COMPLEX-VALUED SPARSE REPRESENTATION BASED ON SMOOTHED ℓ^0 NORM

G. H. Mohimani, M. Babaie-Zadeh[*]

Sharif University of Technology
Department Of Electrical Engineering
Tehran,Iran

C. Jutten

Institut National Polytechnique de Grenoble (INPG)
Labratoire des Images et de Signaux (LIS)
Grenoble,France

ABSTRACT

In this paper we present an algorithm for complex-valued sparse representation. In our previous work we presented an algorithm for Sparse representation based on smoothed ℓ^0-norm. Here we extend that algorithm to complex-valued signals. The proposed algorithm is compared to FOCUSS algorithm and it is experimentally shown that the proposed algorithm is about two or three orders of magnitude faster than FOCUSS while providing approximately the same accuracy.

Index Terms— complex-valued sparse component analysis, over-complete atomic decomposition.

1. INTRODUCTION

Obtaining sparse solutions of under-determined systems of linear equations is of significant importance in signal processing and statistics. Some of the potential applications include Sparse Component Analysis (SCA) [1, 2, 3, 4], atomic decomposition on overcomplete dictionaries [5, 6] and decoding real field code [7]. Despite recent theoretical developments [8, 1, 2], the computational cost of the methods has remained as the main restriction, especially for large systems (large number of unknowns/equations).

The aim of sparse representation is to obtain the sparsest solution $\mathbf{s} = [s_1, \cdots, s_n]^T$ of $\mathbf{x} = \mathbf{A}\mathbf{s}$, i.e. the solution in which most of the elements s_j, $1 \leq j \leq n$ are nearly zero. We may search for a solution of the system having minimal ℓ^0 norm, i.e. minimum number of nonzero components. It is usually stated in the literature [5, 3, 4, 2] that searching the minimum ℓ^0 norm is an intractable problem as the dimension increases (because it requires a combinatorial search), and as the noise increases (because any small amount of noise completely changes the ℓ^0 norm of a vector). Consequently, the researchers look for other approaches to find sparse solution of $\mathbf{x} = \mathbf{A}\mathbf{s}$ which are tractable. One of the most successful approaches is Basis Pursuit (BP) [6, 8, 4, 2, 9] which finds the minimum ℓ^1 norm (that is, the solution of $\mathbf{x} = \mathbf{A}\mathbf{s}$ for which $\sum_j |s_j|$ is minimized). Such a solution can be easily found by Linear Programming (LP) methods. The idea of Basis Pursuit is based on the observation that for large systems of equations, the minimum ℓ^1 norm solution is also the minimum ℓ^0 norm solution [8, 6, 9]. By utilizing fast LP algorithms, specifically interior-point LP solvers, large-scale problems with thousands of sources and mixtures become tractable. However, it is still very slow, and in the recent years several authors have proposed improvements for BP (for speeding up the algorithm and for handling the noisy case) [5]. Another family of algorithms is Iterative Re-weighted Least Squares (IRLS), with FOCUSS [10] as an important member. These are faster than BP, but their estimation quality is worse (especially if the number of non-zero elements of the sparsest solution is not very small). Another approach is Matching Pursuit (MP) [11, 2] which is very fast, but is a greedy algorithm and does not always provide good estimation of the sources.

In many applications, the sparsity of signals is in the frequency domain. In these applications, Fourier transform is a necessary pre-processing which gives rise to complex systems and signals. However, most of the proposed algorithms (such as BP methods) cannot handle the complex case.

In our previous work [12], a new sparse representation approach was proposed that provided a considerable reduction in complexity. In that article, we presented SL0, a fast method for finding the sparse solution of an under-determined system of linear equations, which was based on minimization of ℓ^0 norm. However, the method was restricted to real signals. In this article we modify that work and extend it to the complex-valued system and signal case and justify its performance in complex case. The paper is organized as follows. The next section introduces the main idea. The algorithm is then stated in Section 3. Finally, Section 4 provides some experimental results of our algorithm and its comparison with FOCUSS.

