Fast Block-Sparse Decomposition Based on SL0

Sina Hamidi Ghalehjegh, Massoud Babaie-Zadeh, Christian Jutten

Fast Block-Sparse Decomposition Based on SL0

Sina Hamidi Ghalehjegh[1], Massoud Babaie-Zadeh[1] *, and Christian Jutten[2]

[1] Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran
[2] GIPSA-Lab, Grenoble, and Institut Universitaire de France, France
shghamidi@gmail.com, mbzadeh@yahoo.com, Christian.Jutten@inpg.fr

Abstract. In this paper we present a new algorithm based on Smoothed ℓ^0 (SL0), called Block SL0 (BSL0), for Under-determined Systems of Linear Equations (USLE) in which the nonzero elements of the unknown vector occur in clusters. Contrary to the previous algorithms such as Block Orthogonal Matching Pursuit (BOMP) and mixed ℓ^2/ℓ^1 norm, our approach provides a fast algorithm, while providing the same (or better) accuracy. Moreover, we will see experimentally that BSL0 has better performance than SL0, BOMP and mixed ℓ^2/ℓ^1 norm when the number of nonzero elements of the source vector approaches the upper bound of uniqueness theorem.

Key words: Under-determined Systems of Linear Equations, Block-Sparsity, Sparse Decomposition, SL0 Algorithm.

1 Introduction

Sparse solutions of USLE have recently attracted a lot of attentions because of their potential applications in many different areas. They are used, for example, in compressed sensing [1, 2], under-determined Sparse Component Analysis (SCA) and source separation [3, 4] and atomic decomposition on overcomplete dictionaries [5, 6]. Generally, an USLE has infinitely many solutions but it is shown in [7] that under some conditions the sparsest solution of the system is unique.

More concretely, let $\mathbf{s}$ be an $m \times 1$ unknown vector that is observed through an $n \times m$, $n < m$, measurement matrix $\mathbf{A}$ according to $\mathbf{x} = \mathbf{A}\mathbf{s}$. Since this system is an USLE, there are infinitely many possible solutions of $\mathbf{s}$ that satisfy it for a given $\mathbf{x}$ and $\mathbf{A}$. Therefore, we must add some extra conditions to insure the uniqueness of $\mathbf{s}$. Let Spark($\mathbf{A}$) be the smallest number r such that there exists a set of r columns in $\mathbf{A}$ which are linearly dependent [7]. Moreover, define the ℓ^0 norm of $\mathbf{s}$ as its number of nonzero elements. $\mathbf{s}$ is called k-sparse if its ℓ^0 norm is k. It is shown in [7] that if k is less than $r/2$, then sparsest solution of the system is unique.

* This work has been partially funded by Iran NSF (INSF), and also by ISMO and French embassy in Tehran in the framework of a GundiShapour collaboration program.

The above uniqueness theorem of USLE has led to development of many different recovery algorithms [1,5,8,9]. Most of the algorithms find the sparsest solution by optimizing a cost function that measures the distance of $\mathbf{s}$ from being sparse. Therefore, the more precise our cost function involves the sparsity concept, the closer the recovered vector will be to the exact solution. Two successful studied algorithms are Basis Pursuit (BP) [5,9] and SL0 [8,10] which are based on minimization of ℓ^1 and smoothed-ℓ^0 norm, respectively.

In this paper, we consider recovery of an unknown solution whose nonzero elements occur in clusters. Such a signal is called block-sparse [11,12]. For this purpose, we modify the SL0 idea to use this block structure in the recovery algorithm. We will see in Sect. 4 that taking this structure into account can yield better reconstruction results.

