[
e(t + 1) = s^{(t+1)} - s^{(t)} = g(s^{(t)}) - g(s^{(t)}) \\
\frac{\partial g(s^{(t)})}{\partial s} \cdot [s^{(t)} - s^{(t)}] \\
= (2 - p) \left( \text{diag}[h(s^{(t)})] - G(s^{(t)}) \cdot \text{diag}[h(s^{(t)})] \right) [s^{(t)} - s^{(t)}] \\
= (2 - p) \cdot \text{diag}[h(s^{(t)})] \cdot [s^{(t)} - s^{(t)}] \\
= (2 - p) \cdot e(t) \\
\Rightarrow \lim_{t \to +\infty} \frac{\|e^{(t+1)}\|}{\|e^{(t)}\|} = \lim_{t \to +\infty} \frac{\|s^{(t+1)} - s^{(t)}\|}{\|s^{(t)} - s^{(t)}\|} = 2 - p < 1
\]

for \( 1 < p < 2 \).

Then, from (33), (34) and (36), we can obtain

\[
e(t + 1) = s^{(t+1)} - s^{(t)} = g(s^{(t)}) - g(s^{(t)}) \\
\frac{\partial g(s^{(t)})}{\partial s} \cdot [s^{(t)} - s^{(t)}] \\
= (2 - p) \left( \text{diag}[h(s^{(t)})] - G(s^{(t)}) \cdot \text{diag}[h(s^{(t)})] \right) [s^{(t)} - s^{(t)}] \\
= (2 - p) \cdot \text{diag}[h(s^{(t)})] \cdot [s^{(t)} - s^{(t)}] \\
= (2 - p) \cdot e(t) \\
\Rightarrow \lim_{t \to +\infty} \frac{\|e^{(t+1)}\|}{\|e^{(t)}\|} = \lim_{t \to +\infty} \frac{\|s^{(t+1)} - s^{(t)}\|}{\|s^{(t)} - s^{(t)}\|} = 2 - p < 1
\]

for \( 1 < p < 2 \).
converges at linear rate and
\[
\lim_{t \to +\infty} \frac{\|s(t+1) - s^(*)\|}{\|s(t) - s^(*)\|} = 2 - p.
\]

The Theorem 10 is demonstrated by a numerical example in Fig.6.

**Proof:** As described previously, \(s_N^(*)\) and \(s_O^(*)\) denote the nonzero and zero components of \(s^(*)\), respectively. Thus, we have \(m < \#\{s^(*)\} \leq m\) and \(k < n\). Noting that \(\lim_{t \to +\infty} s_N^t = s_O^t = 0_{(n-k)\times 1}\) and following the proof of Theorem 8 and Theorem 9, we can analogously obtain the infinitesimal expressions

\[
e_O^{(t+1)} = s_O^{(t+1)} - s_O^t = g_O(s^t) - g_O(s^*) = \left[\frac{\partial g_O(s^*)}{\partial s_N}ight]^T \cdot [s_N^t - s_N^*] = 0_{(n-k)\times k} \cdot e_N^t = o[e_N^{(t)}],
\]

i.e.,

\[
e_O^{(t+1)} = o[e_N^{(t)}].
\]

Moreover, following the proof of Theorem 9, we can analogously obtain

\[
e_N^{(t+1)} = (2 - p) \cdot e_N^t + o[e_N^{(t)}],
\]

or

\[
e_N^{(t)} = \frac{1}{2 - p} \cdot e_N^{(t+1)} + o[e_N^{(t)}].
\]

