Faster and better sparse blind source separation through mini-batch optimization

C Kervazo, T Liaudat, Jerome Bobin

Faster and better sparse blind source separation through mini-batch optimization

C. Kervazo, T. Liaudat and J.Bobin*

IRFU, CEA, Université Paris-Saclay, F91191 Gif-sur-Yvette, France

Abstract

Sparse Blind Source Separation (sBSS) plays a key role in scientific domains as different as biomedical imaging, remote sensing or astrophysics, which require the development of increasingly *faster* and *scalable* BSS methods *without sacrificing the separation performances*. To that end, a new distributed sparse BSS algorithm is introduced based on a mini-batch extension of the Generalized Morphological Component Analysis algorithm (GMCA). Precisely, it combines a robust projected alternate least-squares method with mini-batches optimization. The originality further lies in the use of a manifold-based aggregation of asynchronously estimated mixing matrices. Numerical experiments are carried out on realistic spectroscopic spectra, and highlight the ability of the proposed *distributed* GMCA (dGMCA) to provide very good separation results even when very small mini-batches are used. Quite unexpectedly, it can further outperform the (non-distributed) state-of-the-art methods for highly sparse sources.

*Corresponding author: Jerome Bobin

Email address: name.familyname@cea.fr (C. Kervazo, T. Liaudat and J.Bobin)

Keywords: Blind source separation, Sparse representations, Alternating least-squares, Mini-batches, Matrix factorization, Robust estimator aggregation, Riemannian manifold

1. Introduction

1.1. Towards large-scale BSS

During the last decades, Blind source separation (BSS) has become a major analysis tool to learn meaningful decompositions of multivalued data, in a wide variety of scientific fields such as audio processing [1, 2], biomedical data processing [3] or astrophysics [4], to only cite three of them. According to the standard linear mixture model, the data are composed of m observations, which stem from linear combinations of n elementary signals called sources, each of them having t samples. In matrix form, the model then writes as:

$$\mathbf{X} = \mathbf{AS} + \mathbf{N} \tag{1}$$

where $\mathbf{X}$ (size $m \times t$) is called the observation matrix and $\mathbf{N}$ is the noise matrix accounting for model perturbations. As such, the goal of BSS is to retrieve the mixing coefficients $\mathbf{A}$ (size $m \times n$) as well as the sources $\mathbf{S}$ (size $n \times t$) up to a mere scaling and permutation indeterminacy. BSS is a special case of constrained matrix factorization. Being an ill-posed problem, extra prior information needs to be exploited, such as the statistical independence of the sources (Independent Component Analysis family – ICA [5]), the non-negativity of $\mathbf{A}$ and $\mathbf{S}$ (Non-negative Matrix Factorization – NMF [6, 7, 8, 9]). In this work, we will focus on sparse modelling [10, 11, 12] and assume the sources to be sparse in some domain $\mathbf{\Phi_S}$. This signal representation should

be adapted to the geometrical content of the sources to be separated [13]. The rapid increase of the data size to be analysed mandates the development of *fast* and *scalable* algorithms. So far, sparse BSS has mainly been applied to small (*e.g* the space telescope Chandra [14] – a few million pixels) to middle size data (*e.g* the data of the space mission Planck [15] – half a billion pixels). As such, although currently available methods can be very effective, they are not suited to cope with the data challenges to come in this field (*e.g* the Square Kilometer Array radio-telescope [16] will provide several billions of pixels). The main challenge is then how to design a fast and distributed sparse BSS algorithm *without sacrificing the separation performances ?*

1.2. BSS as an unsupervised matrix factorization problem

In this article, we will focus on tackling noisy sparse BSS problems. For that purpose, the following standard assumptions are made:

- The noise is assumed be Gaussian additive independently and identically distributed [1]. This naturally entails that the cost function will first be composed of a quadratic data fidelity term: $\frac{1}{2}\left\|\mathbf{X}-\mathbf{AS}\right\|_F^2$, where the Frobenius norm $\|.\|_F$ is used to measure the discrepancy between the data and the model.

- The sources are assumed to be sparse in some signal representation $\mathbf{\Phi_S}$ (matrix of size $T \times t$, with $T \geqslant t$). For the sake of simplicity, $\mathbf{\Phi_S}$ will be assumed to be an orthogonal representation. This could also be

[1] This could be easily extended to non-white Gaussian noise by taking into account the noise covariance matrix.

relaxed to more general tight frames [17]. The sparsity of the sources is classically measured by a re-weighted ℓ_1-norm : $\left\|\mathbf{R_S} \odot (\mathbf{S}\mathbf{\Phi_S^T})\right\|_1$. The operator the $\odot$ denotes the Hadamard product. The regularization parameters $\mathbf{R_S}$ (size $n \times T$) control the trade-off between the data fidelity and the sparsity terms.

