Robust Non-Negative Matrix Factorization for Multispectral Data with Sparse Prior

Jérémy Rapin[1,2], Jérôme Bobin[2], Anthony Larue[1], and Jean-Luc Starck[2]

[1]CEA, LIST, Laboratoire d'Outils pour l'Analyse de Données, 91191 Gif-sur-Yvette CEDEX,France
[2]CEA, IRFU, Service d'Astrophysique, 91191 Gif-sur-Yvette CEDEX,France

June 29, 2012

Abstract

In this work, we study Non-Negative Matrix Factorization (NMF) and compare standard algorithms with an extension to NMF of a Blind Source Separation algorithm using sparsity, Generalized Morphological Component Analysis (GMCA). We also develop a more robust version of GMCA handling more precisely the *priors* through sub-iterations, which we call rGMCA. We present preliminary results showing GMCA is well suited to solve this kind of problem and in particular that the decreasing threshold it uses is helpful to disambiguate the sources. We also show that rGMCA is more robust to correlation between the sources.

Keywords: Blind Source Separation, Robust Generalized Morphological Component Analysis, Non-Negative Matrix Factorization, Sparsity

1 Introduction to Non-Negative Matrix Factorization

Recently the rapid development of multi-wavelength sensors in astrophysics has increased the need for dedicated efficient data analysis tools. Such kind of data are generally made of a collection of observations or images of the same physical phenomena in different wavelength bands. It is customary to assume that these observations are mixtures of elementary physical components which do not share the same spectrum. In the field of astrophysics, examples of these components include point sources observed by the Fermi space telescope.

In this setting, a classically used data analysis technique to extract the different physical components of the data is Blind Source Separation. The model is that the data is a linear mixture of these sources, in other words:

$$Y = AS + N$$

Y being the data, S the sources, A the mixing/weight matrix and N the noise. We therefore want to reconstruct both A and S using the data Y.

Depending on the data, several properties can be used to disambiguate the sources, such as the positivity of both A (if it represents concentrations for instance) and S (which can represent an intensity) which lead to Non-negative Matrix Factorization (NMF) as we will see in this section. Other *priors* can also be added such as the sparsity in Generalized Morphological Component Analysis, which we adapt to NMF in section (2.2). We also propose in section (2.3) a more robust version of this algorithm which properly takes into account the positivity constraints. Finally, we compare and analyze the different algorithms in section (3).

1.1 Formulation

The NMF problem can be seen as a geometrical problem, such as in [1], or a statistical problem related to independent component analysis as in [2] for instance, but we will here focus on the optimization formulation.

We will use the following notations:

- n the number of samples of the signals.
- m the number of observations.
- r the number of sources. Typically, $r < n$ and $r \leq m$ and r is given.
- $Y \in \mathcal{M}_{m,n}(\mathbb{R})$ the data matrix.
- $A \in \mathcal{M}_{m,r}(\mathbb{R})$ the observation/mixing matrix, which is unknown.
- $S \in \mathcal{M}_{r,n}(\mathbb{R})$ the source matrix, which is unknown.

Whitout noise, the model is $Y = AS$ so that we want our reconstruction AS to be as close as possible to the data Y for a given distance or divergence $\mathcal{D}$, which yields the minimization problem (1).

$$\left\{ \begin{array}{l} \underset{A,S}{\text{argmin}}\ \mathcal{D}(Y||AS) \\ \forall(i,j)\ A_{ij} \geq 0,\ S_{ij} \geq 0 \end{array} \right. \tag{1}$$

A usual choice for the divergence is $\mathcal{D}(Y||AS) = \frac{1}{2}||Y - AS||_2^2$, which means we want Y to be as close to AS in L_2 norm. This corresponds to maximizing the likelihood when the data is corrupted by Gaussian noise. Other choices include for instance Kullback-Leibler divergence [3] among others [4] in order to include different noise *priors*. In this work we will focus on the quadratic difference such as it is written above.

There are 2 indeterminacies which are essential to notice:

- *scale*: with Δ a diagonal matrix of size $r \times r$ with strictly positive coefficients, we have $AS = (A\Delta)(\Delta^{-1}S)$. It is therefore impossible to recover the exact scaling which are shared between A and S. A normalization is usually applied on A to prevent this scaling issue to interfere with the convergence.
- *permutations*: $AS = \sum_i a_i s^i = \sum_i a_{\pi(i)} s^{\pi(i)}$ (with a_i a column of A and s^i a line of S) for any permutation π so that even without considering the scaling issue, we can only recover A and S up to a permutation.

Finally and most importantly, because of the product between A and S, the NMF problem is not convex and can present several local minima which makes the problem extremely difficult to solve. However the sub-problem in A, which reads for instance

$$\underset{A \geq 0}{\text{argmin}}\ \frac{1}{2}||Y - AS||_2^2$$

and the sub-problem in S are both convex. Most algorithms rely on this fact, minimizing alternatively in A and S, in order to converge to a minimum which may however not be global. Still, we can hope to find a unique solution for the whole NMF problem under some conditions [5] (up to the indeterminacies).