Block-sparsity occurs when dealing with multi-band signals [13] and equalization of sparse communication channels [14]. Furthermore, it was shown in [13] that block-sparsity model can be used to treat the problem of sampling signals that lie in a union of subspaces. A block-sparse signal can be stated as follow:

$$\mathbf{s} = [\underbrace{s_{11}, \cdots, s_{1d}}_{\mathbf{s}_1}, \underbrace{s_{21}, \cdots, s_{2d}}_{\mathbf{s}_2}, \cdots, \underbrace{s_{N1}, \cdots, s_{Nd}}_{\mathbf{s}_N}]^T \quad , \tag{1}$$

where $\mathbf{s}_i$, $i = 1, \cdots, N$ is called the i^{th} block of $\mathbf{s}$ and d is the block size. A signal of dimension m which consists of N blocks of size $d = m/N$ is k sparse if at most k blocks of the signal out of N are nonzero. It is clear that if $d = 1$, block-sparsity reduces to conventional sparsity.

Using ℓ^1 relaxation [5,9] for reconstructing $\mathbf{s}$ does not exploit the fact that the signal is block-sparse, i.e., that the nonzero entries occur in consecutive positions. Therefore, different techniques were suggested in recent years. Stojnic *et. al.* in [15] modify the ℓ^1 norm cost function and call it mixed ℓ^2/ℓ^1 norm, to exploit the block-sparsity. They suggest the following optimization problem for the recovery of $\mathbf{s}$:

$$\operatorname*{argmin}_{\mathbf{s}} \|\mathbf{s}_1\|_2 + \|\mathbf{s}_2\|_2 + \cdots + \|\mathbf{s}_N\|_2 \quad \text{s.t.} \quad \mathbf{x} = \mathbf{A}\mathbf{s} \ . \tag{2}$$

In [11] it is shown that under some conditions on measurement matrix, (2) is guaranteed to recover any block-sparse signal, irrespectively of the locations of the nonzero blocks. Furthermore, recovery will be robust in the presence of noise and modeling error. [11] uses semi-definite programming to find the sparsest solution of (2). However, it is still very slow and becomes worse as the dimension increases.

Another approach presented in [16], called Block Orthogonal Matching Pursuit (BOMP), modifies standard Orthogonal Matching Pursuit (OMP) algorithm [17] to use the block structure. BOMP is fast, but is a greedy algorithm and does not provide good estimation of the sources.

Contrary to previous approaches, the method we present in this paper tries to directly work with the block version of ℓ^0 norm and is based on the idea of smoothed ℓ^0 (SL0) [8]. We will see experimentally that the proposed algorithm outperforms both mixed ℓ^2/ℓ^1 norm and BOMP methods.

The paper is organized as follows. The next section introduces the basic principles of our approach. The final algorithm is then stated in Sect. 3. Finally, Sect. 4 provides some experimental results of our algorithm and its comparison with conventional SL0, mixed ℓ^2/ℓ^1 norm and BOMP algorithms.

2 Basic Principles of Our Approach

As our proposed method is based on SL0, we first have a brief review on this algorithm. Let $\mathbf{s}$ be an unknown $m \times 1$ vector and related to $n \times 1$ measurement vector $\mathbf{x}$, $n < m$, through the measurement matrix $\mathbf{A}$ by:

$$\mathbf{x} = \mathbf{A}\mathbf{s} \ . \tag{3}$$

A recovery algorithm tries to find the sparsest solution of (3). The reconstruction can be then stated as the following optimization problem:

$$\underset{\mathbf{s}}{\operatorname{argmin}} \quad \|\mathbf{s}\|_0 \quad \text{s.t.} \quad \mathbf{x} = \mathbf{A}\mathbf{s} \ . \tag{4}$$

To solve (4) one may search for a solution with minimal ℓ^0 norm. This exhaustive search will have complexity of $O(\binom{m}{k}n^3)$ [15] and hence it is an intractable problem as the dimension increases. Recently, [8] has used a smoothed version of ℓ^0 norm, called SL0, which is experimentally shown that is a very fast algorithm and has a very good performance.

