Sparse Decomposition Over Non-Full-Rank Dictionaries

Massoud Babaie-Zadeh, Vincent Vigneron, Christian Jutten

SPARSE DECOMPOSITION OVER NON-FULL-RANK DICTIONARIES

Massoud Babaie-Zadeh[1], Vincent Vigneron[2], and Christian Jutten[3]*

[1]Department Of Electrical Engineering, Sharif University of Technology, Tehran, Iran
[2]IBISC Laboratory, University of Evry, Evry, France
[3]GIPSA-LAB, Grenoble, France

ABSTRACT

Sparse Decomposition (SD) of a signal on an overcomplete dictionary has recently attracted a lot of interest in signal processing and statistics, because of its potential application in many different areas including Compressive Sensing (CS). However, in the current literature, the dictionary matrix has generally been assumed to be of full-rank. In this paper, we consider non-full-rank dictionaries (which are not even necessarily overcomplete), and extend the definition of SD over these dictionaries. Moreover, we present an approach which enables to use previously developed SD algorithms for this non-full-rank case. Besides this general approach, for the special case of the Smoothed ℓ^0 (SL0) algorithm, we show that a slight modification of it covers automatically non-full-rank dictionaries.

Index Terms— Atomic Decomposition, Sparse Decomposition, Compressive Sensing (CS), Overcomplete Signal Representation, Sparse Component Analysis (SCA).

1. INTRODUCTION

Let an n-dimensional vector $\mathbf{x}$ is to be decomposed as a linear combination of the vectors $\mathbf{a}_i, i = 1, \ldots, m$. After [1], the vectors $\mathbf{a}_i, i = 1, \ldots m$ are called *atoms* and they collectively form a *dictionary* over which the vector $\mathbf{x}$ is to be decomposed. We may write $\mathbf{x} = \sum_{i=1}^{m} s_i \mathbf{a}_i = \mathbf{As}$, where $\mathbf{A} \triangleq [\mathbf{a}_1, \ldots, \mathbf{a}_m]$ is the $n \times m$ dictionary (matrix) and $\mathbf{s} \triangleq (s_1, \ldots, s_m)^T$ is the $m \times 1$ vector of coefficients. If $m > n$, the dictionary is overcomplete, and the decomposition is not necessarily unique. However, the so called "Sparse Decomposition" (SD), that is, a decomposition with as much zero coefficients as possible has recently found a lot of attention in the literature because of its potential applications in many different areas. For example, it is used in Compressive Sensing (CS) [2], underdetermined Sparse Component Analysis (SCA) and source separation [3], decoding real field codes [4], image deconvolution [5], image denoising [6], elec-

tromagnetic imaging and Direction of Arrival (DOA) finding [7], and Face Recognition [8].

The sparse solution of the Underdetermined System of Linear Equations (USLE) $\mathbf{As} = \mathbf{x}$ is useful because it is unique under some conditions: Let $spark(\mathbf{A})$ denote the minimum number of columns of $\mathbf{A}$ which form a linear dependent set [9]. Then, if the USLE:

$$\mathbf{As} = \mathbf{x} \qquad (1)$$

has a solution $\mathbf{s}$ with less than $\frac{1}{2} spark(\mathbf{A})$ non-zero components, it is the unique sparsest solution [9, 10]. As a special case, if every $n \times n$ sub-matrix of $\mathbf{A}$ is invertible (which is called the Unique Representation Property or URP in [7]), then a solution of (1) with less than $(n + 1)/2$ non-zero elements is the unique sparsest solution.

For finding the sparse solution of (1), one may search for a solution for which the ℓ^0 norm of $\mathbf{s}$, i.e. the number of non-zero components of $\mathbf{s}$, is minimized. This is written as:

$$\text{minimize} \sum_{i=1}^{m} |s_i|^0 \quad \text{subject to} \quad \mathbf{As} = \mathbf{x} \qquad (2)$$

Direct solution of this problem needs a combinatorial search and is NP-hard. Consequently, many different algorithms have been proposed in recent years for finding the sparse solution of (1). Some examples are Basis Pursuit (BP) [11], Smoothed ℓ^0 (SL0) [12, 13], and FOCUSS [7]. Many of these algorithms, replace the ℓ^0 norm in (2) by another function of $\mathbf{s}$, and solve the problem:

$$\text{minimize} f(\mathbf{s}) \quad \text{subject to} \quad \mathbf{As} = \mathbf{x} \qquad (3)$$

For example, in BP, $f(\mathbf{s})$ is the ℓ^1 norm of $\mathbf{s}$ (i.e. $\sum_{i=1}^{m} |s_i|$); and in SL0, $f(\mathbf{s})$ is a smoothed measure of the ℓ^0 norm.