1.2 Standard Algorithms

One of the first algorithms aiming at solving NMF is the multiplicative update rule [6]. At each iteration we alternatingly update A and S with a point-wise multiplication between the current value and a well-chosen positive matrix, hence keeping the positivity property. This positive matrix is chosen so that the cost function does not increase.

$$\left\{ \begin{array}{l} A \leftarrow A \odot (YS^T) \oslash (ASS^T) \\ S \leftarrow S \odot (A^T Y) \oslash (A^T AS) \end{array} \right. \tag{2}$$

with $\odot$ the point-wise multiplication and $\oslash$ the point-wise division. Because of this division, it can be necessary to add a small regularization to the denominator in order to avoid dividing by 0.

This multiplicative update rule is widely used in practice and is usually considered as a standard because of its convenience, with no parameter to set, and for being among the first algorithms in the field [3] [6]. However, it has been shown to be slow and does not yield very good results. Also, the algorithm may not even converge to a local minimum as it is observed experimentally in [7] and well explained in [8].

In the case of the least square cost function it is however easy to solve the exact unconstrained problem, which can then be projected on the positivity constraint [9]. Indeed, considering A for instance (as long as SS^T is invertible):

$$\frac{\partial \mathcal{D}(Y||AS)}{\partial A} = 0 \Longleftrightarrow A = YS^T(SS^T)^{-1}$$

With the notation $[x]_+$ standing for $max(x, 0)$, the update rule for this algorithm, usually called Alternating Least Squares (ALS), is:

$$\left\{ \begin{array}{l} A \leftarrow \left[YS^T(SS^T)^{-1}\right]_+ \\ S \leftarrow \left[(A^T A)^{-1} A^T Y\right]_+ \end{array} \right. \tag{3}$$

This method is also widely used since it is extremely simple to implement and efficient in decreasing the cost function. However, as we mentioned, it is only convenient for the least square cost function. Also, as there is no convergence property, we will show in section (3.6) that ALS can present some instabilities when getting close to the solution.

2 NMF and Sparsity

2.1 BSS and Sparsity

In BSS, the aim is to extract several source signals from observations in which all the sources can be mixed. In order to differentiate the sources, it is essential that they present some kind of diversity. Typically, the independence of the sources is often used, yielding Independent Component Analysis (ICA).

However, recently, a new type of diversity based on sparse representations of signals has drawn the attention of the research community [10]. The idea is that in some well-chosen domains, it can be easier to distinguish the sources. In the cocktail party problem for instance, several people are talking at the same time, but they can be distinguishable in the time-frequency domain since they have different voice pitches. These "nice" domains are the ones which tend to concentrate the information (make the signal sparse) and therefore if the sources are not too correlated, these concentrations will appear at locations different enough to disambiguate the sources. This can be modelled by using a sharp *prior* on the distribution of the values [10] or by imposing that the signals contain few active coefficients such as in Generalized Morphological Component Analysis (GMCA) [11].

In comparison to ICA, sparsity based methods have been able to better take into consideration correlation between the sources. They can be fast to compute and robust to local minima. Finally, as noise is typically not sparse it is much easier to differentiate it from the sources in these sparse domain and therefore the sparsity based algorithms are more robust to noise.

2.2 GMCA for NMF

Considering the contributions of sparsity to BSS, it seems natural to try including sparse *priors* as well in NMF problems. This has already been done with some success by extending NMF algorithms to sparsity [12, 13, 14]. We propose here to extend a sparse BSS algorithm, GMCA, to NMF. To this end, we need to recast the problem under another form. Considering the least square data fidelity term and adding a sparse *prior* $\lambda||S||_1$ to the cost function in order to favor solutions with many zeros in S (which is the case for a spectrum for instance), we obtain the following problem:

$$\begin{cases} \underset{A,S}{\operatorname{argmin}} \ \frac{1}{2}||Y - AS||_2^2 + \lambda||S||_1 \\ \forall(i,j)\ A_{ij} \geq 0,\ S_{ij} \geq 0 \end{cases} \tag{4}$$

However, if we define the characteristic function of the positive orthant $\mathbb{I}_{\mathbb{R}^+_{mr}}$:

$$\mathbb{I}_{\mathbb{R}^+_{mr}} : M_{mr}(\mathbb{R}) \to \bar{\mathbb{R}} \tag{5}$$

$$X \to \begin{cases} 0 \text{ if } \forall(i,j)\ X_{ij} \geq 0 \\ +\infty \text{ otherwise} \end{cases}$$

we can recast the problem (4) under the unconstrained form:

$$(4) \iff \underset{A,S}{\operatorname{argmin}} \ \frac{1}{2}||Y - AS||_2^2 + \lambda||S||_1 + \mathbb{I}_{\mathbb{R}^+_{mr}}(A) + \mathbb{I}_{\mathbb{R}^+_{rn}}(S) \tag{6}$$