- To alleviate degenerated solutions ($\|\mathbf{A}\|_F \to \infty$ and $\|\mathbf{S}\|_F \to 0$) due the usual scale indeterminacy in BSS, the last term enforces the mixing matrix to belong to the oblique ensemble: all the columns $\mathbf{A}^i, i \in [1, n]$ of the mixing matrix must lie on the ℓ_2 unit hypersphere. The characteristic function of an arbitrary set U is denoted as $\iota_U(.)$.

Altogether, these assumptions lead to the following generic formulation of the sparse BSS problem:

$$\underset{\mathbf{A}\in\mathbb{R}^{m\times n},\mathbf{S}\in\mathbb{R}^{n\times t}}{\text{minimize}} \frac{1}{2}\|\mathbf{X}-\mathbf{AS}\|_F^2 + \left\|\mathbf{R_S} \odot (\mathbf{S}\mathbf{\Phi_S^T})\right\|_{\ell_1} + \iota_{\{\forall i\in[1,n], \|\mathbf{A}^i\|_{\ell_2}^2=1\}}(\mathbf{A}), \quad (2)$$

Being a multi-convex matrix factorization [18] problem, it is generally tackled by sequentially alternating minimization steps with respect to the mixing matrix $\mathbf{A}$ and the sources $\mathbf{S}$.

Beyond sparse BSS, a standard strategy to cope with large-scale data in generic matrix factorization problems of the form (2) consists in processing small size batches (*i.e.* mini-batches) in a distributed manner. As such, the data are decomposed into sub-matrices (sets of columns of $\mathbf{X}$). So far, the generic distributed sparse matrix factorization algorithms in the literature can be decomposed into two categories :

- *Stochastic approximations*, which have been introduced in the general framework of dictionary learning. In [19], the authors proposed an

online algorithm in which the dictionary (*i.e.* equivalently the mixing matrix) is computed only by minimizing an *upperbound* of the empirical cost. While the online setting is a special case with mini-batches size of $t_b = 1$, the algorithm is generalized to arbitrary values of t_b. More recently, this work has been extended in [20] to cope with datasets that, in our context, would be large both in the number of samples t and number of observations m.

- *Stochastic gradient descent - SGD [21]:* [22] extended the use of the Proximal Alternating Linearized Minimization (PALM) to mini-batches, making it possible to tackle large datasets. Such an approach has also been extended to the case of hyperspectral imaging in [23], in which the authors argue a higher flexibility than in [22] (where the asynchronicity has a high impact on the allowable step sizes, counter-balancing its positive effects).

On the one hand, these generic algorithms can virtually tackle any sparsity-regularized matrix factorization problem with a quadratic data fidelity term. They have however not been particularly adapted for sBSS problems, where the optimization strategy plays a key role to provide satisfactory results [24]. Furthermore, they are generally based on gradient descent, for which such a tuning is cumbersome [24].

On the other hand, the GMCA algorithm [11] is a dedicated sparse BSS algorithm. In contrast, it is based on a projected Alternating Least Square (pALS) scheme, which originates from early works in NMF [25]. More precisely, when updating one of the variables $\mathbf{A}$ or $\mathbf{S}$, a full minimization of the (differentiable) data-fidelity term is carried out, and then a projection over

the non-differentiable constraints is performed by applying the corresponding proximal operator [26]. In this context, the GMCA has been quite successful as it comes with an automatic scheme to fix the regularization parameters $\mathbf{R_S}$. This procedure has been showed to increase its robustness to initialization, local minima and noise. Unfortunately, the GMCA algorithm does not computationally scale very well to large size data.

As such, existing state of the art methods do not currently handle the large-scale sBSS problem properly: on the one hand classical sBSS methods do not scale well with the data size, while generic matrix factorization methods, although computationally efficient, do not usually gives good results in the context of sBSS.