From (37) and (39), we can derive

\[
e_O^{(t+1)} = o[e_N^{(t)}] = o\left[\frac{1}{2 - p} \cdot e_N^{(t+1)} + o[e_N^{(t)}]\right] = o[e_N^{(t+1)}]
\]

\[
\implies e_O^{(t)} = o[e_N^{(t)}].
\]

as \(t \to +\infty\). Combing (37), (38) and (40), we have

\[
\lim_{t \to +\infty} \frac{\|e_N^{(t+1)}\|}{\|e_N^{(t)}\|} = \lim_{t \to +\infty} \frac{\|e_N^{(t+1)}\|}{\|e_N^{(t)}\|}
\]

\[
= \lim_{t \to +\infty} \left\|\left[\begin{array}{c} e_N^{(t+1)} \\ e_O^{(t+1)} \end{array}\right]\right\| = \lim_{t \to +\infty} \left\|\left[\begin{array}{c} e_N^{(t)} \\ e_O^{(t)} \end{array}\right]\right\|
\]

\[
= \lim_{t \to +\infty} \left\|\left[\begin{array}{c} e_N^{(t)} \\ e_O^{(t)} \end{array}\right]\right\| = \lim_{t \to +\infty} \left\|\left[\begin{array}{c} (2 - p) \cdot e_N^{(t)} + o[e_N^{(t)}] \\ o[e_N^{(t)}] \end{array}\right]\right\|
\]

\[
= \lim_{t \to +\infty} \left\|\left[\begin{array}{c} (2 - p) \cdot e_N^{(t)} + o[e_N^{(t)}] \\ 0_{(n-k)\times 1} \end{array}\right]\right\| = \lim_{t \to +\infty} \left\|\left[\begin{array}{c} (2 - p) \cdot e_N^{(t)} + o[e_N^{(t)}] \\ 0_{(n-k)\times 1} \end{array}\right]\right\|
\]

\[
= 2 - p.
\]

Therefore, we also have \(\lim_{t \to +\infty} \frac{\|s(t+1) - s^(*)\|}{\|s(t) - s^(*)\|} = 2 - p\) for \(1 < p < 2\) if \(m < \#\{s^(*)\} < n\).

**VII. DISCUSSIONS**

It is worth mentioning that for \(p \leq 0\), the FOCUSS algorithm (3) usually obtain an approximately sparse solution \(s^(*)\) such that \(\#\{s^(*)\} = n\) but the number of significantly nonzero components of \(s^(*)\) is less than \(n\), i.e., \(\#\{s^(*)\} = n\) but \(\#\{i : |s_i^(*)| > \epsilon\} < n\), where \(\epsilon\) is a small positive number (e.g., \(\epsilon = 10^{-2}\)). So in practice, for \(1 < p < 2\), the Theorem 9 occurs much more often than the Theorem 8 and Theorem 10.

Comparing the Theorem 6 and the Theorem 9, we know that the FOCUSS algorithm converges more rapidly for \(1 < p < 2\) than \(1 < p < 2\), while it is relatively easy to get stuck into local minima however if \(p\) is too small (e.g., \(p = 0.1\)) because the optimization problem (2) is not convex for \(1 < p < 1\). Accordingly, it is suggested to select a value, slightly small than 1 but not too small, for \(p\). Typically, we can set \(p = 0.8\) for the larger scale problems.

**VIII. CONCLUSIONS**

The FOCUSS method is one of the most efficient algorithms for sparse representation and compressive sensing, which is easy to implement. In this paper, we provide a thorough convergence analysis on this algorithm towards establishing a systematic convergence theory for it. At first, we propose a rigorous derivation via auxiliary function. Then, we prove its convergence. In particular, we have rigorously analyzed its convergence rate for different sparsity parameter and demonstrated its convergence rate by numerical experiments.

**APPENDIX A**

**AN ALGORITHM GENERATING DATASETS FOR THEOREM 8**

In order to demonstrate the Theorem 8 and Theorem 10, we specially design two algorithms generating some appropriate datasets satisfying the conditions of two theorems, respectively.