Let us now focus only on the update of S and consider therefore that A is fixed. Notice that the cost function is convex, as a sum of convex functions. It can actually be split into a convex and derivable function$\frac{1}{2}||Y - AS||_2^2$ and a convex non-derivable function $\lambda||S||_1 + \mathbb{I}_{\mathbb{R}^+_{rn}}(S)$ which admits an explicit proximal operator:

$$\operatorname{prox}_{\mathbb{I}_{\mathbb{R}^+_{rn}} + \lambda||.||_1}(X) = [X - \lambda \mathbb{1}_{rn}]_+ = \mathrm{ST}^+_\lambda(X)$$

with $\mathbb{1}_{rn}$ the matrix of size $r \times n$ and full of ones, and ST standing for "soft-thresholding". We will also use HT for "hard-thresholding".

The idea of GMCA is to solve exactly the least square data fidelity term and then project on the non-derivable *priors* and constraints using the proximal operator. The solution of the problem is a fixed point of this process.

As the L_1 penalization induces a bias, the authors prefer to use hard-thresholding (setting to zero all coefficients smaller than the threshold in absolute value) instead of soft-thresholding in the proximal step, approaching an L_0 regularization. While this is not rigorous, it yields good results in practice.

One important feature of this algorithm is the use of a decreasing thresholding parameter. In other words, the problem evolves from a large penalization parameter λ to smaller and smaller one, much like the regularization parameter in [15] though it is not applied in the same way. This strategy has a number of advantages:

- identifying the mixing directions using the largest coefficients and refining them afterwards.
- converging faster [16] especially when using sub-iterations.

- converging to the global minima as it is the only one which remains stable for a varying thresholding parameter. [11]
- denoising when necessary, as thresholding is often used as a non-linear denoiser.

The algorithm such as we adapted it and used it is given in Algorithm 1. We will explain in a subsequent section how to choose the evolution of the threshold throughout the iterations.

```
**Input**: $Y$ (measures)
**Output**: $A$ and $S$
Initialize $A_0$, $S_0$
$n = 0$
**while** *not converged* **do**
    $n = n + 1$
    Choose $\lambda_n$
    $S_n = \mathrm{HT}^{+}_{\lambda_n}((A_{n-1}^T A_{n-1})^{-1} A_{n-1}^T Y)$
    $A_n = [Y S_n^T (S_n S_n^T)^{-1}]_+$
$A = A_N$
$S = S_N$
```

Algorithm 1: GMCA for NMF

Finally, as a way to avoid the scaling issue, after both the updates the scale is re-spread equally between S and of A. This leads a column of A and its corresponding row of S to have the same L_2 norm, since there does not seem to be any reason to privilege either S or A in this setting.

2.3 Robust GMCA for NMF

While GMCA is extremely fast, it does not solve the exact constrained sub-problem at each iteration and it is limited to the case of the least square data fidelity term. We propose here to overcome this issue by using an iterative algorithm able to solve the constrained sub-problems. This has already been partly done by Lin [17] but a very appropriate framework to incorporate both positivity and sparse *priors* is proximal splitting and in particular the Forward-Backward algorithm [18] also reviewed in in [19], and which can be accelerated such as in [20] and FISTA [21] which is the algorithm we use in practice.

With the same splitting as in section 2.2, we can directly use FISTA to compute the solution of the constrained problem:

$$\underset{S \geq 0}{\operatorname{argmin}} \; \frac{1}{2}||Y - AS||_2^2 + \lambda||S||_1$$

The pseudo-code of our algorithm called robust GMCA can be seen in Algorithm 2, where the sub-problems are solved using FISTA [21].

Our approach differs from [17] by using the decreasing L_1 regularization from GMCA and the acceleration from [20] while using the largest step size allowed in the theory. It also differs from [22] who did not use the L_1 decreasing regularization either and who uses the acceleration but does not sub-iterate hence losing the theoretical warranty of fast convergence.