1.3. Contributions

The main contribution of this article is to introduce a new algorithm coined *distributed* GMCA (dGMCA), which combines projected alternate least-squares with mini-batch optimization. The dGMCA algorithm builds upon the parallel computation of estimates of the mixing matrix $\mathbf{A}$ from data mini-batches. The originality of the dGMCA algorithm then lies in the construction of an aggregated estimate using a robust mean on the hypersphere, which better respects the Riemannian geometry of the oblique constraint. Numerical experiments have been carried out on various mixing scenarios. They first highlight that the proposed algorithm provides a scalable sparse BSS algorithm *without sacrificing the quality of the factorization.* Furthermore, and more surprisingly, the dGMCA algorithm can significantly outperform standard sparse BSS algorithm in challenging problems, such as cases where the sources have a highly sparse representation. We discuss

the connections between dGMCA and SGD, which gives some insight into the behavior of dGMCA based on the exploratory power of stochastic mini-batch optimization and the resulting implicit regularization. The dGMCA algorithm is further illustrated on realistic γ-spectroscopy data. Preliminary results were presented in the SPARS 2019 conference [27].

1.3.1. Notations

In the following, scalars will be written in lower case letters a, vectors in bold $\mathbf{a}$ and matrices as $\mathbf{A}$. $\mathbf{A}^j$ stands for the j^{th} column of $\mathbf{A}$, while $\mathbf{A}_i$ is the i^{th} line. These notation are extended to subsets of lines or columns: if J_b is a subset of $[1, t]$, $\mathbf{S}^{J_b}$ denotes the columns indexed by J_b. Generally speaking, we will consider B subsets $J_b, b \in [1, B]$ that form a partition of $[1, t]$, which will be used to index mini-batches. We will furthermore write $\#J_b$ the number of elements in J_b, that is in each mini-batch.

For each mini-batch, we will have access to several estimations of the *same* matrix $\mathbf{A}$. Each one will be denoted by $\mathbf{A}[J_b], b \in [1, B]$. The aggregated estimate of the different $\mathbf{A}^{J_b}$ is denoted as $\hat{\mathbf{A}}$.

In iterative algorithm, we will write $a^{(k)}$ the estimation at iteration k of the variable a.

2. dGMCA: a distributed sparse alternating least-squares algorithm

2.1. The distributed GMCA algorithm

When it comes to distributed computation, a straightforward approach consists in performing the most intensive computational burden on smaller

chunks of data. Following generic distributed matrix factorization algorithms, the proposed method performs by first splitting the data matrix $\mathbf{X}$ into B disjoint sub-matrices. Without loss of generality, the number of samples t is assumed to be a multiple of B, thus leading to a complete decomposition of the full data $\mathbf{X}$ with constant batch size. Therefore, each of the mini-batches has $t_b = t/B$ columns ($\#J_b = t_b$).

In the GMCA algorithm, the updates of mixing matrix $\mathbf{A}$ and the source matrix $\mathbf{S}$ have the largest computational cost. Each iteration k can be described by the following two steps:

1 - $\mathbf{S}$ is updated assuming a fixed $\mathbf{A}$.

$$\mathbf{S}^{(k)} = \mathcal{S}_{\mathbf{R_S}} \left(\mathbf{A}^{(k-1)\dagger} \mathbf{X} \right), \tag{3}$$

where $\mathbf{A}^{(k-1)\dagger}$ is the pseudo-inverse of $\mathbf{A}^{(k-1)}$. The operator $\mathcal{S}_{\mathbf{R_S}}$ is the soft-thresholding operator with thresholding parameters $\mathbf{R_S}$.

2 - $\mathbf{A}$ is updated assuming a fixed $\mathbf{S}$:

$$\mathbf{A}^{(k)} = \Pi_{\|.\|_2=1} \left(\mathbf{X} \mathbf{S}^{(k)\dagger} \right), \tag{4}$$

where $\Pi_{\|.\|_2=1}$ is the projection onto the m-sphere.

In the proposed dGMCA, each main iteration or epoch[2] of the distributed GMCA algorithm can be decomposed into the following two stages:

- The first stage amounts to performing an estimation of the sources and the mixing matrix independently from each batch $\mathbf{X}^{J_b}, b \in [1, B]$. This

[2]If one uses the standard machine learning vocabulary.

is the most computationally intensive part of the algorithm and it is performed in a distributed way.

- Each mini-batch leads to an independent estimate $\{\mathbf{A}[J_b]\}_{b\in[1,B]}$ of the mixing matrix. The second stage consists in combining or *aggregating* these different estimates to eventually produce a single one.