2. THE MAIN IDEA

The main idea of SL0 approach [12] is to approximate the ℓ^0 norm by a smooth (continuous) function, which lets us to use gradient based methods for its minimization and also solves the problem of sensitivity of ℓ^0 norm to noise. In this section

[*]This work has been partially funded by Sharif university of Technology, Advanced Communication Research Institute (ACRI), ISMO and Iran NSF (INSF).

we introduce a family of smooth complex approximators of ℓ^0 norm, whose optimization results in a fast algorithm for finding the sparse solution while preserving noise robustness.

The ℓ^0 norm of $\mathbf{s} = [s_1 \dots s_n]^T$ is defined as the number of non-zero components of $\mathbf{s}$. In other words, if we define

$$\nu(s) = \begin{cases} 1 & s \neq 0 \\ 0 & s = 0 \end{cases} \tag{1}$$

then the ℓ^0 norm is $\|\mathbf{s}\|_0 = \sum_{j=1}^{n} \nu(s_j)$. It is clear that the discontinuities of the ℓ^0 norm are caused by discontinuities of the function ν. If we replace ν by a smooth estimation of it, we obtain a smooth estimation of the ℓ^0 norm. This also provides some robustness to additive noise.

Different functions may be utilized for this aim. In this paper, we extend the zero-mean Gaussian family of functions used in [12] to complex zero-mean Gaussian family of functions which seems to be very useful for this application, because of their differentiability. By defining:

$$f_\sigma(s) = \exp(-|s|^2/2\sigma^2) = \exp(-(s_r^2 + s_i^2)/2\sigma^2), \tag{2}$$

where $|s|$, s_r and s_i represent the module, real and imaginary parts of $s \in \mathbb{C}$, we have:

$$\lim_{\sigma \to 0} f_\sigma(s) = \begin{cases} 1 & s = 0 \quad (s_r = s_i = 0) \\ 0 & s \neq 0 \quad (s_r \neq 0 \text{ or } s_i \neq 0) \end{cases} . \tag{3}$$

Consequently, $\lim_{\sigma \to 0} f_\sigma(s) = 1 - \nu(s)$, and therefore by defining $F_\sigma(\mathbf{s}) = \sum_{j=1}^{n} f_\sigma(s_j)$, we have:

$$\lim_{\sigma \to 0} F_\sigma(\mathbf{s}) = \sum_{j=1}^{n} (1 - \nu(s_j)) = n - \|\mathbf{s}\|_0. \tag{4}$$

and as a result, $\|\mathbf{s}\|_0 \approx n - F_\sigma(\mathbf{s})$. The value of σ specifies a trade-off between accuracy and smoothness of the approximation: the smaller σ, the better approximation, and the larger σ, the smoother approximation. From (4), minimization of the ℓ^0 norm is equivalent to maximization of F_σ for sufficiently small σ. This maximization should be done on the affine set $\mathcal{S} = \{\mathbf{s} \mid \mathbf{x} = \mathbf{A}\mathbf{s}\}$.

For small values of σ, F_σ contains a lot of local maxima. Consequently, it is very difficult to directly maximize this function for very small values of σ. However, as the value of σ grows, the function becomes smoother and smoother, and for sufficiently large values of σ, as we will see, there is no local maxima.

Our idea for escaping from local maxima is then to decrease the value of σ gradually[1]: for each value of σ we use a steepest ascent algorithm for maximizing F_σ, and *the initial value of this steepest ascent algorithm is the maximizer of F_σ obtained for the previous (larger) value of σ*. Since the value of σ changes slowly, the steepest ascent algorithm is initialized not far from the actual maximum, and hence, it is less likely to get trapped into local maxima.