Finally, just as with GMCA, instead of solving the problems using soft-thresholding, we can use hard-thresholding to approach an L_0 penalization. In order to differenciate both versions, we will call (H)rGMCA the version using hard-thresholding and (S)rGMCA the version using soft-thresholding.

```
Input: $Y$ (measures)
Output: $A$ and $S$
Initialize $A_0$, $S_0$
$n = 0$
while not converged do
    $n = n + 1$
    Choose $\lambda_n$
    $S_n = \underset{S\geq 0}{\operatorname{argmin}} \frac{1}{2}||Y - A_{n-1}S||_2^2 + \lambda||S||_1$
    $A_n = \underset{A\geq 0}{\operatorname{argmin}} \frac{1}{2}||Y - AS_n||_2^2$
$A = A_N$
$S = S_N$
```

Algorithm 2: rGMCA for NMF

2.4 (Re)initializations

As a starting point for all the algorithms, a random initialization is used with several steps of ALS as it is the fastest decreasing algorithm at the beginning. Also, we use some corrections such as Optimal Brain Surgeon [23] to help get a correct initialization using second order information. While this is relatively slow to compute, it can be worthy at the very beginning.

In the GMCA-based algorithms, since the values of the threshold are large in the beginning of the algorithms, it often happens that a row of S or a column of A is set to 0, and hence both the line and the corresponding column are set to 0 after sharing the amplitudes between A and S. As they will not reappear by themselves (this position is stable), they need to be reset efficiently.

A fast way is to check the residue $Y - AS$. As we are looking for a row of S or a column of A which is positive and correlates positively, we can discard the negative part of the residue and study $[Y - AS]_+$. We can directly select a new mixing direction among the columns of the positive residue to make sure it correlates positively with at least one column in A. We take for instance the column of $[Y - AS]_+$ with the largest L_2 norm, yielding a new direction a. We finally define the corresponding row of s as the positive correlation between the selected a and the residue: $x = [a^T[Y - AS]_+]_+$.
We then keep looping to fine other directions if need be, using the new residue $[[Y - AS]_+ - as]_+$.
This method is somehow similar to the "Random Acol" procedure proposed in [24], shown to be relatively efficient and very fast.

While we did not write it in the pseudo-code of the algorithms, this method is implemented for ALS, GMCA and rGMCA so that a line and/or column is reinitialized as soon as an algorithm sets it to zero.

2.5 Thresholding Strategy

At the beginning of all the GMCA-based algorithms, we want only a small set of elements in S to be active, as they will be the largest coefficient, more likely to be relevant. The solution can then be refined little by little. To do so, the threshold must evolve from a large value to its final value λ_∞ swiftly enough to preserve the continuity of the solutions. It will still be necessary to choose a number of steps N large enough to apply the strategy. Also, in our experiments, we applied the decreasing threshold strategy for 60% of the iterations and left the threshold fixed to its final value λ_∞ for the remaining iterations, in order for the solution to be able to stabilize itself with the final threshold.

For GMCA, we preserve this continuity by choosing the evolution of λ so as to have a linear increase of the number of active coefficients, starting from a fraction of $\frac{1}{N}$ coefficients active until the activation of all coefficients above λ_∞.
When a signal is contaminated by a white noise of standard deviation σ, it is classical to use a threshold at $\lambda = 3\sigma$ in order to denoise it. The noiseless case will be considered in the experiments so that λ_∞ is set to 0.

The thresholding strategy is nearly the same with (H)rGMCA. Indeed, we can roughly select the number of coefficients which will be activated for a given threshold by computing a first gradient step

$$S_{n-1} - \tau \nabla_S \mathcal{D}(S_{n-1})$$

However, unlike with GMCA, this is only an estimate, and the thresholding mostly applies on the gradient and not the actual values.

On the other hand, the soft-thresholding step for the source estimation in (S)rGMCA is very different and can be seen as:

$$S_n \leftarrow [S_{n-1} - \tau \nabla_S D(S_{n-1}) - \frac{\lambda}{L}\mathbb{1}_{rn}]_+$$

The continuity is preserved as long as λ evolves swiftly, so that for soft-thresholding, we will use a linear decrease for λ. As for the initial threshold, we want the signals in S_0 to contract towards 0 for the first few iterations, so that we can take λ_0 to be $max(-\nabla_S D(S_0))$. Indeed, only the highest coefficient, which are more likely to be significant, will then remain active

3 Experiments

3.1 Settings

We generate A and S uniformly respectively from the distribution of $|B_{p_A} G_{\alpha_S}|$ and $|B_{p_S} G_{\alpha_S}|$, where:

- B_p is a Bernoulli variable with activation parameter p, i.e it has a probability of p to be 1 and $1-p$ to be 0.
- G_α is a generalized centered and reduced Gaussian variable with shape parameter α.

In practice p and α control 2 kinds of sparsity: the Bernoulli parameter affects the number of actual zeros in A and S, while α selects the sharpness of the distribution. As special cases, for $\alpha = 2$, G_α is a Gaussian variable, and with $\alpha = 1$ it is a Laplacian variable. With $\alpha \leq 1$, G_α is considered as sparse (the lower α, the sparser the density). Both A and S will have different distribution parameters which we will differentiate with subscripts. In the following sections, the data we will consider is directly $Y = AS$.

This model is still very simple as for now all the signals have the same intensity, and we do not add any noise.