However, up to our best knowledge, in all of the previous works, it is explicitly or implicitly assumed that the dictionary matrix is full-rank. Note that having the URP is a more strict assumption than being full-rank, that is, a matrix which has the URP is full-rank, but a full-rank matrix has not necessarily the URP. Consider however a dictionary $\mathbf{A}$ which is not full-rank (and hence has not the URP), but $spark(\mathbf{A}) > 2$. This dictionary may still be useful for SD applications, because

a solution of (1) with less than $\frac{1}{2} spark(\mathbf{A})$ non-zero components is still unique and is the sparsest solution. As an example, $\mathbf{A} = [1, 2, -1, 1; 2, -1, 1, 0; 3, 1, 0, 1]$ is not full-rank (its 3rd row is the sum of its first two rows), but every two of its columns are linearly independent, and hence $spark(\mathbf{A}) = 3$.

On the other hand, for a non-full-rank $\mathbf{A}$, the system (1) does not even necessarily admit a solution (that is, $\mathbf{x}$ cannot be necessarily stated as a linear combination of the atoms $\mathbf{a}_i, i = 1, \ldots, m$. For example, $\mathbf{x} = (1, 2, 3.1)^T$ cannot be stated as a linear combination of the columns of the above mentioned $\mathbf{A}$, because contrary to $\mathbf{A}$, its last row (here, just the last entrie) is not the sum of its first two rows. In this case, all the algorithms based on (2) or (3) will fail, because the solution set of (1) is empty. In fact, in this case, since (1) has no solution, clearly the 'sparsest solution' of (1) makes no sense.

Non-full-rank dictionaries may be encountered in some applications. For example, in SD based classification [8], the idea is to express a new point $\mathbf{x}$ as a sparse linear combination of all data points $\mathbf{a}_i, i = 1, \ldots, m$, and assign to $\mathbf{x}$ the class of the data points $\mathbf{a}_i$ which have more influence on this representation. In this application, if for example, one of the features (one of the components of $\mathbf{a}_i$) can be written as a linear combination of the other features for all the 'data' points $\mathbf{a}_i, i = 1, \ldots, m$, then the dictionary $\mathbf{A}$ is non-full-rank. If this is also true for the new point $\mathbf{x}$, then we are in the case that (1) has solutions but $\mathbf{A}$ is non-full-rank; and if not, then (1) has no solution and our classifier (based on most current SD algorithms) will fail to provide an output.

For a non-full-rank overcomplete dictionary, one may propose to simply remove the rows of $\mathbf{A}$ that are dependent to other rows, and obtain a full-rank dictionary. This naive approach is not desirable in many applications. In Compressive Sensing (CS) language, this is like throwing away some of the measurements, which were useful in presence of measurement noise for a better estimation of $\mathbf{s}$ (recal that in parameter estimation, if several measurements are available, removing all but one does decrease the accuracy of the parameter estimation).

In this paper, we generalize the definition of sparse decomposition to non-full-rank dictionaries in Section 2, and show how to calculate it using previously known SD algorithms in Section 3. Then, in Section 4, we present a uniqueness theorem. Finally, in Section 5, for the special case of the SL0 algorithm, we will modify the algorithm itself to directly cover non-full-rank dictionaries.