The main idea of SL0 is to approximate ℓ^0 *discontinuous* function by a suitable *continuous* one. For example, consider the following one-variable function:

$$f_\sigma(s) \triangleq e^{-s^2/2\sigma^2} \ . \tag{5}$$

It is clear that:

$$\lim_{\sigma \to 0} f_\sigma(s) = \begin{cases} 1 & s = 0 \\ 0 & s \neq 0 \end{cases} \ . \tag{6}$$

Then, by defining $F_\sigma(\mathbf{s}) \triangleq \sum_{i=1}^{m} f_\sigma(s_i)$ we can see that:

$$\|\mathbf{s}\|_0 \approx m - F_\sigma(\mathbf{s}) \ , \tag{7}$$

for small values of σ. Consequently, [8] finds the solution of (4) by maximizing $F_\sigma(\mathbf{s})$ (subject to $\mathbf{x} = \mathbf{A}\mathbf{s}$) for a very small value of σ. The value of σ determines how smooth the function F_σ is: the larger value of σ, the smoother F_σ (but worse approximation of ℓ^0 norm) and vice versa. For small values of σ, F_σ is highly non-smooth and contains a lot of local maxima, and hence its maximization is not easy. On the other hand, for larger values of σ, F_σ is smoother and contains less local maxima. Consequently, to avoid trapping in local maxima, the authors of [8] propose to use a 'decreasing' sequence for σ: for maximizing F_σ for each value of σ (using e.g. gradient algorithms), the initial value of the maximization algorithm is the maximizer of F_σ for the previous (larger) value of σ. More details about this concept and also theoretical analysis of the algorithm is stated in [8, 10]. Moreover, [8] suggests that the best initial value of $\mathbf{s}$ for the algorithm is the minimum ℓ^2 norm solution of $\mathbf{x} = \mathbf{A}\mathbf{s}$, i.e. the solution given by the pseudo-inverse of $\mathbf{A}$.

We now modify the above idea for reconstruction of *block-sparse* signals from their measurements. Let $\mathbf{s}$ be as (1) and $I(x)$ be as follow:

$$I(x) \triangleq \begin{cases} 1 & x \neq 0 \\ 0 & x = 0 \end{cases} .$$

By denoting

$$\|\mathbf{s}\|_{2,0} \triangleq \sum_{i=1}^{N} I(\|\mathbf{s}_i\|_2) \ , \tag{8}$$

a vector $\mathbf{s}$ is block k-sparse if $\|\mathbf{s}\|_{2,0} \leq k$. Therefore, the optimization problem will be:

$$\underset{\mathbf{s}}{\text{argmin}} \quad \|\mathbf{s}\|_{2,0} \quad \text{s.t.} \quad \mathbf{x} = \mathbf{A}\mathbf{s} \ . \tag{9}$$

By defining $b_i \triangleq \|\mathbf{s}_i\|_2$ we can rewrite (8) as follows:

$$\|\mathbf{s}\|_{2,0} = \sum_{i=1}^{N} I(b_i) \triangleq \|\mathbf{b}\|_0 \ , \tag{10}$$

where $\mathbf{b} \triangleq [b_1, b_2, \cdots, b_N]^T$. Using (7) it is clear that $\|\mathbf{b}\|_0 \approx N - F_\sigma(\mathbf{b})$ where $F_\sigma(\mathbf{b}) = \sum_{i=1}^{N} f_\sigma(b_i)$ and $f_\sigma(b_i) \triangleq e^{-b_i^2/2\sigma^2}$. Therefore, we can write:

$$\|\mathbf{s}\|_{2,0} \approx N - \sum_{i=1}^{N} e^{-\frac{\sum_{j=1}^{d} s_{ij}^2}{2\sigma^2}} \triangleq N - H_\sigma(\mathbf{s}) \ , \tag{11}$$

and the final optimization problem will be:

$$\underset{\mathbf{s}}{\text{argmax}} \quad H_\sigma(\mathbf{s}) \quad \text{s.t.} \quad \mathbf{x} = \mathbf{A}\mathbf{s} \ . \tag{12}$$

To solve the optimization problem (12) we use steepest ascent method. So, we need to compute gradient vector of $H_\sigma(\mathbf{s})$. Because of the block nature of $\mathbf{s}$, it is better to express the gradient vector in a block notation. So, we denote ∇H_σ as follow:

$$\nabla H_\sigma = [\underbrace{h_{11}, \cdots, h_{1d}}_{1^{st}\text{ block}}, \underbrace{h_{21}, \cdots, h_{2d}}_{2^{nd}\text{ block}}, \cdots, \underbrace{h_{N1}, \cdots, h_{Nd}}_{N^{th}\text{ block}}]^T \ . \tag{13}$$

