Heuristics For Efficient Sparse Blind Source Separation

C Kervazo¹, J Bobin¹ and C Chenot²
¹ CEA Saclay, Gif-sur-Yvette, 91191 cedex, France
² Institute for Digital Communications, University of Edinburgh, UK
E-mail: christophe.kervazo@cea.fr

Abstract. Sparse Blind Source Separation (sparse BSS) is a key method to analyze multichannel data in fields ranging from medical imaging to astrophysics. However, since it relies on seeking the solution of a non-convex penalized matrix factorization problem, its performances largely depend on the optimization strategy. In this context, Proximal Alternating Linearized Minimization (PALM) has become a standard algorithm which, despite its theoretical grounding, generally provides poor practical separation results. In this work, we first investigate the origins of these limitations, which are shown to take their roots in the sensitivity to both the initialization and the regularization parameter choice. As an alternative, we propose a novel strategy that combines a heuristic approach with PALM. We show its relevance on realistic astrophysical data.

1. Introduction

1.1. Blind Source Separation problem

In the BSS [1] framework, the data are composed of m observations, each of which has t samples. These observations are supposed to be some linear combinations of n sources. The objective of BSS is to retrieve the sources as well as the mixing coefficients. In matrix form, the goal is therefore to find two matrices $S$ (of size $n \times t$) and $A$ (of size $m \times n$), called respectively the source and the mixing matrices, such that: $X = AS + N$, where $X$ (of size $m \times t$) is the observation matrix that is corrupted with some unknown noise $N$. Since it requires tackling an ill-posed unsupervised matrix factorization problem, further assumptions are needed, including the statistical independance of the sources (ICA - [1]), the non-negativity of $A$ and $S$ [2]. In this work, we will focus on the sparsity of the sources [3–6]. In this framework, sparse BSS will aim at finding a (local) minimum of:

$$\min_{A,S} \frac{1}{2} \|X - AS\|^2_F + \epsilon_{\{\|A_{i}\|_2 \leq 1\}}(A) + \|R_{S} \odot (S\Phi_{S}^T)\|_1$$

(1)

The first data fidelity term promotes a faithful reconstruction of the data. The use of the Frobenius norm $\|\cdot\|_F$ stems from the assumption of a Gaussian noise $N$. The second term corresponds to the oblique constraint ensuring that the columns of $A$ are all in the $\ell_2$ ball. This avoids degenerated $A$ and $S$ matrices where $\|A\| \to \infty$ and $\|S\| \to 0$. The last term involving the Hadamard product $\odot$ enforces an $\ell_1$ sparsity constraint in a transformed domain $\Phi_S$. In the following, $\Phi_S$ will be taken equal to either the identity (in which case the sparsity is enforced in
the direct domain) or the starlet transform. $R_S$ controls the trade-off between the data fidelity and the sparsity terms. It can be decomposed into $R_S = \Lambda_S \odot W$ where $\Lambda_S$ is a diagonal matrix of the regularization parameters $\lambda_1, \lambda_2, \ldots, \lambda_n$ and $W$ is a matrix used to introduce individual penalization coefficients in the context of reweighted $\ell_1$ [7] (when no reweighting is used, $W$ is equal to the identity matrix).

1.2. Sparse BSS in practice

Since sparse BSS requires solving a penalized matrix factorization problem, it is important to highlight that the separation quality strongly depends on the optimization strategy. For that purpose, different algorithmic frameworks have been used so far: projected Alternate Least-Square (ALS - [2]), Block-Coordinate Descent (BCD - [8]) and PALM [9]. However, any practitioner can draw the same conclusion: the solution of sparse BSS methods is highly sensitive to the initial point and the values of the regularization parameters, which are generally tricky to tune without any first guess of the solution.

- **Initialization**: as problem (1) is not convex but multi-convex, the algorithm performing its minimization can be trapped in spurious critical points, depending on the initial matrices $A$ and $S$. Different optimization strategies can be affected differently.