More formally, the dGMCA algorithm is summarized in Algorithm 1 The

Algorithm 1 dGMCA

for $k = 1, \cdots, K$ **do**
  Choose $J_1, J_2, \cdots, J_B$ as a partition of $[1, t]$
  for $b = 1, \cdots, B$ **do**
    $\hat{\mathbf{S}}^{J_b(k)} = \mathcal{S}_{\mathbf{R_S}^{(k)}}(\hat{\mathbf{A}}^{(k-1)\dagger}\mathbf{X}^{J_b(k)})$ ▷ Use Eq. (12) for $\mathbf{R_S}$ choice
    $\mathbf{A}[J_b(k)] = \Pi_{\|.\|_2=1}(\mathbf{X}^{J_b(k)}\hat{\mathbf{S}}^{J_b(k)\dagger})$
  end for
  $\hat{\mathbf{A}}^{(k)} = \texttt{AGGREGATE}(\mathbf{A}[J_1(k)], \mathbf{A}[J_2(k)], ..., \mathbf{A}[J_B(k)])$
end for
return $\hat{\mathbf{A}}^{(K)}, \hat{\mathbf{S}}^{(K)}$

function `AGGREGATE` is used as a generic terminology for the aggregation procedure, which will be described in details in the following section.

2.2. Manifold-based mixing matrix aggregation

The objective of the aggregation step is to build a single estimate of $\mathbf{A}$ from the various estimates $\mathbf{A}[J_b], b \in [1, B]$. For that purpose, the aggregated estimate $\hat{\mathbf{A}}$ can naturally be defined as the barycenter of the different estimates according to some distance ϕ, with some weights $\{\omega_b\}_{b\in[1,b]}$ as follows:

$$\hat{\mathbf{A}} = \text{Argmin}_{\mathbf{A}} \sum_{b=1}^{B} \omega_b \phi\left(\mathbf{A}, \mathbf{A}[J_b]\right), \tag{5}$$

where the barycentric weights are positive and sum to one: $\forall b \in [1, b], \omega_b \geq 0$ and $\sum_{b=1}^{B} \omega_b = 1$. A straightforward choice for ϕ could be the standard Euclidean distance : $\phi\left(\mathbf{A}, \mathbf{A}[J_b]\right) = \sum_{j=1}^{n} \|\mathbf{A}^j - \mathbf{A}^j[J_b]\|_{\ell_2}^2$, which eventually defines $\hat{\mathbf{A}}$ as a standard weighted average of the estimates. However, a key property of (2) is that the mixing matrix should belong to the Oblique manifold, which would not be necessarily satisfied with a standard Euclidean barycenter. For that purpose, dedicated distances will be rather used to precisely take into account the Riemannian geometry of the Oblique constraint.

Frechet mean on the hypersphere

The oblique constraint implies that each column of the mixing matrix $\mathbf{A}$ belongs to the hypersphere of dimension $m-1$ or $m-1$-sphere $\mathcal{S}_{m-1}$, which is a Riemannian manifold (see [28] for more details). Building an aggregation procedure that respects the underlying Riemannian geometry is naturally done by choosing ϕ as the geodesic distance on the $m-1$-sphere. The extension of the notion of barycenter of data points onto a Riemannian manifold equipped with its geodesic distance is known as the Fréchet mean. Precisely, for each column j of the mixing matrix, the Fréchet mean of the individual estimators $\mathbf{A}[J_b]$ is defined as:

$$\forall j \in [1, n], \quad \hat{\mathbf{A}}^j = \text{Argmin}_{\mathbf{a} \in \mathbb{R}^m} \sum_{b \in [1, B]} \omega_b \, d^\beta\left(\mathbf{a}, \mathbf{A}[J_b]^j\right), \tag{6}$$

where $1 \leq \beta < +\infty$. The case $\beta = 2$ corresponds to the ℓ_2 norm along geodesics of the $m-1$-sphere.

Finding the solution to the above minimization problem is not straightforward but can be computed thanks to a dedicated iterative gradient descent algorithm [29]. Under some conditions, this algorithm is proved to converge to a critical point of the above problem. More precisely, each iteration of Afari's algorithm (*cf.* Algorithm 2) are decomposed into two stages:

- *Computation of the gradient :* for $\beta \geq 1$ the gradient $\nabla \mathcal{J}^{\beta}(\hat{\mathbf{A}}^{j^{(k)}})$ of the cost function $\mathcal{J}^{\beta}(\hat{\mathbf{A}}^{j^{(k)}}) = \sum_{b=1}^{B} \omega_b \, d^{\beta}\left(\hat{\mathbf{A}}^{j^{(k)}}, \mathbf{A}^{j}[J_b]\right)$ is defined as :

$$\nabla \mathcal{J}^{\beta}(\hat{\mathbf{A}}^{j^{(k)}}) = -\sum_{b=1}^{B} \omega_b d^{\beta-2}\left(\hat{\mathbf{A}}^{j^{(k)}}, \mathbf{A}^{j}[J_b]\right) \log_{\hat{\mathbf{A}}^{j^{(k)}}}\left(\mathbf{A}^{j}[J_b]\right), \qquad (7)$$

  where the *logmap* $\log_{\hat{\mathbf{A}}^{j^{(k)}}}$ is roughly speaking the projection onto the tangent plane of the $m-1$-sphere about $\hat{\mathbf{A}}^{j^{(k)}}$ [28].