Suppose the FOCUSS algorithm (3) converges to \(s^(*)\) such that \(\#\{s^(*)\} = m\). Then \(x\) can be represented as

\[
x = As^(*) = [A_N, A_O] \cdot \begin{bmatrix} s_N^(*) \\ s_O^(*) \end{bmatrix},
\]

where \(s_N^(*) = [s_{01}^(*), \cdots, s_{0m}^(*)]^T\) and \(s_O^(*) = [s_{n1}^(*), \cdots, s_{nm}^(*)]^T\) are the zero and nonzero components of \(s^(*)\), respectively; by analogy, \([A_N, A_O] \cdot [s_N^(*) + 0_{(n-k)\times 1}]^T\) respectively correspond to \(s_N^(*)\) and \(s_O^(*)\) in terms of \(x = As^(*) = A_N \cdot s_N^(*) + A_O \cdot s_O^(*)\).
Theorem 11. For $1 < p < 2$, supposing $A$, $x$ and $s^{(0)}$ satisfy Assumption I and the FOCUS algorithm converges to $\mathbf{s}^{(s)}$, the sufficient and necessary condition satisfying $\#\{s^{(s)}\} = m$ is

$$A_T^T \cdot (A_N^{-1})^T \cdot \begin{bmatrix} |s_1^{(s)}|^{{p-1}} \cdot \text{sign}(s_1^{(s)}) \\ |s_2^{(s)}|^{{p-1}} \cdot \text{sign}(s_2^{(s)}) \\ \vdots \\ |s_{{n-m}}^{(s)}|^{{p-1}} \cdot \text{sign}(s_{{n-m}}^{(s)}) \end{bmatrix} = 0_{(n-m) \times 1},$$

(41)

**Proof:** Note that $\#\{s^{(s)}\} = \#\{s^{(s)}\} = m$ and $s^{(s)} = [s_{{i_1}}^{(s)}; \cdots; s_{{i_{{n-m}}}^{s}}] \in R^{m \times 1}$. For $1 < p < 2$, from (13) and (19), we can derive

$$p \cdot \begin{bmatrix} |s_{{i_1}}^{(s)}|^{{p-1}} \cdot \text{sign}(s_{{i_1}}^{(s)}) \\ |s_{{i_2}}^{(s)}|^{{p-1}} \cdot \text{sign}(s_{{i_2}}^{(s)}) \\ \vdots \\ |s_{{i_{{n-m}}}^{s}}|^{{p-1}} \cdot \text{sign}(s_{{i_{{n-m}}}^{s}}) \end{bmatrix} + A_T^T \cdot \alpha^{(s)} = 0,$$

where

$$\alpha^{(s)} = [A \cdot \Pi^{-1}(\mathbf{s}^{(s)} \cdot A_T)^{-1}] \cdot x = [A \cdot \Pi^{-1}(\mathbf{s}^{(s)} \cdot A_T)^{-1}] \cdot x \neq 0.$$  

Then, $\#\{s^{(s)}\} = m \iff s^{(s)} = [s_{{i_1}}^{(s)}; \cdots; s_{{i_{{n-m}}}^{s}}] \cdot T = 0_{(n-m) \times 1} \iff

$$\begin{bmatrix} |s_{{i_1}}^{(s)}|^{{p-1}} \cdot \text{sign}(s_{{i_1}}^{(s)}) \\ |s_{{i_2}}^{(s)}|^{{p-1}} \cdot \text{sign}(s_{{i_2}}^{(s)}) \\ \vdots \\ |s_{{i_{{n-m}}}^{s}}|^{{p-1}} \cdot \text{sign}(s_{{i_{{n-m}}}^{s}}) \end{bmatrix} = 0 \iff A_T^T \cdot \alpha^{(s)} = 0 \iff

A_T^T \cdot \alpha^{(s)} = A_T^T \cdot [A \cdot \Pi^{-1}(s^{(s)} \cdot A_T)^{-1}] \cdot x

= A_T^T \cdot [A \cdot \Pi^{-1} s^{(s)} \cdot A_T^{-1}] \cdot A_N x

= A_T^T \cdot (A_N^{-1})^T \cdot \Pi(s^{(s)} \cdot A_N^{-1} \cdot A_N x)

= A_T^T \cdot (A_N^{-1})^T \cdot \Pi(s^{(s)} \cdot s_N)

= A_T^T \cdot (A_N^{-1})^T \cdot \begin{bmatrix} |s_{{n_1}}^{(s)}|^{{p-1}} \cdot \text{sign}(s_{{n_1}}^{(s)}) \\ |s_{{n_2}}^{(s)}|^{{p-1}} \cdot \text{sign}(s_{{n_2}}^{(s)}) \\ \vdots \\ |s_{{n_{{m}}}^{s}}|^{{p-1}} \cdot \text{sign}(s_{{n_{{m}}}^{s}}) \end{bmatrix}

= 0_{(n-m) \times 1}.$$  

The proof is completed.