2. DEFINITION OF SPARSE DECOMPOSITION

In linear algebra, when the linear system $\mathbf{A}\mathbf{s} = \mathbf{x}$ is inconsistent (underdetermined as well as overdetermined), one usually considers a Least Squares (LS) solution, that is, a solution which minimizes $\|\mathbf{A}\mathbf{s} - \mathbf{x}\|$, where $\| \cdot \|$ stands for the ℓ^2 (Eucidean) norm throughout this paper. Naturally, we define the sparse decomposition as a decomposition $\mathbf{x} \approx s_1\mathbf{a}_1 +$

$\cdots + s_m\mathbf{a}_m = \mathbf{A}\mathbf{s}$ which has the sparsest $\mathbf{s}$ among all of the (global) minimizers of $\|\mathbf{A}\mathbf{s} - \mathbf{x}\|$ (see also Lemma 2):

Definition 1. *Let $\mathcal{S}$ be the set of LS solutions of (1), that is:*

$$\mathcal{S} \triangleq \{\mathbf{s} \mid \|\mathbf{A}\mathbf{s} - \mathbf{x}\| \text{ is minimized}\}. \tag{4}$$

By a sparse LS solution of (1), we mean the $\mathbf{s} \in \mathcal{S}$ which has the minimum number of non-zero components, that is:

$$\operatorname*{argmin}_{\mathbf{s}} \sum_{i=1}^{m} |s_i|^0 \text{ subject to } \mathbf{s} \in \mathcal{S} \tag{5}$$

Note that the constraint $\mathbf{A}\mathbf{s} = \mathbf{x}$ in (2) has been replaced by $\mathbf{s} \in \mathcal{S}$ in (5). If (1) admits a solution, $\mathcal{S}$ will be the set of solutions of (1), and the above definition is the same as (2).

Remark. The above definition, as well as all of the results of this paper, also holds for the case $n \geq m$ (determined or overdetermined case). In this case, non-full-rank means $\text{rank}(\mathbf{A}) < m$ (which ensures that $\mathcal{S}$ is not a singleton).

3. HOW TO FIND THE SPARSE LS SOLUTION: A GENERAL APPROACH

In this section, we present a simple approach that enables to use all of the previously known SD algorithms for the non-full-rank case as well. The idea is to express $\mathbf{s} \in \mathcal{S}$ equivalently as $\mathbf{B}\mathbf{s} = \mathbf{y}$, where $\mathbf{B}$ is a full-rank[1] $r \times m$ matrix, and $\mathbf{y}$ is a length r vector (r is the rank of $\mathbf{A}$). Then (5) is converted to a full-rank (and underdetermined) problem (2), for which there are many already developed SD algorithms.

To do this, we first characterize the set $\mathcal{S}$ defined in (4). Let $\mathbf{A}_{n \times m} = \mathbf{U}_{n \times n}\mathbf{S}_{n \times m}\mathbf{V}^H_{m \times m}$ be the Singular Value Decomposition (SVD) of $\mathbf{A}$, where the subscripts denote the dimensions, $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices, and:

$$\mathbf{S} = \begin{bmatrix} \boldsymbol{\Sigma}_{r \times r} & \mathbf{0}_{r \times (m-r)} \\ \mathbf{0}_{(n-r) \times r} & \mathbf{0}_{(n-r) \times (m-r)} \end{bmatrix} \tag{6}$$

in which $\boldsymbol{\Sigma} = \text{diag}(\sigma_1, \ldots, \sigma_r)$ where $\sigma_i > 0, i = 1, \ldots, r$ are the singular values of $\mathbf{A}$. We are looking for all $\mathbf{s}$ which minimize the cost function $\mathcal{C} \triangleq \|\mathbf{A}\mathbf{s} - \mathbf{x}\|^2$. Then:

$$\begin{aligned} \mathcal{C} &= \|\mathbf{U}\mathbf{S}\mathbf{V}^H\mathbf{s} - \mathbf{x}\|^2 & \text{(where } \mathbf{s}' \triangleq \mathbf{V}^H\mathbf{s}) \\ &= \|\mathbf{U}(\mathbf{S}\mathbf{s}' - \mathbf{U}^H\mathbf{x})\|^2 & \text{(where } \mathbf{x}' \triangleq \mathbf{U}^H\mathbf{x}) \\ &= \|\mathbf{S}\mathbf{s}' - \mathbf{x}'\|^2 & \text{(}\mathbf{U}\text{ is a rotation)} \quad (7) \\ &= \sum_{i=1}^{r} |\sigma_i s_i' - x_i'|^2 + \sum_{i=r+1}^{n} |x_i'|^2 \end{aligned}$$

The second term of $\mathcal{C}$ does not depend on $\mathbf{s}'$ (and hence $\mathbf{s}$), and the first term can be made equal to zero by setting $s_i' = x_i'/\sigma_i, i = 1, \ldots, r$. In other words, $\mathbf{s}$ is a minimizer of

---

[1]Note that the normal equations $\mathbf{A}^H\mathbf{A}\mathbf{s} = \mathbf{A}^H\mathbf{x}$ cannot be used for this aim: although it is equivalent to $\mathbf{s} \in \mathcal{S}$, its coefficient matrix is non-full-rank.