If we keep the same notations as previously, $A \in \mathcal{M}_{m,r}(\mathbb{R})$ is the mixing matrix, $S \in \mathcal{M}_{r,n}(\mathbb{R})$ is the source matrix, with n the number of samples, m the length of the signals, r the number of sources. We will study in particular the influence of the number of sources r and their sparsity α_S which have an important impact on the complexity of the problem.

We will evaluate GMCA, (H)rGMCA, (S)rGMCA and compare them with the multiplicative update rule as it is one of the most used algorithm and is the historical way to deal with the problem, and with ALS which is simple and yield much more competitive results. As here $\lambda_\infty = 0$, we will not add a sparse penalization for ALS and the multiplicative update. This preliminary study will therefore permit to observe the influence of a decreasing threshold and the differences between GMCA and rGMCA.

Since the convergence of the algorithms is not always clear, we stopped the algorithms after a large enough number of iteration which we set to 500 (except for the multiplicative update which is much slower) and 100 sub-iteration maximum for (H)rGMCA and (S)rGMCA though in practice the average number of subiteration is about 30.

3.2 Evaluation of the Results

We need evaluation methods which are scale invariant and permutation invariant. We propose to use 2 simple criteria based on the matrix A, which is less biased by the thresholding than S as we do not apply any sparse *prior* to it. We will write A_{ref} for the reference (ground truth) of A and A_{est} for our estimate. In order to make it scale invariant we rescale all the columns of both A to unit norm.

- For a signal x and an estimation of it y, we can compute a signal to noise ratio: $\text{SNR}(x, y) = 20 \log_{10}(\frac{||x||_2}{||x-y||_2})$ which tends to infinity for perfect reconstruction. We can therefore compute an SNR for each pair of reference mixing direction (column of A_{ref}) and estimated mixing direction (column of A_{est}), yielding a matrix $M^{\text{SNR}} = (\ \text{SNR}(a^i_{ref}, a^j_{est})\ )_{ij}$. We finally need to associate the reference and estimated mixing directions by pairs, in other word each reference direction will be associated to a unique estimated direction in order to bypass the permutation invariance. With Π the set of all permutations, this can be done by solving $\hat{\pi} = \underset{\pi \in \Pi}{\text{argmax}}\ tr(M^{\text{SNR}}_{\pi,\pi})$ with the Hungarian algorithm. We can finally compute the mean SNR value for the estimate as $\frac{1}{r} tr(M^{\text{SNR}}_{\hat{\pi},\hat{\pi}})$ which is the first criterion we will use.
- As the problem is multiplicative, it also makes sense to use a multiplicative criterion. In order to do this, a first possibility is to compute $\text{Arccos}(A^T_{est}A_{ref})$ where Arccos is computed componentwise. As the columns of A_{est} and A_{ref} are normalized, this matrix represents the angles between each a^i_{ref} and each a^j_{est}. Once again, we can apply the Hungarian algorithm and find

  $$\min_{\pi \in \Pi} \frac{1}{r} tr(\text{Arccos}(A^T_{est}A_{ref})_{\pi,\pi})$$

  which is a criterion tending to 0 for perfect reconstruction.

3.3 Influence of the number of sources

As a first setting, we can have a look at the evolution of the criteria and their standard deviation when the number of sources is increasing. Indeed, this parameter controls an important part of the complexity of the problem: it increases the possible confusion between the sources and the number of variables to estimate. The other parameters are set to: $m = 200$, $n = 200$, $p_A = 1$ (all coefficients active), $\alpha_A = 2$ (not sparse), $p_S = 0.8$, $\alpha_S = 1$ ("Laplacian sparse"). As there is no noise, $\lambda_\infty = 0$ is used for GMCA-based algorithms.

There are several things to observe from the results of this experience which can be seen in figure 1:

- both criteria behave as was expected, increasing with the number of sources for the mean angle, while the mean SNR is decreasing. We cut the curve for the multiplicative update which is clearly not as good as the other algorithms.
- GMCA-based algorithms outperform the others. However, they do not perform best on the same settings. Indeed, for a very low number of sources, GMCA performs extremely well. When increasing the number of sources, all the algorithms have more difficulty to solve the problem, but (H)rGMCA performs best. Finally for a large number of sources, (S)rGMCA identifies the sources better than all the other algorithms.