- *Update of the current mean estimate :* the updated estimate is then defined as the exponential map about $\mathbf{A}^{j}[J_b]$ applied to the gradient with step size ρ[3]. This step merely "back-projects" the gradient – which belongs to the tangent space about $\hat{\mathbf{A}}^{j^{(k)}}$ – onto the $m-1$-sphere.

Robust *Fréchet mean on the hypersphere*

The use of small mini-batches makes the separation process more prone to generate outliers in the estimated $\mathbf{A}[J_b]$ (*cf.* Section 3.4 and numerical experiments of Section 3). Unfortunately, the Fréchet mean is not robust to such outliers. As such, more robust distances have been used [30, 31]. In the present setting, a natural choice would be to choose $\beta = 1$, which

[3]It can possibly vary during the optimization process. It will be kept fixed in this article.

Algorithm 2 Fréchet mean

procedure FRÉCHET MEAN ON THE OBLIQUE ENSEMBLE $\mathcal{O}_b(m)$
  for $j = 1, \cdots, n$ **do**
    while convergence is not reached **do**
      $\nabla \mathcal{J}^{\beta}(\hat{\mathbf{A}}^{j^{(k)}}) = -\sum_{b=1}^{B} \omega_b \log_{\hat{\mathbf{A}}^{j^{(k)}}} (\mathbf{A}^j[J_b])$
      $\hat{\mathbf{A}}^{j^{(k+1)}} = \exp_{\hat{\mathbf{A}}^{j^{(k)}}} \left(-\rho \nabla \mathcal{J}^{\beta}(\hat{\mathbf{A}}^{j^{(k)}}))\right)$
    end while
  end for
end procedure

corresponds to the usual ℓ_1 norm. However, the ℓ_1 norm is not differentiable about 0. In dGMCA, this is mitigated by building a smooth approximation d^1_μ of d^1 based on Nesterov's smoothing technique [32]:

$$d^1_\mu(\mathbf{a}, \mathbf{b}) = \text{Argmax}_{\|\mathbf{u}\|_\infty \leq 1} \ < \mathbf{a} - \mathbf{b}, \mathbf{u} > -\frac{\mu}{2}\|\mathbf{u}\|^2_{\ell_2}. \tag{8}$$

For $\mathbf{a}, \mathbf{b} \in \mathbb{R}^m$. This approximated distance is differentiable and its gradient is μ-Lipschitz[4]. This entails that the gradient of the cost function takes the form:

$$\nabla \mathcal{J}^{\beta}(\hat{\mathbf{A}}^{j^{(k)}}) = -\sum_{b=1}^{B} \omega_p \nabla d^1_\mu \left(\hat{\mathbf{A}}^{j^{(k)}} - \mathbf{A}^j[J_b]\right) \log_{\hat{\mathbf{A}}^{j^{(k)}}} \left(\mathbf{A}^j[J_b]\right), \tag{9}$$

where the gradient of d^1_μ is given by [33] for all $i \in [1, m]$:

$$\nabla d^1_\mu(\hat{\mathbf{A}}^{j^{(k)}})_i = \begin{cases} \mu^{-1}\, \hat{\mathbf{A}}_i^{j^{(k)}}, & \text{if } |\hat{\mathbf{A}}_i^{j^{(k)}}| < \mu, \\ \text{sign}(\hat{\mathbf{A}}_i^{j^{(k)}}), & \text{otherwise.} \end{cases} \tag{10}$$

[4]In practice, Nesterov's smoothing parameter was fixed to 0.1, which provides a good balance between speed of convergence and approximation accuracy for this application.

The resulting algorithm is similar to Algorithm 2, with the exception that the distance and its gradient are evaluated according to Equations 8 and 10.