$\mathcal{C}$ if and only if $\mathbf{s}' \triangleq \mathbf{V}^H \mathbf{s}$ satisfies the following system of r equations (note that $s'_{r+1}, \ldots, s'_m$ are arbitrary):

$$\begin{bmatrix} \sigma_1 & & & 0 & \cdots & 0 \\ & \ddots & & \vdots & \ddots & \vdots \\ & & \sigma_r & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} s'_1 \\ s'_2 \\ \vdots \\ \vdots \\ s'_m \end{bmatrix} = \begin{bmatrix} x'_1 \\ \vdots \\ x'_r \end{bmatrix} \quad (8)$$

or in matrix form $\left[\boldsymbol{\Sigma}_{r \times r}, \mathbf{0}_{r \times (m-r)} \right] \mathbf{V}^H \mathbf{s} = \mathbf{x}'_1$, where $\mathbf{x}'_1 \triangleq (x'_1, \ldots, x'_r)^T$. Note also that if $\mathbf{U} = [\mathbf{U}_1, \mathbf{U}_2]$, where $\mathbf{U}_1$ is composed of the first r columns of $\mathbf{U}$, and $\mathbf{U}_2$ is the remaining $n - r$ columns, then $\mathbf{x}' \triangleq \mathbf{U}^H \mathbf{x} = (\mathbf{U}_1^H \mathbf{x}, \mathbf{U}_2^H \mathbf{x})^T$, and hence $\mathbf{x}'_1 = \mathbf{U}_1^H \mathbf{x}$. In summary, we have proved the following theorem:

Theorem 1. *Let r be the rank of the dictionary matrix $\mathbf{A}$, and let its SVD be $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^H$, where $\mathbf{S}$ is of the form (6). Let also that $\mathbf{U}_1$ be composed of the first r columns of $\mathbf{U}$, and $\mathcal{S}$ be defined according to (4). Then $\mathbf{s} \in \mathcal{S}$ if and only if:*

$$\mathbf{A}_1 \mathbf{s} = \mathbf{x}'_1 \quad (9)$$

where $\mathbf{A}_1 \triangleq \left[\boldsymbol{\Sigma}_{r \times r}, \mathbf{0}_{r \times (m-r)} \right] \mathbf{V}^H$, and $\mathbf{x}'_1 \triangleq \mathbf{U}_1^H \mathbf{x}$.

Using the above theorem, a general approach for finding the sparsest LS solution (5) is *to replace the constraint $\mathbf{s} \in \mathcal{S}$ by (9), and then to use any of the previously known SD algorithms* which had been developed for full-rank dictionaries.

4. UNIQUENESS OF THE SPARSE LS SOLUTION

In this section we first state some more characteristics of $\mathcal{S}$, and then a uniqueness result for the sparse LS solution.

The following lemma is very famous for LS solutions of overdetermined linear systems, but it holds for inconsistent underdetermined systems, too [14, p.279]:

Lemma 1. *Among the members of $\mathcal{S}$, the vector $\mathbf{s}_{\min} \triangleq \mathbf{A}^\dagger \mathbf{x}$, where $\mathbf{A}^\dagger$ is the Moore-Penrose pseudoinverse of $\mathbf{A}$, has minimum Euclidean norm.*

Although the above result is already known, we prefer to re-state its proof here, because it comes directly from (8):