- **Regularization parameters**: the practical choice of the regularization parameters $\Lambda_S$ is of paramount significance:
  - The parameters are directly impacting the shape of the solution through the trade-off between the sparsity level and fidelity to the data.
  - The problem being non-convex, a change in the parameters can also bring the optimization algorithm to stabilize in the neighborhood of a different critical point, potentially leading to a very different solution.

1.3. Contributions

While PALM is theoretically well rooted and yields rather fast minimization schemes (in contrast to BCD), it generally provides poor separation results. We first investigate the origins of this poor practical efficiency, and especially the sensitivity to both initialization and regularization parameter settings. We further show how PALM-based implementations can benefit from the information provided by heuristic approaches which are in practice more robust. The robustness of the proposed combined strategy is demonstrated on realistic astrophysical data.

2. Optimization strategy: PALM and GMCA

As emphasized before, the optimization strategy is decisive to avoid stationary points. In the following, we will discuss PALM [9] and projected ALS such as Generalized Morphological Component Analysis (GMCA - [3]). BCD will not be studied since it requires an exact minimization at each iteration, leading in general to a high computational cost.

2.1. PALM algorithm

PALM [9] attractiveness partially comes from the fact that it is proved to converge to a local minimum of (1) under mild conditions [9]. PALM is an iterative algorithm. At each iteration, it alternates between a proximal gradient step on $A$ and $S$. In our case, it can be shown that the proximal operator corresponding to the $\ell_1$ sparsity term in (1) is the soft-thresholding operator $S_{\lambda_S}(\cdot)$ applied in the transform domain. Furthermore, the operator corresponding to the oblique constraint is the projection of each column of $A$ on the $\ell_2$ unit ball, which we shall denote as $\Pi_{\|\cdot\|_2}(\cdot)$. Each iteration $(k)$ can then be read as:
2.2. GMCA algorithm
GMCA algorithm is based on projected ALS. At each iteration \((k)\), the gradient step of (2) and (4) is replaced by a multiplication by a pseudo-inverse: \(\hat{S} = A^{(k-1)\dagger}X\) and \(\hat{A} = XS^{(k)\dagger}\). Compared to PALM, GMCA cannot be proved to converge (while stabilizing in most of practical cases after some iterations). Furthermore, even when stabilizing the output of GMCA is not exactly minimizing (1) but rather an approximation of the cost function. However, thanks to heuristics, it benefits from an automatic thresholding strategy (see Sec.4.1.1) which has been empirically shown to improve its robustness with respect to the initialization.

3. PALM-based sparse BSS in practice
The objective of this section is to empirically shed light on the lack of robustness of PALM to motivate the need of finding a relevant strategy to enable its use on practical problems.

3.1. Experimental protocol
To bring out PALM sensibility and the mechanisms at stake, our experiments focus in this subsection on simple data coming from \(n = 2\) equilibrated (with equal \(\ell_2\) norm) sources. The sources are assumed to be exactly sparse in the direct domain, a proportion \(\ell\) of samples being non-zeros and drawn according to a standard normal distribution. The mixing is performed through a \(A\) matrix drawn randomly following a standard normal distribution and modified to have unit columns. Its condition number is \(C_d = 10\). There is \(m = n\) observations. To complete the creation of the \(X\) data, a Gaussian additive noise is added to the mixing, such that the Signal to Noise Ratio is \(\text{SNR} = 60\,\text{dB}\).

Once the algorithm launched, it stops based on a convergence criterion computed through the angular distance of the columns of \(A\) between two successive iterations. The separation quality is then measured using a global criterion on \(A\) [10]:

\[
C_A = -10\log_{10}(\text{median}([PA^\dagger A^* - I_d]))
\]

with \(A^*\) the true mixing matrix and \(PA^\dagger\) the pseudo-inverse of the solution found by the algorithm corrected through \(P\) for the scale and permutation indeterminacies. The higher \(C_A\), the better the separation.

To study PALM robustness, we perform an exhaustive search for the parameters. This would be intractable on practical cases with a higher number of sources and unknown true \(A^*\) and \(S^*\). In this section, the parameters are held fixed during the whole algorithm, the goal being to show the sensibility to the parameter choice. The results might however be sub-optimal due to the fact that varying parameters could define a better trajectory avoiding spurious critical points.