2.3. Implementation details

Use of a reweighted *aggregation*

Using mini-batches of small size generally yields very diverse estimates of the mixing matrix. For instance, in mini-batches where a source is dimly active or inactive at all, the estimated mixing matrix is likely to exhibit a degraded condition number, which eventually leads to a dramatic increase of the noise that contaminates the estimated sources. Therefore, the barycentric weights used for either the Fréchet mean or its robust version are proportional to the noise variance of the estimated sources. Let's denote by $\mathbf{W}[J_b] = \mathbf{A}[J_b]^+$ the pseudo-inverse of the mixing matrix estimated from batch J_b, each weight is defined as ω_b^j:

$$\forall b \in [1, B]; \quad \omega_b^j = \frac{\left(\mathbf{W}[J_b]_j \mathbf{\Sigma_N} \mathbf{W}[J_b]_j{}^T\right)^{-1}}{\sum_{p=1}^{B} \left(\mathbf{W}[J_p]_j \mathbf{\Sigma_N} \mathbf{W}[J_p]_j{}^T\right)^{-1}} \tag{11}$$

where $\mathbf{\Sigma_N}$ is the data noise covariance matrix and $\mathbf{W}[J_b]_j$ is the j-th line of $\mathbf{W}[J_b]$. The maximum number of iterations of the (robust) Fréchet mean evaluation algorithm was fixed to 1000 with a stopping rule so that the algorithm stops when the angle between two iterates is lower than 10^{-6}.

Choosing the regularization parameters

As pointed out earlier, the regularization parameters play a key role, as they correspond to thresholds applied to the sources. Thus, they select specific entries of the sources during the separation process. Following the

morphological diversity principle [17], the last versions of the GMCA algorithm make use of a strategy where the percentile of retained entries of $\mathbf{S}^{(k)}$ linearly increases from step to step, which enables to keep an increasing number of entries in a way which is well adapted to the actual distribution of the sources. This strategy has been shown to lead to a high robustness of the algorithm with respect to spurious local critical points. However, to be applied it requires the knowledge of all the source samples, which is impossible in the proposed distributed optimization procedure, where the mini-batches are processed separately (*i.e.* without communication between the distributed processes). To that end, a strategy based on an exponential decay of the parameters is used:

$$\mathbf{R}_{i_{\mathbf{S}_i}}^{i(k)} = \kappa\sigma_i + \left(\left\|\hat{\mathbf{S}}_i^{(k)}\right\|_\infty - \kappa\sigma_i\right)\exp(-k\alpha_i), \tag{12}$$

where σ_i is the estimated standard deviation[5] of the noise contaminating the source $\mathbf{S}_i$, the constant κ is generally chosen between 1 and 3 based on a desired probability of false discoveries and α_i is a parameter controlling the exponential decay decrease. In practice, α_i is fixed to 2 for all the sources; other choices close to the unit do not change significantly the separation results. The proposed strategy is adapted to distributed computing as $\left\|\hat{\mathbf{S}}_i^{(k)}\right\|_\infty$ can be evaluated as the maximum over each mini-batch: $\left\|\hat{\mathbf{S}}_i^{(k)}\right\|_\infty = \max_{b\in[1,B]}\left\|\hat{\mathbf{S}}_i^{J_b(k)}\right\|_\infty$. As well, σ_i can be evaluated empirically as the median of its estimations over the mini-batches: $\sigma_i = \text{median}_{b\in[1,b]}\,\sigma_i[J_b(k)]$.

[5]For instance, using the Median Absolute Deviation (MAD) estimator.

dGMCA convergence and stopping rule

From the theoretical point of view, the convergence of the distributed GMCA can hardly be proved since: i) it is based on an inexact minimization scheme that alternates least-squares estimates and projections/proximal step, and ii) for the sake of robustness, the thresholding strategy is based on the decrease of the regularization parameter value during the separation process, and iii) each iterate is obtained as a non-linear aggregation of batch-based estimates. However, experiments tend to show that the algorithm stabilizes after a certain number of iterates. If convergence cannot be rigorously claimed, there are empirical clues that support the stabilization of the algorithm in practice. To obtain convergence guarantees, one could envision to incorporate dGMCA as a warm-up step of a two-step algorithm, similarly as in [24]. The second refinement step would then be an asynchronous PALM algorithm [22]. This is out of the scope of this article and left to future work. The stopping rule is based on the maximum (across columns) angular distance between two iterates of the estimated mixing matrix. The algorithm stops when such distance is smaller than 10^{-6}. The maximum number of iterations is fixed to 10^4. As a pre-processing, the columns of the data may be randomized to provide more homogeneous batches. Similarly, randomization has also been tested at each iteration to mimic stochastic mini-batch optimization in the projected ALS framework. For the experiments below, randomisation at each iteration did not provide further significant improvements.

3. Numerical experiments

In this section, the performances of the dGMCA algorithm are investigated in various experimental settings, on both simulated and realistic data.