Proof. Since $\|\mathbf{s}'\| = \|\mathbf{V}^H \mathbf{s}\| = \|\mathbf{s}\|$, for having minimum $\|\mathbf{s}'\|$, the free parameters $s'_{r+1}, \ldots, s'_m$ in (8) should be set equal to zero. This gives $\mathbf{s}_{\min} = \mathbf{A}^\dagger \mathbf{x}$, where $\mathbf{A}^\dagger \triangleq \mathbf{V}\mathbf{S}^\dagger \mathbf{U}^H \mathbf{x}$, and:

$$\mathbf{S}^\dagger_{m \times n} \triangleq \begin{bmatrix} \boldsymbol{\Sigma}^{-1}_{r \times r} & \mathbf{0}_{r \times (n-r)} \\ \mathbf{0}_{(m-r) \times r} & \mathbf{0}_{(m-r) \times (n-r)} \end{bmatrix} \quad (10)$$

$\square$

Lemma 2. *A general form for the members of $\mathbf{s} \in \mathcal{S}$ is:*

$$\mathbf{s} = \mathbf{A}^\dagger \mathbf{x} + (\mathbf{I} - \mathbf{A}^\dagger \mathbf{A})\mathbf{z} \quad (11)$$

where $\mathbf{z}$ is an arbitrary m-dimensional vector.

The above lemma, too, is already known ([15, p.125] and [16, p.49]). However, we prefer to give another proof here based on the general form (8).

Proof. Let $\mathbf{s}$ be of the form (11). Left multiplying both sides of (11) by $\mathbf{A}$ and using $\mathbf{A}\mathbf{A}^\dagger\mathbf{A} = \mathbf{A}$, we have $\mathbf{A}\mathbf{s} = \mathbf{A}(\mathbf{A}^\dagger \mathbf{s}) = \mathbf{A}\mathbf{s}_{\min}$. That is, the distance of $\mathbf{A}\mathbf{s}$ to $\mathbf{x}$ is equal to the distance of $\mathbf{A}\mathbf{s}_{\min}$ to $\mathbf{x}$, and hence $\mathbf{s} \in \mathcal{S}$.

Now letting $\mathbf{s} \in \mathcal{S}$, we must show that there is a $\mathbf{z}$ such that $\mathbf{s}$ can be written as (11). From (8), a general form for $\mathbf{s}' = \mathbf{V}^H \mathbf{s}$ is $\mathbf{s}' = (x'_1/\sigma_1, \ldots, x'_r/\sigma_r, \alpha_{r+1}, \ldots, \alpha_m)^T$, where $\alpha_{r+1}, \ldots, \alpha_m$ are arbitrary scalars. It can be written as $\mathbf{s}' = (x'_1/\sigma_1, \ldots, x'_r/\sigma_r, 0, \ldots, 0)^T + (0, \ldots, 0, \alpha_{r+1}, \ldots, \alpha_m)^T = \mathbf{S}^\dagger \mathbf{x}' + (0, \ldots, 0, \alpha_{r+1}, \ldots, \alpha_m)^T$, and hence:

$$\mathbf{s} = \mathbf{V}\mathbf{s}' = \mathbf{V}\mathbf{S}^\dagger\mathbf{U}^H\mathbf{x} + \mathbf{V}(0, \ldots, 0, \alpha_{r+1}, \ldots, \alpha_m)^T$$
$$= \mathbf{A}^\dagger\mathbf{x} + (\alpha_{r+1}\mathbf{v}_{r+1} + \cdots + \alpha_m\mathbf{v}_m)$$

where $\mathbf{v}_i$ denotes the i-th column of $\mathbf{V}$. We have therefore to show that there is a $\mathbf{z}$ such that $(\mathbf{I} - \mathbf{A}^\dagger\mathbf{A})\mathbf{z} = \sum_{i=r+1}^m \alpha_i \mathbf{v}_i$. Using the definition of $\mathbf{A}^\dagger$ (just before (10)), it is easy to see that $\mathbf{A}^\dagger\mathbf{A} = \sum_{i=1}^r \mathbf{v}_i\mathbf{v}_i^H$, and hence from orthonormality of $\mathbf{V}$ we have $\mathbf{I} - \mathbf{A}^\dagger\mathbf{A} = \sum_{i=r+1}^m \mathbf{v}_i\mathbf{v}_i^H$. Consequently, it remains to show that for every $\alpha_{r+1}, \ldots, \alpha_m$, there is a $\mathbf{z}$ such that $\mathbf{v}_i^H\mathbf{z} = \alpha_i$ for $i = r+1, \ldots, m$. This is evident, because the linear system $(\mathbf{v}_{r+1}, \ldots, \mathbf{v}_m)^H\mathbf{z} = (\alpha_{r+1}, \ldots, \alpha_m)^T$ is underdetermined and full-rank (because of the orthogonality of $\mathbf{v}_i$'s), and hence has an infinite number of solutions. $\square$