**Remark 2.** The condition (41) in the Theorem 11 can be employed to generate a synthetic dataset for demonstrating the Theorem 8. The detailed procedure is described as Algorithm 7.

**Algorithm 1:** Generating datasets for demonstrating the Theorem 8

**Output:** $x \in R^{m \times 1}$ and $A \in R^{m \times n}$

1. Randomly generate $A = [A_N, A_O] \in R^{m \times n}$, where $A_N \in R^{m \times m}$ and $A_O \in R^{m \times (n-m)}$.
2. Generate a vector $v \in R^{m \times 1}$ by eigenvalue decomposition (EVD) such that

$$A_T^T \cdot (A_N^{-1})^T \cdot v = 0_{(n-m) \times 1},$$

where

$$v = \begin{bmatrix} v_{{n_1}} \\ \vdots \\ v_{{n_m}} \end{bmatrix} = \begin{bmatrix} |s_{{n_1}}^{(s)}|^{{p-1}} \cdot \text{sign}(s_{{n_1}}^{(s)}) \\ \vdots \\ |s_{{n_{{m}}}^{s}}|^{{p-1}} \cdot \text{sign}(s_{{n_{{m}}}^{s}}) \end{bmatrix}.$$  

3. Compute $s_N^{(s)} = [s_{{n_1}}^{(s)}; \cdots; s_{{n_{{m}}}^{s}}]^T$ through (41) as follows:

$$s_N^{(s)} = |v_{{n_1}}|^{{1/p}} \cdot \text{sign}(v_{{n_1}}), \ i = 1, \cdots, m.$$  

4. Compute $x$ by $x = A s^{(s)}$, where $s^{(s)} = [s_N^{(s)}; 0_{(n-k) \times 1}].$

By Algorithm 7, we randomly generated the datasets with the parameters: $m = 15$, $n = 20$ and $\#\{s^{(s)}\} = m = 15$. We demonstrated the Theorem 8 on these datasets in Fig.4.

**Appendix B**

**An Algorithm generating Datasets for the Theorem 11**

Theorem 12. For $1 < p < 2$, supposing $A$, $x$ and $s^{(0)}$ satisfy Assumption I and the FOCUS algorithm converges to $s^{(s)}$ such that $m < \#\{s^{(s)}\} = \#\{s_N^{(s)}\} = k < n$, the sufficient and necessary condition satisfying $s_O^{(s)} = 0_{(n-k) \times 1}$ is

$$A_T^T \cdot \alpha^{(s)} = A_T^T \cdot [A_N \cdot \Pi_N^{-1}(s_N^{(s)} \cdot A_T)^{-1}] \cdot x = 0_{(n-k) \times 1},$$

(42)

where

$$x = A s^{(s)} = [A_N, A_O] \begin{bmatrix} s_N^{(s)} \\ s_O^{(s)} \end{bmatrix}.$$  

$s_N^{(s)} = [s_{{n_1}}^{(s)}; \cdots; s_{{n_{{k}}}^{s}}]^T$ and $s_O^{(s)} = [s_{{n_{{k+1}}}^{s}}; \cdots; s_{{n_{{n}}}^{s}}]^T$ are the zero and nonzero components of $s^{(s)}$, respectively.
Algorithm 2: Generating datasets for demonstrating the Theorem[10]

Output: \( \mathbf{x} \in \mathbb{R}^{m \times 1} \) and \( \mathbf{A} \in \mathbb{R}^{m \times n} \)