3.1. Experiments on simulated data

Comparison set-up

In this subsection, we first make use of synthetic random data, which allows to assess the robustness of the different methods in various experimental scenarios. The influence of several parameters has been evaluated: the number of sources n and observations m, the condition number of the mixing matrix and the sparsity level of the sources. Since all led to similar conclusions, we hereafter focus more specifically on the two last items, which yield to the most insightful results. To that end, the data are synthesized as follows:

- The entries of the sources $\mathbf{S}_j, j \in [1, n]$ are independently and identically distributed according to a Generalized Gaussian distribution with parameter $0 < \gamma \leq 1$.
- The mixing matrix is picked at random from a Gaussian distribution, and further processed to have columns with unit ℓ_2 norm and a predefined condition number.

Unless stated differently, each single experimental result will be given as the mean over 25 Monte-Carlo simulations with different mixing matrices, sources and noise realizations.

Comparisons will be carried out with the following unsupervised matrix factorization algorithms:

- **GMCA :** This non-distributed algorithm [11] serves as a baseline to evaluate the proposed distributed separation procedure.
- **Online dictionary learning :** This algorithm [19] is a classical one for solving large-scale sparse matrix factorization problems. For fairer comparisons, the regularization parameter has been optimised based on several simulations.
- **distributed GMCA :** the dGMCA will come with two distinct aggregation procedures, namely with the Fréchet mean and its robust alternative.

To assess the separation quality, the mixing matrix criterion C_A is used:

$$C_A = \text{mean}(|\mathbf{P}\hat{\mathbf{A}}^{\dagger}\mathbf{A} - \mathbf{I}|) \tag{13}$$

With $\mathbf{A}$ the true mixing matrix and $\mathbf{P}\hat{\mathbf{A}}^{\dagger}$ the pseudo-inverse of the solution found by the algorithm corrected through $\mathbf{P}$ for the scale and permutation indeterminacy. The mean is the average of all the elements inside the matrix. In contrast to more standard criteria that are based on the estimated sources, the advantage of the mixing matrix criterion is that it is less sensitive to the regularization of the sources, which generally differs between algorithms.

Sparsity level ρ

The sparsity level ρ of the sources impacts the separation process in two ways:

- The sources are generated so as to be statistically stationary. However, in the very sparse regime (*e.g.* $\rho = 0.1$), the values taken by a single

small mini-batch may largely change between two realizations. This can lead to outlier mini-batches that will impact the aggregation process. In contrast, mildly sparse sources should lead to a more stable aggregation procedure;

- Very sparse sources tend to lead to sharper critical points that are much harder to escape. These local minima are more likely to be smoothed out when the sparsity level decreases.

The number of observations is set to 20, the number of sources to 5 and the number of samples per sources to 10000. The signal-to-noise ratio is fixed to 40dB. The condition number of the mixing matrix is set to 3. As an illustration, Figure 1 displays the evolution of the mixing matrix criterion C_A with respect to the mini-batch size for two values of the sparsity level of the mixing matrix: $\rho = 0.25$ in the left panel and $\rho = 0.5$ in the right panel. These results reveal two first distinct regimes:

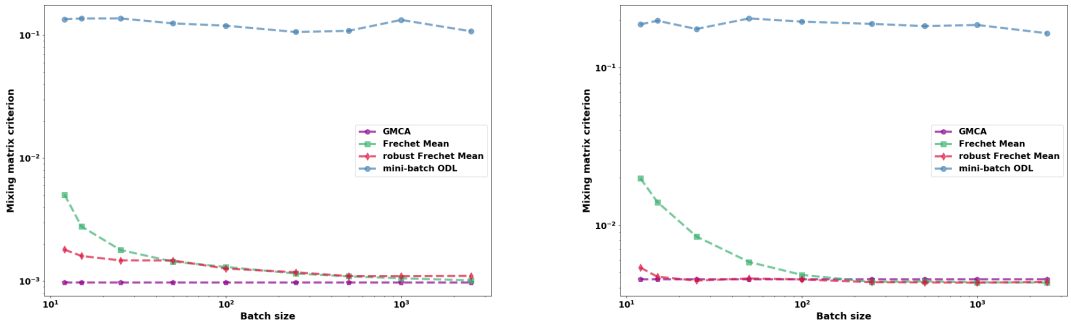
i) For mildly sparse sources (left panel of Fig. 1), the GMCA algorithm yields better results and there is only a slight discrepancy between the two dGMCA methods. This highlights that equipped with the (robust) Fréchet mean, combining mini-batch optimization and aggregation leads no performance loss: dGMCA allows to perform distributed computation without deteriorating significantly the separation quality.

ii) For less sparse sources (cf. right panel of Fig. 1), the results of robust dGMCA are consistent with the previous regime: robust dGMCA reaches a separation quality that is very close to GMCA for even smaller

mini-batches. One of the differences with regime i) is that GMCA obtains slightly worse results. This is expected, as more partial correlations may occur in this regime. Furthermore, the discrepancy between Fréchet mean aggregation and its robust counterpart increases again strongly when the batch size becomes smaller than 100 samples. This might also be due to an increased number of partial correlations: small mini-batches containing such samples might be much more difficult to unmix, generating more inhomogeneous mini-batches.