Therefore we can write:

$$h_{uv} = \frac{\partial H_\sigma(\mathbf{s})}{\partial s_{uv}} = -\frac{s_{uv}}{\sigma^2} e^{-\frac{\sum_{j=1}^{d} s_{uj}^2}{2\sigma^2}} \ . \tag{14}$$

3 The Final Algorithms

The final algorithm is given in Fig. 1. Like SL0, we choose minimum ℓ^2 norm solution of $\mathbf{x} = \mathbf{A}\mathbf{s}$, obtained by pseudo-inverse of $\mathbf{A}$, as the initial value of the algorithm. Also, as stated in [8], to avoid trapping in local maxima we use a 'decreasing' sequence for σ. Also, steepest ascent consists of iterations of the form $\mathbf{s} \leftarrow \mathbf{s} + \mu_j \nabla H_\sigma$. As said in [8], the step-size parameters μ_j should be decreasing, i.e., for smaller values of σ, smaller values of μ_j should be applied. This is because for smaller values of σ the function H_σ is more 'fluctuating', and hence smaller step-size should be used for its maximization. It is shown in [8] that μ_j should be proportional to σ^2. Therefore, we choose $\mu_j = \mu\sigma^2$.

```
– Initialization
    1. Let $\hat{\mathbf{s}}_0$ be equal to the minimum $\ell^2$ norm solution of $\mathbf{A}\mathbf{s} = \mathbf{x}$ obtained by
       pseudo-inverse of $\mathbf{A}$.
    2. Choose a suitable decreasing sequence for $\sigma : [\sigma_1, \cdots, \sigma_J]$.
– For $j = 1, \cdots, J$:
    1. Let $\sigma = \sigma_j$
    2. Maximize (approximately) the function $F_\sigma$ on the feasible set $\{\mathbf{s}|\mathbf{A}\mathbf{s} = \mathbf{s}\}$
       using $L$ iterations of steepest ascent algorithm (followed by projection
       onto the feasible set):
        • Initialization: $\mathbf{s} = \hat{\mathbf{s}}_{j-1}$
        • For $l = 1 \cdots L$ (loop $L$ times):
            ∗ Let $\nabla H_\sigma$ as (13).
            ∗ Let $\mathbf{s} \leftarrow \mathbf{s} + (\mu\sigma^2)\nabla H_\sigma$ (where $\mu$ is a small positive constant).
            ∗ Project $\mathbf{s}$ back onto the feasible set:
                  $\mathbf{s} \leftarrow \mathbf{s} - \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}(\mathbf{A}\mathbf{s} - \mathbf{x})$ .
    3. Set $\hat{\mathbf{s}}_j = \mathbf{s}$
– Final answer is $\hat{\mathbf{s}} = \hat{\mathbf{s}}_J$
```

Fig. 1: The final BSL0 algorithm

4 Experimental Results

In this section, we discuss the performance of our proposed algorithm and compare it with conventional SL0, BOMP and mixed ℓ^2/ℓ^1 norm methods. In all of the experiments, block-sparse sources are artificially produced. A block k-sparse signal is created as follow: First, the locations of k nonzero blocks are randomly chosen. Each element in chosen blocks is 'active' and the rest elements of $\mathbf{s}$ are 'inactive'. Active elements have normal distribution of $\mathcal{N}(0, 1)$ and inactive elements have been set zero.

Each column of the mixing matrix is randomly generated using a normal distribution with zero mean and variance of 1 and then is normalized to unity.

Then the mixtures are generated using $\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{n}$ where $\mathbf{n}$ is an additive white Gaussian noise with covariance matrix $\sigma_n \mathbf{I}_n$ (where $\mathbf{I}_n$ stands for the $n \times n$ identity matrix).

To evaluate the estimation quality, Signal-to-Noise Ratio (SNR) is used. SNR (in dB) is defined as $20 \log(\|\mathbf{s}\|/\|\mathbf{s} - \hat{\mathbf{s}}\|)$, where $\mathbf{s}$ and $\hat{\mathbf{s}}$ denote the actual source and its estimation, respectively.