1) Randomly generate \( \mathbf{A}_N \in \mathbb{R}^{m \times k} \) and \( \mathbf{x} \in \mathbb{R}^m \), where \( m < k < n \).
2) Given an entrywise nonzero initialization \( s^{(0)}_N \in \mathbb{R}^{k \times 1} \), compute \( s^{(N)}_N \) by the FOCUSS algorithm (3) as follows
\[
s^{(N)}_N = \text{FOCUSS}(\mathbf{x}, \mathbf{A}_N, s^{(0)}_N)
\]
such that \( \mathbf{x} = \mathbf{A}_N \cdot s^{(N)}_N \).
3) Compute \( \mathbf{A}_O \) by EVD such that
\[
\mathbf{A}_O^T \cdot [\mathbf{A}_N \cdot \Pi_N(s^{(s)}_N) \cdot \mathbf{A}_N^T]^{-1} \mathbf{x} = 0.
\]
4) Compute \( \mathbf{A} = [\mathbf{A}_N, \mathbf{A}_O] \) and \( s^{(s)} = \begin{bmatrix} s^{(s)}_N & 0_{(n-k) \times 1} \end{bmatrix} \),
where \( \mathbf{x} = \mathbf{A}s^{(s)} \).

Proof: For \( 1 < p \leq 2 \), from (\ref{18}) and (\ref{19}), we can derive
\[
\begin{bmatrix}
|s^{(s)}_o|^{p-1} \cdot \text{sign}(s^{(s)}_o) \\
|s^{(s)}_o|^{p-1} \cdot \text{sign}(s^{(s)}_o) \\
\vdots \\
|s^{(s)}_{o_{n-k}}|^{p-1} \cdot \text{sign}(s^{(s)}_{o_{n-k}})
\end{bmatrix} + A^T_0 \cdot \alpha^{(s)} = 0,
\]
where
\[
\alpha^{(s)} = [\mathbf{A} \Pi^{-1}(s^{(s)}_N) \mathbf{A}^T]^{-1} \mathbf{x} = [\mathbf{A}_N \Pi^{-1}(s^{(s)}_N) \mathbf{A}_N^T]^{-1} \mathbf{x} \neq 0.
\]
Hence, we have
\[
s^{(s)}_O = 0_{(n-k) \times 1} \iff \begin{bmatrix}
|s^{(s)}_o|^{p-1} \cdot \text{sign}(s^{(s)}_o) \\
|s^{(s)}_o|^{p-1} \cdot \text{sign}(s^{(s)}_o) \\
\vdots \\
|s^{(s)}_{o_{n-k}}|^{p-1} \cdot \text{sign}(s^{(s)}_{o_{n-k}})
\end{bmatrix} = 0
\]
\[
\iff A^T_0 \cdot \alpha^{(s)} = 0_{(n-k) \times 1} \iff A^T_0 \cdot \alpha^{(s)} = A^T_0 \cdot [\mathbf{A} \Pi(s^{(s)}_N) \mathbf{A}^T]^{-1} \mathbf{x} = A^T_0 \cdot [\mathbf{A}_N \Pi_N(s^{(s)}_N) \mathbf{A}_N^T]^{-1} \mathbf{x} = 0_{(n-k) \times 1}.
\]

Remark 3. By analogy, the condition (42) in the Theorem [12] also plays an essential role in generating the synthetic datasets for demonstrating the Theorem [10]. Based on it, the detailed procedure is given in Algorithm 2.

In Fig 6 we demonstrated the Theorem [10] on the datasets generated randomly by Algorithm 2.

REFERENCES

[1] J. A. Tropp, “Just relax: Convex programming methods for identifying sparse signals in noise,” IEEE Trans. Information Theory, vol. 52, no. 3, pp. 1030–1051, March 2006.

[2] S. J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, “An interior-point method for large-scale \( \ell_1 \)-regularized least squares,” IEEE Journal on Selected Topics in Signal Processing, vol. 1, no. 4, pp. 606–617, December 2007.