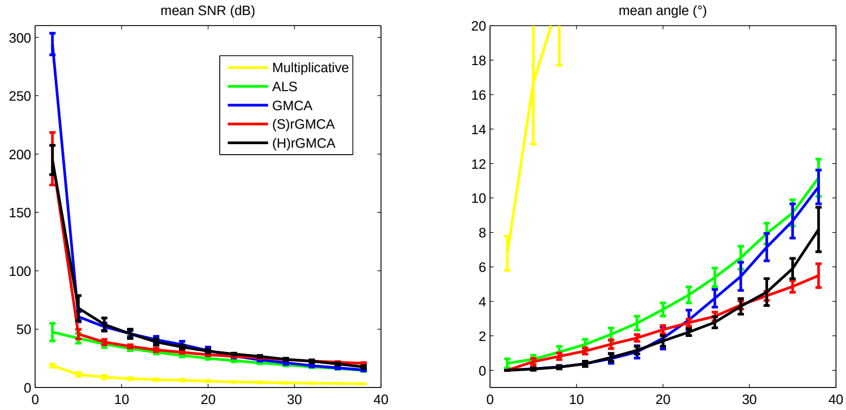

Figure 1: Influence of r on the reconstruction of A

- (S)rGMCA and (H)rGMCA are more consistent/robust in solving the NMF problem. Indeed we can observe that they have the lowest standard deviation for all the criteria (except for the standard deviation of the SNR for the multiplicative update, which is consistently not performing well).
- as in the final iterations ALS and GMCA are performing exactly the same update rules for A and S, we can clearly observe that the decreasing threshold helps avoiding local minima because GMCA is consistently performing better than ALS.

3.4 Influence of the approximate sparsity of the sources

Let us use the same settings as in the previous experience but:

- fixing the number of sources r to 35 in order to focus on the breaking point where soft-thresholding becomes better than hard-thresholding.
- having the parameter α_S vary from 0.4 to 1, i.e. from very sparse to sparse, in the sense of the sharpness of the coefficient distribution. This sharpness is a measure of complexity of the problem as well. Indeed, the sharper, the better is our model and sparsity, and the less correlated the sources are.

The result can be observed in figure 2.

While all the algorithm are sensitive to α_S and become less and less efficient when the sparsity is reduced, (H)rGMCA is the most affected. Indeed, it is very efficient for very small α_S but (S)rGMCA gives better result for larger α_S.

The same behaviour could be observed if we changed the activation rate. However, we could show that the behavior of GMCA is much more dependent on it. Indeed, for low activation rate the precision of the reconstruction is extremely high but decreases quickly so that for large activation rate (S)rGMCA becomes better. As both (H)rGMCA and GMCA apply the same thresholding strategy, the difference must come from the ALS algorithm which does not take well enough into account the correlation between the sources.

3.5 L_0 and L_1 Regularizations

L_1 regularization leads to soft-thresholding, while hard-thresholding tends towards an L_0 regularization. It was observed that both thresholdings led to different behavior in the NMF problem. In order to give

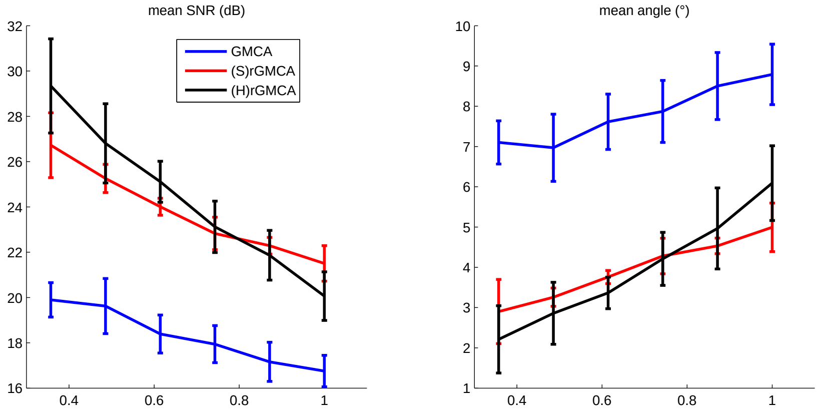

Figure 2: Influence of the activation rate α_S on the reconstructions of A

an explanation to this differences, it is important to understand what effect they have on signals. As can be observed in figure 3 for 2D points with components s_1 and s_2, points which have large values for two sources suffer soft-thresholding twice as much as the ones having only one large value, while with hard-thresholding no large value is penalized.

This is definitely linked to what was shown previously. Indeed, in the case of high sparsity and small number of sources, we observe very large coefficients along the mixing directions and many unimportant small coefficients Those small coefficients are not selected by either thresholdings but soft-thresholding will induce a bias which does not appear with hard-thresholding. It is therefore logical to observe better results of (H)rGMCA in these cases (see figures 1 and 2).

On the other hand, with a larger number of sources, or less sparse data, we increase the probability to obtain points in the quadrant (large values in at least two directions) as there will be more correlations between the sources. By penalizing both values, soft-thresholding tends to give a larger relative weight to the larger of the values hence privileging this direction and un-mixing as much as possible. On the contrary, a point appearing in the quadrant will not be penalized anymore by hard thresholding and comes with its full force, which can unbalance the algorithm if the mixing directions were not properly determined.