Remark. Equation (4) proposes that $F_\sigma(\cdot)$ can be seen as a 'smooth measure of sparsity' of a vector (especially for small values of σ): the sparser $\mathbf{s}$, the larger $F_\sigma(\mathbf{s})$. In fact, $F_\sigma(\mathbf{s})$ is approximately the number of elements of $\mathbf{s}$ which have significant values (compared with σ), and for very small σ's it is simply the number of non-zero elements of $\mathbf{s}$.

```
• Initialization:
    1. Choose an arbitrary solution from the feasible set $\mathcal{S}$, $\mathbf{v}_0$,
       e.g.the minimum $\ell^2$ norm solution of $\mathbf{x} = \mathbf{A}\mathbf{s}$ obtained
       by pseudo-inverse (see the text).
    2. Choose a suitable decreasing sequence for $\sigma$,
       $[\sigma_1 \dots \sigma_K]$.
• for $k = 1, \dots, K$:
    1. Let $\sigma = \sigma_k$.
    2. Maximize (approximately) the function $F_\sigma$ on the feasi-
       ble set $\mathcal{S}$ using $L$ iterations of the steepest ascent algo-
       rithm (followed by projection onto the feasible set):
        – Initialization: $\mathbf{s} = \mathbf{v}_{k-1}$.
        – for $j = 1 \dots L$ (loop $L$ times):
            (a) Let: $\Delta\mathbf{s} =$
                $[s_1 \exp(\frac{-|s_1|^2}{2\sigma_k^2}), \dots, s_n \exp(\frac{-|s_n|^2}{2\sigma_k^2})]^T$
            (b) Let $\mathbf{s} \leftarrow \mathbf{s} - \mu\Delta\mathbf{s}$ (where $\mu$ is a small positive
                constant).
            (c) Project $\mathbf{s}$ back onto the feasible set $\mathcal{S}$:
                    $\mathbf{s} \leftarrow \mathbf{s} - \mathbf{A}^H(\mathbf{A}\mathbf{A}^H)^{-1}(\mathbf{A}\mathbf{s} - \mathbf{x})$
    3. Set $\mathbf{v}_k = \mathbf{s}$.
• Final answer is $\mathbf{s} = \mathbf{v}_l$.
```

Fig. 1. The final algorithm.

3. THE ALGORITHM

The final algorithm for complex case is presented in Fig. 1, which is a modification of the real case algorithm of [12]. As indicated in the algorithm, the final value of the previous estimation is used for the initialization of the next steepest ascent. By choosing a slowly decreasing sequence of σ, we may escape from getting trapped into local maxima, and obtain the sparsest solution.

Remark 1. The internal loop (steepest ascent for a fixed σ) is repeated a fixed and small number of times (L). In other words, for increasing the speed, we do not wait for the (internal loop of the) steepest ascent algorithm to converge. This may be justified by gradual decrease in value of σ, and the fact that for each value, we do not need the exact maximizer of F_σ. All we need, is to enter a region near the (absolute) maximizer of F_σ for escaping from its local maximizers.

[1]The idea for optimizing a non-convex function is called Graduated Non-Convexity (GNC) [13].

Remark 2. Steepest ascent consists of iterations of the form $\mathbf{s} \leftarrow \mathbf{s} + \mu_k \nabla F_\sigma(\mathbf{s})$. Here, the step-size parameters μ_k should be decreasing, i.e. for smaller values of σ, smaller values of μ_k should be applied. This is because for smaller values of σ, the function F_σ is more 'fluctuating', and hence smaller step-sizes should be used for its maximization. In fact, we may think about changing the value of σ in (2) as looking at the same curve (or surface) at different 'scales', where the scale is proportional to σ. For having equal (i.e. proportional) steps of the steepest ascent algorithm in these different scales, it is not difficult to show that μ_k should be proportional to σ^2 . Letting $\mu_k = \mu\sigma_k^2$, for some constant μ, we obtain $\mathbf{s} \leftarrow \mathbf{s} - \mu\Delta\mathbf{s}$ as stated in the algorithm of Fig. 1, where $\Delta\mathbf{s} \triangleq -\sigma^2 \nabla F_\sigma(\mathbf{s})$.

Remark 3. The algorithm may work by initializing $\mathbf{v}_0$ (initial estimation of the sparse solution) to an arbitrary solution of $\mathbf{x} = \mathbf{A}\mathbf{s}$. However, the best initial value of $\mathbf{v}_0$ is the minimum ℓ^2 norm solution of $\mathbf{x} = \mathbf{A}\mathbf{s}$, which is given by the pseudo-inverse of $\mathbf{A}$. It is because this solution is the (unique) maximizer of $F_\sigma(\mathbf{s})$ on the feasible set $\mathcal{S}$, where σ tends to infinity[2]. This is formally stated in the following theorem (for the proof, refer to [12]).