3.2. Sensibility to the thresholding strategy
Following our experimental protocol, we obtain Figure 1, on which the mean of \(C_A\) is displayed as a function of the threshold of each of the 2 sources. The lack of robustness and sensibility to the parameter choice is underlined by the following remarks:

\[
1 - \text{Update of } S \text{ using a fixed } A:
\]

\[
\hat{S} = S^{(k-1)} - \frac{\gamma}{\|A^{(k-1)\dagger}A^{(k-1)}\|_2} A^{(k-1)\dagger}(A^{(k-1)}S^{(k-1)} - X)
\]

(2)

\[
S^{(k)} = S \frac{\gamma \|S\|_2}{\|S^{(k)}S^{(k)\dagger} - A^{(k-1)\dagger}(A^{(k-1)}S^{(k)} - X)S^{(k)\dagger}\|_2} (\Phi S)^\dagger \Phi S
\]

(3)

\[
2 - \text{Update of } A \text{ using a fixed } S:
\]

\[
\hat{A} = A^{(k-1)} - \frac{\delta}{\|S^{(k)}S^{(k)\dagger} - A^{(k-1)\dagger}(A^{(k-1)}S^{(k)} - X)S^{(k)\dagger}\|_2} (A^{(k-1)}S^{(k)} - X)S^{(k)\dagger}
\]

(4)

\[
A^{(k)} = \Pi_{\|\cdot\|_2} (\hat{A})
\]

(5)
Figure 1. Left: Mean of $C_A$ as a 2D function of the thresholds for 5 different initializations of PALM algorithm. Right: Standard deviation of $C_A$ as a 2D function of the thresholds for 5 different initializations of PALM algorithm.

- High quality separation happens for values close to the diagonal. A slight deviation from it can cause a 30 dB drop for a $3 \times 10^{-4}$ modification of one parameter and lead to a very bad separation. One could argue that this problem could be circumvented by imposing $\lambda_1 = \lambda_2$. First, it has to be emphasized that the maximum is \textit{not} on the diagonal but slightly outside, for $\lambda_1 = 3.81 \times 10^{-3}$ and $\lambda_2 = 3.88 \times 10^{-3}$. Merely setting $\lambda_1 = \lambda_2 = 3.81 \times 10^{-3}$ yields a loss of more than 10 dB. Second, in this simple toy example, the sources are equilibrated. It would not necessarily be the case with real-life sources, which could lead to a change in the diagonal structure of the figure;

- Inside of the diagonal itself, a $2 \times 10^{-3}$ change of $\lambda_1 = \lambda_2$ can lead to a loss of about 7 dB.

3.3. Sensibility to the initialization
As described in section 1.2, the initialization impacts the results due to the non-convex nature of problem (1) because the algorithm can stay trapped in spurious critical points. The standard deviation of $C_A$ over different initializations of $A$ is plotted in Fig. 1 to quantify this impact. It can reach up to 5 dB for some regularization parameters choices. The best parameter setting corresponds to a quite high standard deviation and can shift depending on the initialization. It makes the practical choice of an initialization a critical point which directly impact the quality of the local minimum found by the algorithm. It further highlights the sensitivity to the values of the regularization parameters. A key question is thus how to tune PALM in practice and if it can be done in an automatic way without prior knowledge about the solution.

4. Complexity of introducing heuristics in PALM
4.1. Heuristic motivation and description
Building on the automatic thresholding strategy of GMCA, the goal of this part is to try to derive one for PALM.

4.1.1. Automatic parameter choice in GMCA In GMCA, the threshold choice is performed computing the Median Absolute Deviation (MAD) for each source and setting the corresponding threshold to a multiple $k$ of this value:

$$(\lambda_1, \lambda_2, ..., \lambda_n)^T = k \times \text{MAD}(\hat{S})$$ (6)
In this equation and in the remaining of this section, $\Phi_T^S$ was supposed to be the identity matrix without loss of generality. $k$ is a positive number and the MAD operator is computed row-by-row with $\text{MAD}(z) = \text{median}(|z - \text{median}(z)|)$ for $z \in \mathbb{R}^t$. If we suppose that GMCA has found the true $A$ matrix denoted by $A^*$, it can be shown that:

$$k \times \text{MAD}(\hat{S}) \simeq k \times \text{MAD}(A^* N)$$

(7)

The true thresholds should therefore remove a dense signal coming from the back-projection of the data noise $N$. This interpretation of the thresholding is however limited to the case when the estimation of $A$ is close to $A^*$. In the opposite case, the unperfect demixing results in an increase of the thresholds due to an extra noise coming from interferences between the sources.