3.6 rGMCA Vs GMCA: a Matter of Constraints

In order to understand better why each algorithm behaves better in one case or another, it is interesting to have a look at the evolution of the SNR during the iterations on one example, which can be seen in figure 4 for both GMCA and (S)rGMCA. For this test, the parameters are set to $t = 35$, $\alpha_S = 1$ and $p_S = 0.8$.

We observe that GMCA results decay when no more thresholding is applied. In practice it is possible to observe than S slightly "drifts", losing their sparse *prior*, hence the "refining iterations" are actually detrimental to the results of GMCA. At very large number of sources, this drift is such that we loose the contribution of the decreasing threshold: ALS and GMCA have nearly the same results as can be observed in figure 1. On the other end, the SNR with rGMCA greatly improves once the sparsity parameter λ is set to 0. A properly applied positivity constraint can help refining the reconstruction of A and S once the sources have been sufficiently disambiguated.

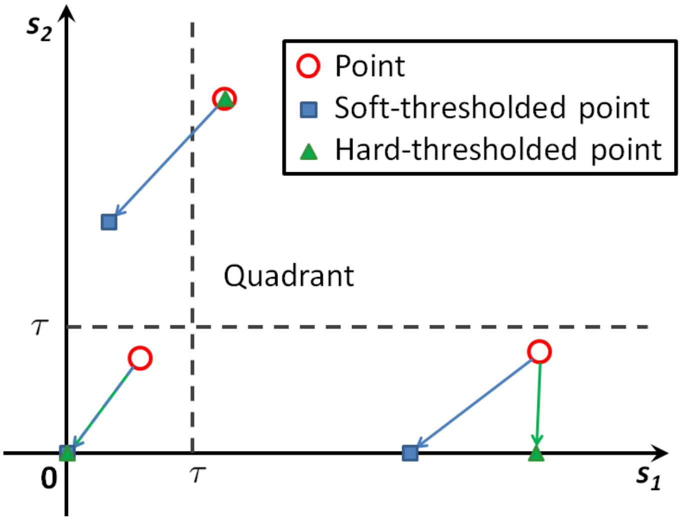

Figure 3: Comparison of soft-thresholding and hard-thresholding

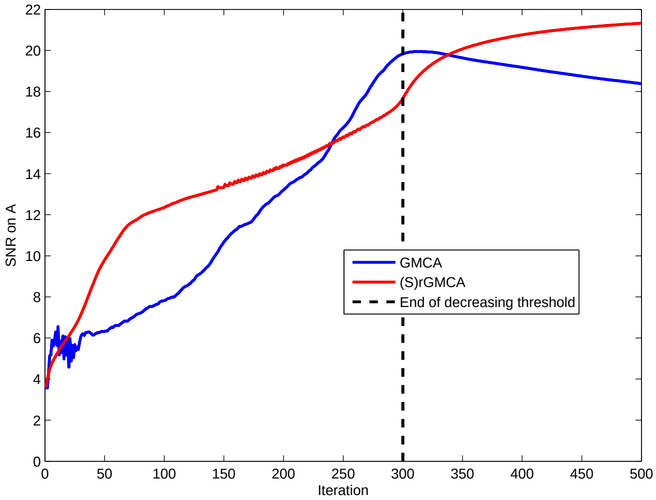

Figure 4: Evolutation of the SNR on A over the iterations in a representative example

Another important indicator to observe is the cost function. As can be seen in the table 1, GMCA is at pain to decrease the cost function and in practice it oscillates when getting too close from the optimal reconstruction. On the other hand, rGMCA can decrease the cost function much further. Once again, this show that not taking properly into account the constraint when there are correlations between the sources lead to instabilities near the solution.

Algorithm	Cost value	Std. dev.
GMCA	1.8760	0.9357
(S)rGMCA	0.0053	0.0023

Table 1: Final cost value for GMCA and rGMCA

Analysis and conclusion

We have tested several algorithms for NMF using simple conditions:

- perfect linear mixture model
- sparse data (both in exact sparsity and approximate sparsity)
- a known number of sources for the reconstruction
- no noise
- same energy for all the sources

In these conditions, we have observed that GMCA performs extremely well in most cases, and that the thresholding strategy has an important impact on the separation of the sources and improves the results.

However, in difficult cases such as with large number of sources, or data which is not sparse enough, rGMCA performs better. Throughout our experiment, we demonstrated that handling correctly the positivity constraints is essential in these cases. Indeed, the correlation between the sources seems to make the global minima unstable when not taking the positivity into account. It is also better to use soft-thresholding than hard-thresholding in those cases and we will therefore focus on soft-thresholding in our future works. Still, rGMCA does not perform as well in more simple cases. The positivity constraint seems therefore too tight to disambiguate the sources properly.