[3] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, “Model-based compressive sensing,” IEEE Trans. Information Theory, vol. 56, no. 4, pp. 1982–2001, April 2010.

[4] S. Gleichman and Y. C. Eldar, “Blind compressed sensing,” IEEE Trans. Information Theory, vol. 57, no. 10, pp. 6958–6975, October 2011.

[5] I. F. Gorodnitsky, J. George, and B. D. Rao, “Neuromagnetic source imaging with FOCUSS: A recursive weighted minimum norm algorithm,” Electroencephalography and Clinical Neurophysiology, vol. 95, no. 4, pp. 231–251, October 1995.

[6] I. F. Gorodnitsky and B. D. Rao, “Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm,” IEEE Trans. Signal Processing, vol. 45, no. 3, pp. 600–616, 1997.

[7] B. D. Rao and K. Kreutz-Delgado, “An affine scaling methodology for best basis selection,” IEEE Trans. Signal Processing, vol. 47, no. 1, pp. 187–200, 1999.

[8] A. Cichocki and S. Amari, Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications. New York: John Wiley & Sons, 2003.

[9] B. D. Rao and K. Kreutz-Delgado, “Stable recovery of sparse overcomplete representations in the presence of noise,” IEEE Trans. Information Theory, vol. 52, no. 1, pp. 6–18, January 2006.

[10] J.-J. Fuchs, “On sparse representations in arbitrary redundant bases,” IEEE Trans. Information Theory, vol. 50, no. 6, pp. 1341–1344, June 2004.

[11] J. J. Fuchs, “Recovery of exact sparse representations in the presence of bounded noise,” IEEE Trans. Information Theory, vol. 51, no. 10, pp. 3601–3608, October 2005.

[12] R. Gribonval and P. Vandergheynst, “On the exponential convergence of matching pursuits in quasi-incoherent dictionaries,” IEEE Trans. Information Theory, vol. 52, no. 1, pp. 255–261, January 2006.

[13] B. D. Rao and K. Kreutz-Delgado, “Deriving algorithms for computing sparse solutions to linear inverse problems,” in Conference Record of the Thirty-First Asilomar Conference on Signals, Systems and Computers, vol. 1, 1997, pp. 955–959.

[14] B. D. Rao, “Signal processing with the sparseness constraint,” in Proceedings of the ICASSP, vol. III, Seattle, WA, 1998, pp. 1861–1864.

[15] B. D. Rao and K. Kreutz-Delgado, “Sparse solutions to linear inverse problems with multiple measurement vectors,” in Proceedings of the 8th IEEE Digital Signal Processing Workshop. Bryce Canyon National Park, Utah: New York: IEEE, August 1998.

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

[17] B. D. Rao, K. Engan, S. F. Cotter, J. Palmer, and K. Kreutz-Delgado, “Subset selection in noise based on diversity measure minimization,” IEEE Trans. Signal Processing, vol. 51, no. 3, pp. 760–770, March 2003.

[18] D. L. Donoho and M. Elad, “Maximal sparsity representation via \( \ell_1 \) minimization,” in Proc. National Academy Science, vol. 100, pp. 2197–2202, 2003.

[19] K. Kreutz-Delgado, J. F. Murry, B. D. Rao et al., “Dictionary learning algorithms for sparse representation,” Neural Computation, vol. 15, pp. 349–396, 2003.

[20] Y. Q. Li, A. Cichocki, and S. Amari, “Analysis of sparse representation and blind source separation,” Neural Computation, vol. 16, pp. 1193–1234, 2004.

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

[22] P. Tseng, “Further results on stable recovery of sparse overcomplete representations in the presence of noise,” IEEE Trans. Information Theory, vol. 55, no. 2, pp. 888–899, February 2009.

[23] R. Gribonval, “Sparse representations in unions of bases,” IEEE Trans. Information Theory, vol. 52, no. 12, pp. 5320–5325, December 2003.

[24] M. E. Davies and R. Gribonval, “Restricted isometry constants where \( \ell_p \) sparse recovery can fail for \( 0 < p \leq 1 \),” IEEE Trans. Information Theory, vol. 55, no. 5, pp. 2203–2214, May 2009.