4.1.2. Heuristic in PALM

Lets assume that the thresholds in PALM are computed the same way as in GMCA through eq. (6), and that the algorithm has converged to both the true matrices $A^*$ and $S^*$. Then, due to the assumption of $S^*$ sparsity:

$$(\lambda_1, \lambda_2, ..., \lambda_n)^T \simeq k \times \frac{\gamma}{\|A^{*T}A^*\|_2} \text{MAD}(A^* N)$$

(8)

Therefore, using the MAD enables a thresholding of a projection of the noise $N$, which yields a similar interpretation as in GMCA. This observation must however be tempered:

- It only holds when and if PALM has converged towards good $A$ and $S$;
- The projection is not performed in the same space as GMCA, for which the noise is back-projected in the source space. Both spaces however merge when $A$ is orthogonal.

Finally, it has to be noted that the thresholds become fixed after a given number of iterations to ensure the convergence.

4.2. Experimental protocol

The mixing and the metric on the separation quality are the same as in sec. 3.1. The automatic parameter setting described in the previous subsection is used inside of a PALM algorithm for different $k$ values.

4.3. Empirical performances of the heuristic

The results are displayed in Fig 2. The low values of $C_A$, reaching at most 6dB, are to be compared with the 39 dB obtained during the exhaustive parameter search in 3.1. While efficient in GMCA, the MAD heuristic leads to a bad separation quality due to the different space in which the thresholding is performed (cf. 4.1.2), yielding a different role to the thresholds. To better understand this role, lets assume that the algorithm is initialized with the true mixing matrix $A^*$ only (and not with $S^*$, contrary to what is described in 4.1.2). In PALM, the use of the transpose $A^{*T}$ in eq. (2) leads towards a re-mixing of the sources at each iteration whose goal is to gradually decrease the data fidelity term. This re-mixing however increases the sparsity term and should be impeded by the thresholds, giving to them a demixing role.

On the opposite, when using GMCA, such a re-mixing between the sources does not exist due to the use of the pseudo-inverse $A^{*T}$ instead of $A^*T$, which alleviate somehow the issue of finding relevant thresholds. Therefore, using the MAD heuristic in PALM does not results in good separations in practice, because the interpretation in terms of noise removal is less relevant than in GMCA.
5. Combining GMCA and PALM: a hybrid strategy

In this part, we propose to combine the best of GMCA and PALM in a two step approach. The algorithm comprehends a warm-up stage, in which GMCA is performed, followed by a refinement stage during which PALM is performed retaining as much information as possible coming from the warm-up stage.

5.1. Motivation and full description of the algorithm

Our approach is motivated by several remarks:

- **PALM theoretical background**: in particular, the 2-step algorithm is thus proved to converge.
- **GMCA robustness with regards to initialization**, which will lead to a robust 2-step algorithm.
- **Benefit from GMCA solution**:
  - Since both $A_{GMCA}$ and $S_{GMCA}$ are close to $A^*$ and $S^*$, they can be used to derive the thresholds using the MAD (after a given number of iterations the thresholds become fixed to ensure convergence).
  - $S_{GMCA}$ should already give a good approximation of the most prominent peaks. This information can be exploited in the refinement stage through the introduction of reweighted L1 [7] using the reweighting matrix $W$ in problem (1).