Experiment 1. In this experiment, we study the computational cost of the presented method, and compare it with conventional SL0, mixed ℓ^2/ℓ^1 norm and BOMP methods. The values used for this experiment are $m = 1000, n = 400, \sigma_n^2 = 0.01$ and $k \times d = 100$. We change k and d during the simulation such that their product stays constant (and equal to 100). Therefore, the possible values for (k, d) will be (100,1), (50,2), (25,4), (20,5), (10,10), (5,20), (4,25), (2,50) and (1,100). For example (k, d) =(25,4) means that $\mathbf{s}$ is a 1000×1 vector with 100 nonzero elements occurring in 25 blocks which contain 4 nonzero elements each.

We use the CPU time as a measure of complexity. Although it is not an exact measure, it gives a rough estimation of the complexity. Our simulations are performed in MATLAB 7.6 environment using an Intel Core 2 Duo 2GHz processor with 3GB of memory, and under Microsoft Windows XP operating system.

The experiment was then repeated 100 times (with the same parameters, but for different randomly generated sources and mixing matrices) and the values of SNR (in dB) and time (in seconds) obtained over these simulations were averaged. Figure 2 shows the result. The averaged CPU time (for $d = 100$) for SL0, BSL0 , ℓ^2/ℓ^1 and BOMP are 0.104, 0.111, 14.623 and 0.070 second, respectively. It is clear that BSL0 has better performance and as we increase the block size, its performance becomes better. This is because of BSL0 cost function in which we directly work with the block version of ℓ^0 norm. Also, as BSL0 is based on the fast SL0 algorithm [8], it is faster than ℓ^2/ℓ^1.

Experiment 2. As said in uniqueness theorem [7], the sparsest solution is unique when $\|\mathbf{s}\|_0 = (k \times d) < n/2 = 200$. So, we set $k \times d = 200$ and chose (k, d) pair from (200,1), (100,2), (50,4), (40,5), (25,8), (20,10), (10,20), (8,25), (5,40), (2,100) and (1,200) to examine its effect on the performance of the algorithms. The other parameters of this experiment are the same as Experiment 1. We can see from Fig. 3 that the conventional SL0 cannot recover the sparsest solution when $\mathbf{s}$ has 200 nonzero elements, but for $d > 4$, BSL0 has a performance of 25 dB or even better. The averaged CPU time (for $d = 200$) for SL0, BSL0, ℓ^2/ℓ^1 and BOMP are 0.091, 0.096, 11.879 and 0.035 second, respectively.

5 Conclusion

In this paper we studied the efficient recovery of block sparse signals using an under-determined system of equations generated from random Gaussian matrices. The motivation for considering block sparse signals is that in many applications the nonzero elements of sparse vectors tend to cluster in blocks. We

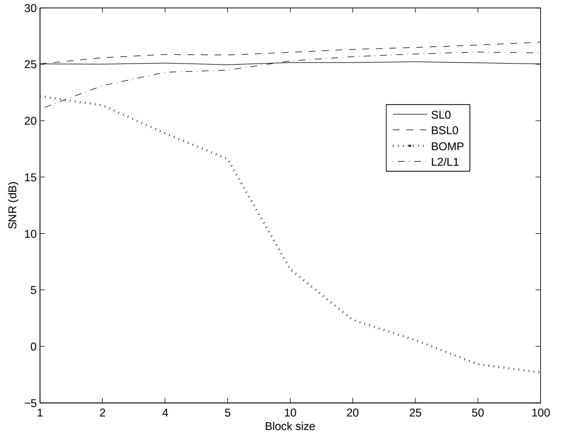

Fig. 2: Performance of BSL0, conventional SL0, BOMP and mixed ℓ^2/ℓ^1 norm as function of block size (d) for $k \times d = 100$.