In future works, we plan to combine GMCA and rGMCA in order to enjoy the contribution of both algorithms. We intend for instance to try initializing rGMCA with the results of GMCA. We will also compare our work with state of the art sparse NMF algorithms. As being able to deal with noise is very important for practical applications, we will study its impact on the algorithms. Finally, most signals in applications are not sparse sparse in the direct domain so that it will be interesting to handle this *prior* it in different bases such as in the wavelet domain.

References

[1] W. S. Ouedraogo, A. Souloumiac, M. Jaidane, and C. Jutten, “Geometrical method using simplicial cones for overdetermined nonnegative blind source separation: application to real PET images,” in *Proceedings of the 10th international conference on Latent Variable Analysis and Signal Separation*, LVA/ICA’12, (Berlin, Heidelberg), pp. 494–501, Springer-Verlag, 2012.

[2] M. D. Plumbley and E. Oja, “A “ Nonnegative PCA ” Algorithm for Independent Component Analysis,” *IEEE Transactions on Neural Networks*, vol. 15, no. 1, pp. 66–76, 2004.

[3] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization.,” *Nature*, vol. 401, no. 6755, pp. 788–91, 1999.

[4] A. Cichocki, R. Zdunek, and S.-i. Amari, "Csiszárs divergences for non-negative matrix factorization: Family of new algorithms," *Independent Component Analysis and Blind Signal Separation*, vol. 3889, no. 1, pp. 32–39, 2006.

[5] D. Donoho and V. Stodden, "When does non-negative matrix factorization give a correct decomposition into parts," *Earth*, vol. 16, p. 1141Ð1148, 2003.

[6] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Advances in neural information processing systems*, vol. 13, no. 1, pp. 556–562, 2001.

[7] E. F. Gonzalez and Y. Zhang, "Accelerating the Lee-Seung Algorithm for Nonnegative Matrix Factorization," *Dept. Comput. Appl. Math. Rice Univ Houston TX Tech. Rep. TR0502*, no. TR05-02, pp. 1–13, 2005.

[8] M. Berry, M. Browne, A. Langville, V. Pauca, and R. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Computational Statistics & Data Analysis*, vol. 52, no. 1, pp. 155–173, 2007.

[9] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, pp. 111—-126, 1994.

[10] M. Zibulevsky and B. A. Pearlmutter, "Blind Source Separation by Sparse Decomposition," *Neural Computation*, no. CS99 - 1, pp. 165–174, 1999.

[11] J. Bobin, J.-L. Starck, Y. Moudden, and J. Fadili, "Blind Source Separation : the Sparsity Revolution," *Advances in Imaging and Electron Physics*, vol. 152, no. January 2008, pp. 1–82, 2008.

[12] J. Eggert and E. Körner, "Sparse coding and NMF," *International Joint Conference on Neural Networks IJCNN*, vol. 2, no. 4, pp. 2529–2533, 2004.

[13] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, no. 5, pp. 1457–1469, 2004.

[14] J. Kim and H. Park, "Sparse Nonnegative Matrix Factorization for Clustering," *Processing*, pp. 1–15, 2006.

[15] R. Zdunek and A. Cichocki, "Nonnegative matrix factorization with constrained second-order optimization," *Signal Processing*, vol. 87, no. 8, pp. 1904–1916, 2007.

[16] E. Hale, W. Yin, and Y. Zhang, "A Fixed-Point Continuation Method for l1 -Regularized Minimization with Applications to Compressed Sensing," *Rice University CAAM Technical Report*, vol. TR07-07, 2007.

[17] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization.," *Neural Computation*, vol. 19, no. 10, pp. 2756–2779, 2007.

[18] P. L. Lions and B. Mercier, "Splitting Algorithms for the Sum of Two Nonlinear Operators," *SIAM Journal on Numerical Analysis*, vol. 16, no. 6, pp. 964–979, 1979.

[19] P. L. Combettes and J.-C. Pesquet, "Proximal Splitting Methods in Signal Processing," *Recherche*, vol. 49, no. 8, pp. 1–25, 2009.

[20] Y. Nesterov, "Gradient methods for minimizing composite objective function," *ReCALL*, vol. 76, no. 2007076, p. 2007, 2007.

[21] A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, p. 183, 2009.

[22] Y. Xu, "Alternating proximal gradient method for nonnegative matrix factorization," *arXiv:1112.5407v1*, December 2011.

[23] B. Hassibi, D. G. Stork, and G. J. Wolff, "Optimal Brain Surgeon and General Network Pruning," *Neural Computation*, vol. 1, no. CRC-TR-9235, p. 293–299, 1992.

[24] R. Albright, J. Cox, D. Duling, A. N. Langville, and C. D. Meyer, "Algorithms, initializations, and convergence for the nonnegative matrix factorization," *NCSU Technical Report Math 81706*, no. 919, pp. 1–18, 2006.