Theorem 1 The solution of the problem:

$$\text{Maximize } F_\sigma(\mathbf{s}) \text{ subject to } \mathbf{x} = \mathbf{A}\mathbf{s},$$

where $\sigma \to \infty$, is the minimum ℓ^2 norm solution of $\mathbf{x} = \mathbf{A}\mathbf{s}$, that is, $\mathbf{s} = \mathbf{A}^H(\mathbf{A}\mathbf{A}^H)^{-1}\mathbf{x}$.

Remark 4. Having initiated the algorithm with the minimum ℓ^2 norm solution (which corresponds to $\sigma \to \infty$), the next value for σ (i.e. σ_1) may be chosen about two to four times of the maximum absolute value of the obtained sources ($\max_j |s_j|$). To see the reason, note first that:

$$\exp(-s_j^2/2\sigma^2) = \begin{cases} 1 & , \text{if } |s_j| \ll \sigma \\ 0 & , \text{if } |s_j| \gg \sigma \end{cases} . \tag{5}$$

Consequently, if we take $\sigma > 4\max_j |s_j|$ for all $1 \le j \le n$, then $\exp(-s_j^2/2\sigma^2) > 0.96 \approx 1$, and comparison with (5) shows that this value of σ acts virtually like infinity for all values of s_j, $1 \le j \le n$. For next values of σ_k, we have used $\sigma_k = \alpha\sigma_{k-1}$, where α is usually between 0.5 and 1.

Remark 5. In applications where the zeros in the sparsest $\mathbf{s}$ are exactly zero, σ can be decreased arbitrarily. In fact, in this case, its minimum value is determined by the desired accuracy. For applications in which inactive elements of $\mathbf{s}$ are small but not exactly zero (say that the 'source' vector is noisy), the smallest σ should be about one to two times of (a rough estimation of) the energy of this noise. This is because, while σ is in this range, (5) shows that the cost function treats small (noisy) samples as zeros (i.e. for which $f_\sigma(s_i) \approx 1$). However, below this range, the algorithm tries to 'learn' these noisy values, and moves away from the true answer.

[2]In another point of view, one may think about the minimum ℓ^2 norm solution as a rough estimate of the sparse solution, which will be modifi ed in the future iterations of the algorithm.

Table 1. Progress of the method for a problem with $n = 1000$, $m = 400$ and $k = 100$ ($p = 0.1$).

itr. #	σ	MSE	SNR (dB)
1	1	$7.89\,e-2$	3.52
2	0.5	$2.57\,e-2$	8.39
3	0.2	$3.96\,e-3$	16.51
4	0.1	$2.77\,e-3$	18.06
5	0.05	$8.27\,e-4$	23.32
6	0.02	$4.18\,e-4$	26.28
7	0.01	$4.90\,e-4$	25.59

algorithm	Time	SNR	STD	Minimum
SL0(k=100)	0.145	26.15	0.56	24.91
FOCUSS(k=100)	64.8	27.93	1.34	21.01
SL0(k=150)	0.127	22.91	3.16	13.56
FOCUSS(k=150)	47.8	22.49	3.95	13.54

4. EXPERIMENTAL RESULTS

In this section, we justify the performance of the presented approach and compare it with FOCUSS [10]. Sparse sources are artificially created using Complex Bernoulli-Gaussian model:

$$s_j \sim p \cdot \mathcal{N}(0, \sigma_{\text{on}}) + (1-p) \cdot \mathcal{N}(0, \sigma_{\text{off}}), \tag{6}$$

where p denotes probability of activity of the sources. σ_{on} and σ_{off} are the standard deviations of the sources in active and inactive mode, respectively. In order to have sparse sources, the parameters are required to satisfy the conditions $\sigma_{\text{off}} \ll \sigma_{\text{on}}$ and $p \ll 1$. Setting $\sigma_{\text{off}} = 0$ results in a spiky model. In the simulation σ_{on} is fixed to 1. Each column of the mixing matrix is randomly generated using the normal distribution which is then normalized to unity.

Using (6), the number of active sources has a binomial distribution with average np. Let define $k = np$. In our simulations, we prefer to work with k instead of working with p directly.