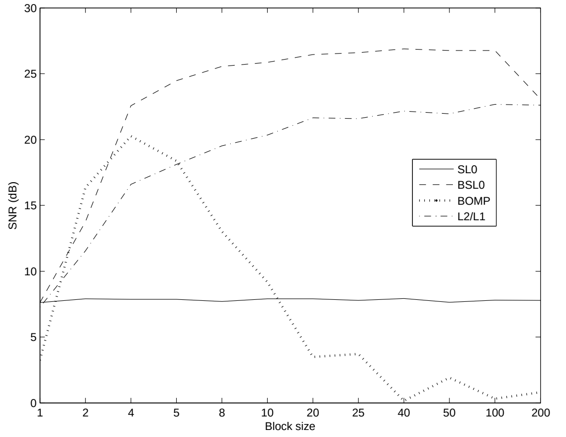

Fig. 3: Performance of BSL0, conventional SL0, BOMP and mixed ℓ^2/ℓ^1 norm as function of block size (d) for $k \times d = 200$.

showed experimentally that BSL0 is highly faster than ℓ^2/ℓ^1, while producing even better estimation accuracy. We also saw experimentally that BSL0 has better performance than SL0 when the number of nonzero elements of the source vector approaches the upper bound of uniqueness theorem.

References

1. Donoho, D.L.: Compressed sensing. IEEE Trans. Info. Theory **52**(4) (April 2006) 1289–1306
2. Baraniuk, R.G.: Compressive sensing. IEEE Signal Processing Magazine **24**(4) (July 2007) 118–124
3. Gribonval, R., Lesage, S.: A survey of sparse component analysis for blind source separation: principles, perspectives, ans new challenges, ESANN'06 (April 2006) 323–330
4. Li, Y., A.Cichocki, Amari, S.: Sparse component analysis for blind source separation with less sensors than sources. ICA2003 (2003) 89–94
5. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM Journal on Scientific Computing **20**(1) (1999) 33–61
6. Donoho, D.L., Elad, M., Temlyakov, V.: Stable recovery of sparse overcomplete representations in the presence of noise. IEEE Trans. Info. Theory **52**(1) (January 2006) 6–18
7. Donoho, D.L., Elad, M.: Optimally sparse representation in general (nonorthogonal) dictionaries via l_1 minimization. Proc. Natl. Acad. Sci **100**(5) (March 2003) 2197–2202
8. Mohimani, H., Babaie-Zadeh, M., Jutten, C.: A fast approach for overcomplete sparse decomposition based on smoothed ℓ^0 norm. IEEE Trans. Signal Processing **57** (January 2009) 289–301
9. Candès, E.J., Romberg, J.K., T.Tao: Robust uncertainty principles: Exact signal representation from highly incomplete frequency information. IEEE Trans. Info. Theory **52**(2) (February 2006) 489–509
10. Mohimani, H., Babaie-Zadeh, M., Gorodnitsky, I., Jutten, C.: Sparse recovery using smoothed ℓ^0 (sl0): Convergence analysis. arXiv:cs.IT/1001.5073
11. Eldar, Y.C., Mishali, M.: Robust recovery of signals from a structured union of subspaces. IEEE Trans. Info. Theory **55** (2009) 5302–5316
12. Eldar, Y.C., Kuppinger, P., Bölcskei, H.: Compressed sensing of block-sparse signals: Uncertainty relations and efficient recovery. arXiv:cs.IT/0906.3173
13. Mishali, M., Eldar, Y.C.: Blind multi-band signal reconstruction: Compressed sensing for analog signals. IEEE Trans. Signal Processing **57**(3) (March 2009) 993–1009
14. Cotter, S., Rao, B.: Sparse channel estimation via matching pursuit with application to equalization. IEEE Trans. on Comm. (March 2002)
15. Stojnic, M., Parversh, F., B.Hassibi: On the reconstruction of block-sparse signals with an optional number of measurements. IEEE Trans. Sig. Proc. (2009)
16. Majumdar, A., Ward, R.: Fast group sparse classification. PACRIM (2009) 11–16
17. Mallat, S., Zhand, Z.: Matching pursuits with time-frequency dictionaries. IEEE Trans. on Signal proc. **41**(12) (1993) 3397–3415