[25] M. D. Plumbley, “On sparse polynomials and the recovery of sparse representations,” IEEE Trans. Information Theory, vol. 53, no. 9, pp. 3188–3195, September 2007.

[26] I. Takigawa, M. Kado, and J. Toyama, “Performance analysis of minimum \( \ell_1 \)-norm solutions for underdetermined source separation,” IEEE Trans. Signal Processing, vol. 52, no. 3, pp. 582–591, March 2004.
[27] Y. Q. Li, S. Amari, A. Cichocki, D. W. C. Ho, and S. L. Xie, “Underdetermined blind source separation based on sparse representation,” IEEE Trans. Signal Processing, vol. 54, no. 2, pp. 423–437, February 2006.

[28] P. Bofill and M. Zibulevsky, “Underdetermined blind source separation using sparse representations,” Signal Processing, vol. 81, pp. 2353–2362, 2001.

[29] R. Saab, Ö. Yılmaz, M. J. McKeown, and R. Abuharbieh, “Underdetermined anechoic blind source separation via $\ell_q$-basis-pursuit with $q < 1$,” IEEE Trans. Signal Processing, vol. 55, no. 8, pp. 4004–4017, August 2007.

[30] S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” IEEE Trans. Signal Processing, vol. 41, no. 12, pp. 3397–3415, December 1993.

[31] J. Tropp, “Greed is good: algorithmic results for sparse approximation,” IEEE Trans. Information Theory, vol. 50, no. 10, pp. 2231 – 2242, October 2004.

[32] M. Saunders, “PDCO: Primal-dual interior method for convex objectives 2002.” [Online], 2002, http://www.stanford.edu/group/SOL/software/pdco.html.

[33] D. L. Donoho and Y. Tsaig, “Fast solution of $\ell_1$-norm minimization problems when the solution may be sparse,” Department of Statistics, Stanford University, Stanford, Calif, USA, Tech. Rep. 2006-18, October 2006.

[34] I. F. Gorodnitsky, “An extension of an interior-point method for entropy minimization,” in ICASSP ’99: Proceedings of the Acoustics, Speech, and Signal Processing, 1999. on 1999 IEEE International Conference. Washington, DC, USA: IEEE Computer Society, 1999, pp. 1697–1700.

[35] B. Wohlberg, “Noise sensitivity of sparse signal representations: reconstruction error bounds for the inverse problem,” IEEE Trans. Signal Processing, vol. 51, no. 12, pp. 3053–3060, December 2003.

[36] J. Chen and X. M. Huo, “Theoretical results on sparse representations of multiple-measurement vectors,” IEEE Trans. Signal Processing, vol. 54, no. 12, pp. 4634–4643, December 2006.

[37] Z. He, A. Cichocki, R. Zdunek, and S. Xie, “Improved focuss method with conjugate gradient iterations,” IEEE transactions on signal processing, vol. 57, no. 1, pp. 399–404, January 2009.

[38] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in Advances in Neural Information Processing Systems, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. Cambridge, MA: MIT Press, 2001, vol. 13, pp. 556–562.

[39] J. J. Fuchs, “Convergence of a sparse representations algorithm applicable to real or complex data,” IEEE Journal of Selected Topics in Signal Processing, vol. 1, no. 4, pp. 598–605, December 2007.

[40] K. B. Petersen and M. S. Pedersen, “The matrix cookbook,” Oct 2008, version 20081110. [Online]. Available: [http://www2.imm.dtu.dk/pubdb/p.php?3274]

[41] L. Xu and M. Jordan, “On convergence properties of the EM algorithm for Gaussian mixtures,” Neural computation, vol. 8, no. 1, pp. 129–151, 1996.

[42] J. Nocedal and S. J. Wright, Numerical optimization, second edition ed., ser. Springer series in operations research and financial engineering, P. Glynn and S. M. Robinson, Eds. New York: Springer-Verlag, August 2006.