The mixtures are generated using the noisy model $\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{n}$, where $\mathbf{n}$ is an additive white complex Gaussian noise with variance $\sigma_n \mathbf{I}_m$ ($\mathbf{I}_m$ is $m \times m$ identity matrix). σ_n is set to 0.02 in the simulation. The values used for the experiment are $n = 1000$, $m = 400$, $k = 100$ (average number of active sources) or equivalently $p = 0.1$, $\sigma_{\text{off}} = 0$, $\sigma_{\text{on}} = 1$, $\sigma_n = 0.02$ and the sequence of σ is fixed to [1, 0.5, 0.2, 0.1, 0.05, 0.02, 0.01]. μ is set equal to 2.5. For each value of σ the gradient-projection loop (the internal loop) is performed three times (i.e. $L = 3$).

Table 1 shows the gradual improvement in the output SNR after each iteration, for a typical run of the algorithm. Moreover, for this run, total time and final SNR have been shown

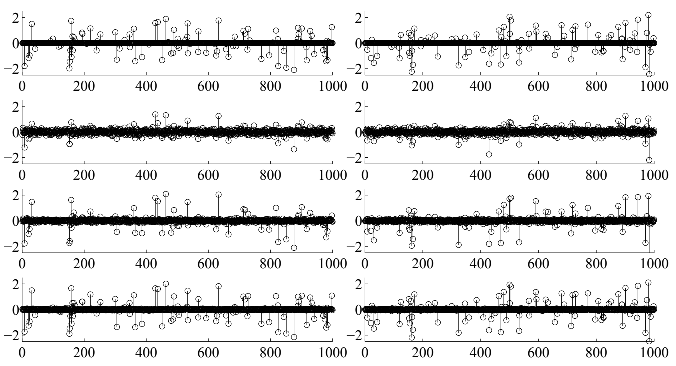

Fig. 2. Evolution of the algorithm toward the solution: $n = 1000$, $m = 400$ and $p = 0.1$. From top to bottom, first plot corresponds to the real and imaginary part of actual source, and their estimation at the first level ($\sigma = 1$), second level ($\sigma = 0.5$), and third level ($\sigma = 0.2$).

for our method and for FOCUSS (using FOCUSS-CNDL Package[3] [14]). It is seen that our method performs two to three orders of magnitude faster than FOCUSS, while it produces approximately the same SNR. Figure 2 shows the actual source and it's estimations in different iterations for this run of the algorithm.

The experiment is then repeated 100 times for $k = 100$ and $k = 150$ (for different randomly generated sources). Average computational time in second, average SNR, standard deviation of SNR and the worst case SNR in dB are shown in table 1. Although the CPU time is not an exact measure of complexity, it can give us a rough estimation of complexity. Our simulations are performed in MATLAB 2006 environment using a 2.00GHz Dou processor with 0.99GB of memory, and under Microsoft Windows XP operating system.

5. CONCLUSIONS

In this article, a fast method for finding sparse solutions of an under-determined system of linear equations of complex variables was proposed (to be applied in complex-valued atomic decomposition and SCA). The method was based on maximizing a 'smooth' measure of sparsity. The overall algorithm was shown to be two to three orders of magnitude faster than FOCUSS, while providing approximately the same accuracy.

6. REFERENCES

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

[2] 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.

[3] P. G. Georgiev, F. J. Theis, and A. Cichocki, "Blind source separation and sparse component analysis for over-complete mixtures," in *Proceedinds of ICASSP'04*, Montreal (Canada), May 2004, pp. 493–496.

[4] Y. Li, A. Cichocki, and S. Amari, "Sparse component analysis for blind source separation with less sensors than sources," in *ICA2003*, 2003, pp. 89–94.

[5] D. L. Donoho, M. Elad, and V. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Trans. Info. Theory*, vol. 52, no. 1, pp. 6–18, Jan 2006.

[6] S. 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, 1999.

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

[8] 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.

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

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

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

[12] C. Jutten G. H. Mohimani, M. Babaie-Zadeh, "Fast sparse representation based on smoothed l0 norm," in *Proc.* ICA'07, London, UK, 2007.

[13] A. Blake and A. Zisserman, *Visual Reconstruction*, MIT Press, 1987.

[14] B. D. Rao K. Engan T. W. Lee K. Kreutz-Delgado, J. F. Murray and T. J. Sejnowski, "Dictionary learning algorithms for sparse representation," Neural Comput.

[3]For FOCUSS, we have used the MATLAB code available at `http://dsp.ucsd.edu/~jfmurray/software.htm`