These remarks lead to the following 2-step algorithm:

**Input**: $X$ (data matrix)

- Random initialization $A_0$ and $S_0$

  **Warm-up stage**:

  $A_{GMCA}, S_{GMCA} = GMCA(X,A_0,S_0)$

  **Refinement stage**:

  $A_{PALM}, S_{PALM} = PALM(X,A_{GMCA},S_{GMCA})$

The initialization, thresholding strategy and reweighting information come from the warm-up stage.
Table 1. Average $C_A$ (10 different initializations) for 3 SNR values and 5 algorithms.

|          | 10 dB | 15 dB | 20 dB |
|----------|-------|-------|-------|
| 2 step   | 11.57 | 16.92 | 22.09 |
| PALM     | 9.38  | 10.94 | 11.01 |
| GMCA     | 8.87  | 12.15 | 15.87 |
| EFICA    | −6.92 | 5.11  | 9.41  |
| RNA      | −6.68 | −5.49 | 6.27  |

5.2. Empirical results
The experimental protocol is the same as described in sec. 4.2 except that the 2-step algorithm is used instead of PALM. The results are plotted in Fig. 2. With values of $C_A$ higher than 33 dB, the demixing is very accurate. Compared to PALM only, the variance of results over different initializations is also much decreased, becoming almost zero, which shows the robustness of the algorithm with regards to the initialization.

6. Experiment on real data
6.1. Data description and experimental protocol
The goal of this part is to apply our algorithm to realistic data to show its efficiency. The $n = 2$ sources come from simulations obtained from real data of Cassiopeia A supernova remnant. As displayed in Fig. 3, they each consists in a 2D image of resolution $t = 128 \times 128$ pixels, supposed to be approximately sparse in the starlet domain. The mixing matrix is simulated as described in 3.1. To increase the realism of the data and further test the algorithm, we tried three relatively low SNR values: 10, 15 and 20 dB. $k$ is set to 3, which would correspond to a classical hypothesis in terms of Gaussian noise removal.

6.2. Empirical results
The mean of $C_A$ values is displayed in Table 1 and the estimation for a SNR of 15 dB is shown in Fig. 3. The 2-step approach always achieve better results than the two classical BSS algorithms with which we performed the comparison, namely Relative Newton Algorithm (RNA) and Efficient FastICA (EFICA). It also outperforms both GMCA and PALM, being always better by at least 2 dB than the best of them.
In addition to the results displayed in Fig. (1), the standard deviation of $C_A$ over different initializations is almost 0, which shows the robustness of the algorithm.

Conclusion
In this work, we show that the ability of recent and theoretically grounded optimization strategies like PALM to provide solutions to sparse BSS problems is highly sensitive to both the initialization and the values of the regularization parameters. To further design an effective as well as robust algorithm, we introduce a 2-step strategy combining PALM with robust heuristic methods such as GMCA. Beyond improving the robustness of PALM-based implementations with respect to initialization, the regularization parameters can be automatically set in the proposed approach. Numerical experiments on both simulated and realistic data demonstrate a high separation quality and good robustness on mixings with low SNR.

Acknowledgment
This work is supported by the European Community through the grant LENA (ERC StG - contract no. 678282).
References

[1] Comon P and Jutten C 2010 *Handbook of Blind Source Separation: Independent component analysis and applications* (Academic Press)

[2] Gillis N and Glineur F 2012 *Neural Computation* **24** 1085–1105

[3] Bobin J, Starck J L, Moudden Y and Fadili M J 2008 *Advances in Imaging and Electron Physics* **152** 221–302

[4] Zibulevsky M and Pearlmutter B A 2001 *Neural Computation* **13** 863–882

[5] Bronstein A M, Bronstein M M, Zibulevsky M and Zeevi Y Y 2005 *International Journal of Imaging Systems and Technology* **15** 84–91

[6] Li Y, Amari S I, Cichocki A, Ho D W and Xie S 2006 *IEEE Transactions on Signal Processing* **54** 423–437

[7] Candes E J, Wakin M B and Boyd S P 2008 *Journal of Fourier analysis and applications* **14** 877–905

[8] Tseng P 2001 *Journal of Optimization Theory and Applications* **109** 475–494

[9] Bolte J, Sabach S and Teboulle M 2014 *Mathematical Programming* **146** 459–494

[10] Bobin J, Rapin J, Larue A and Starck J L 2015 *IEEE Transactions on Signal Processing* **63** 1199–1213