Lemma 3. *$\mathbf{s} \in \mathcal{S}$ if and only if $\mathbf{A}\mathbf{s} = \mathbf{A}\mathbf{s}_{\min}$, where $\mathbf{s}_{\min} = \mathbf{A}^\dagger\mathbf{x}$.*

Proof. $\mathbf{s} \in \mathcal{S} \Rightarrow \mathbf{A}\mathbf{s} = \mathbf{A}\mathbf{s}_{\min}$: Left multiply both sides of (11) by $\mathbf{A}$, and use $\mathbf{A}\mathbf{A}^\dagger\mathbf{A} = \mathbf{A}$.

$\mathbf{A}\mathbf{s} = \mathbf{A}\mathbf{s}_{\min} \Rightarrow \mathbf{s} \in \mathcal{S}$: This is because the distance of $\mathbf{A}\mathbf{s}$ to $\mathbf{x}$ is the same as the distance of $\mathbf{A}\mathbf{s}_{\min}$ to $\mathbf{x}$. $\square$

Theorem 2 (Uniqueness of the sparse LS solution). *If there is a $\mathbf{s} \in \mathcal{S}$ for which $\|\mathbf{s}\|_0 < \frac{1}{2}spark(\mathbf{A})$, then it is the unique sparsest LS solution of (1).*

Proof. Suppose there are $\mathbf{s}_1 \in \mathcal{S}$ and $\mathbf{s}_2 \in \mathcal{S}$ for which $\|\mathbf{s}_1\|_0 < \frac{1}{2}spark(\mathbf{A})$ and $\|\mathbf{s}_2\|_0 < \frac{1}{2}spark(\mathbf{A})$. Then from Lemma 3, we have $\mathbf{A}\mathbf{s}_1 = \mathbf{A}\mathbf{s}_2(= \mathbf{A}\mathbf{s}_{\min})$, and hence $\mathbf{A}\mathbf{y} = \mathbf{0}$, where $\mathbf{y} = \mathbf{s}_1 - \mathbf{s}_2$. Now $\mathbf{y} \neq \mathbf{0}$ is a contradiction, because $\|\mathbf{y}\|_0 < spark(\mathbf{A})$, and a linear combination of $\|\mathbf{y}\|_0$ columns of $\mathbf{A}$ cannot be equal to zero. Hence $\mathbf{y} = 0 \Rightarrow \mathbf{s}_1 = \mathbf{s}_2$. $\square$

5. THE SL0 ALGORITHM

SL0 algorithm [12, 13] is a SD algorithm with two main features: 1) it is very fast, 2) it tries to directly minimize the ℓ^0 norm (and hence does not suffer from replacing ℓ^0 by asymptotic equivalents). For this algorithm, there is no need to use the general approach of Section 3, and the algorithm may be modified to directly work for the sparse LS solutions.

The basic idea of SL0 is to use a smooth measure of the ℓ^0 norm ($f(\mathbf{s})$), and solve (3) by steepest descent. To take into account the constraint $\mathbf{As} = \mathbf{x}$, each iteration of (the full-rank) SL0 is composed of:

- Minimization: $\mathbf{s} \leftarrow \mathbf{s} - \mu \nabla f$

- Projection: Project $\mathbf{s}$ onto $\{\mathbf{s} | \mathbf{As} = \mathbf{x}\}$:

$$\mathbf{s} \leftarrow \mathbf{s} - \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}(\mathbf{As} - \mathbf{x}) \qquad (12)$$

For extending SL0 to non-full-rank dictionaries, the projection step should be modified: $\mathbf{s}$ should be projected onto $\mathcal{S}$ instead of $\{\mathbf{s} | \mathbf{As} = \mathbf{x}\}$. We show here that this can be done simply by replacing $\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}$ in (12) by $\mathbf{A}^\dagger$. This can be seen from the following lemma:

Lemma 4. *Let $\mathbf{s}_0$ be given. Among all the members of $\mathcal{S}$, the minimizer of $\|\mathbf{s} - \mathbf{s}_0\|$ is:*

$$\mathbf{s} = \mathbf{s}_0 + \mathbf{A}^\dagger(\mathbf{x} - \mathbf{As}_0) \qquad (13)$$

Proof. We write $\|\mathbf{As} - \mathbf{x}\| = \|\mathbf{A}(\mathbf{s} - \mathbf{s}_0) - (\mathbf{x} - \mathbf{As}_0)\| = \|\mathbf{Ay} - \mathbf{b}\|$, where $\mathbf{y} \triangleq \mathbf{s} - \mathbf{s}_0$ and $\mathbf{b} \triangleq \mathbf{x} - \mathbf{As}_0$. Consequently, any minimizer of $\|\mathbf{Ay} - \mathbf{b}\|$ corresponds to a minimizer of $\|\mathbf{As} - \mathbf{x}\|$, and vice versa. Moreover (see Lemma 1), among all minimizers of $\|\mathbf{Ay} - \mathbf{b}\|$, the Euclidean norm $\|\mathbf{y}\| = \|\mathbf{s} - \mathbf{s}_0\|$ is minimum for $\mathbf{y} = \mathbf{A}^\dagger\mathbf{b}$, which corresponds to (13). $\square$

In summary, *for extending the SL0 algorithm to non-full-rank dictionaries, just replace $\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}$ in (12) by $\mathbf{A}^\dagger$.*

6. CONCLUSIONS AND PERSPECTIVES

The main aim of this paper was extension of definition of sparse decomposition, uniqueness result, and algorithms to cover non-full-rank dictionaries (and to prevent some algorithms to be blocked when encounter a non-full-rank dictionary). As stated before the last paragraph of Section 1, however, this approach may be also thought as a denoising approach in CS by means of repeated measurements (recall also Remark 1), a point which is under more investigation in our group. As a simple experiment for $m = 100$ and $n = 30$ with a zero mean and Gaussian measurement noise with variance 0.01, by repeating each measurement two times and using (9), the averaged estimation quality of BP increased from 19.8dB to 22.5dB.

7. REFERENCES

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

[2] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.

[3] R. Gribonval and S. Lesage, "A survey of sparse component analysis for blind source separation: principles, perspectives, and new challenges," in *Proceedings of ESANN'06*, April 2006, pp. 323–330.

[4] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Transactions Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[5] M. A. T. Figueiredo and R. D. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 906–916, 2003.

[6] M. Elad, "Why simple shrinkage is still relevant for redundant representations?," *IEEE Transactions on Image Processing*, vol. 52, no. 12, pp. 5559–5569, 2006.

[7] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS, a re-weighted minimum norm algorithm," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 600–616, March 1997.

[8] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Yi Ma, "Robust face recognition via sparse representation," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, , no. 2, pp. 210–227, February 2009.

[9] D. L. Donoho and M. Elad, "Maximal sparsity representation via ℓ^1 minimization," *the Proc. Nat. Aca. Sci.*, vol. 100, no. 5, pp. 2197–2202, March 2003.

[10] R. Gribonval and M. Nielsen, "Sparse decompositions in unions of bases," *IEEE Trans. Inform. Theory*, vol. 49, no. 12, pp. 3320–3325, Dec. 2003.

[11] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal l^1-norm solution is also the sparsest solution," Tech. Rep., 2004.

[12] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "Fast sparse representation based on smoothed l0 norm," in *Proceedings of* ICA'07, LNCS 4666, London, UK, September 2007, pp. 389–396.

[13] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed ℓ^0 norm," *IEEE Trans. on Signal Proc*, vol. 57, no. 1, pp. 289–301, January 2009.

[14] D. S. Watkins, *Fundamentals of Matrix Computations*, John Wiley & Sons, 2nd Edition, 2002.

[15] F. Rotella and P. Borne, *Théorie et pratique du calcul matriciel*, Éditions technip, 1995, in French.

[16] C.R. Rao and S.K. Mitra, *Generalized Inverse of Matrices and its applications*, John Wiley & Sons, 1971.