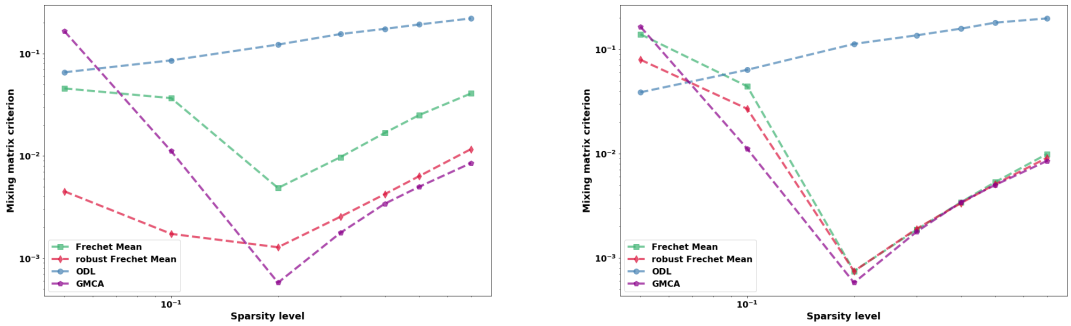
To go further, Fig. 2 shows the evolution of the mixing matrix criterion as a function of the sparsity level for two different mini-batch sizes: $t_b = 10$ and $t_b = 100$ samples. For $\rho > 0.2$, this figure confirms the above comments: equipped with the robust Fréchet mean, the dGMCA and GMCA algorithms provide very close results. As well, the proposed robust aggregation procedure yields significantly better separation quality.

When the sources are highly sparse (typically for $\rho < 0.1$), the performances of the GMCA algorithm rapidly degrade. Very likely, this originates from the sharpness of the critical points, which are much harder to escape. Quite astonishingly, in this precise case, the dGMCA algorithm with the robust Fréchet mean aggregation performs much better than GMCA for small batch size. This surprising results originate from the exploratory power of small batch sizes, which better prevents the dGMCA algorithm to be trapped in sharp critical points. This phenomenon will be discussed in more details in Section 3.4.



(a) $\rho = 0.25$ (b) $\rho = 0.5$

Figure 1: Evolution of the mixing matrix criterion as a function of the sparsity level with $\rho = 0.25$ (left panel) and $\rho = 0.5$.



(a) $t_b = 10$ (b) $t_b = 100$

Figure 2: Evolution of mixing matrix criterion as a function of the sparsity level for mini-batch sizes $t_b = 10$ and $t_b = 100$.

Condition number

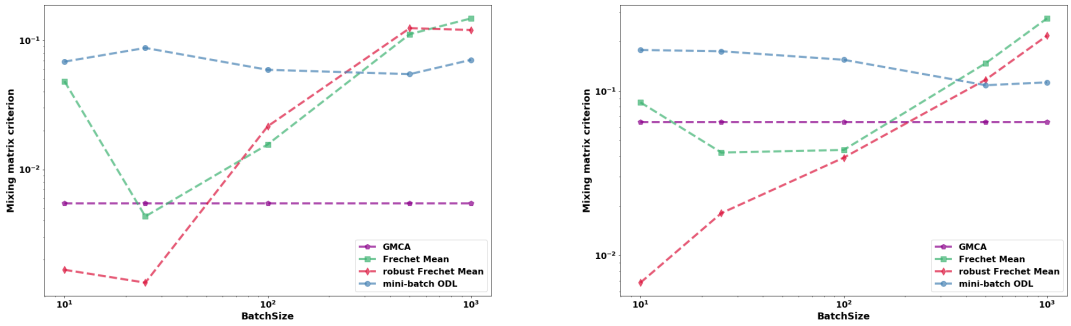
Ill-conditioned mixtures, as measured by the condition number of the mixing matrix $\mathbf{A}$, generally lead to arduous separation problems. Mixing

matrices with large condition numbers yield two major bottlenecks: i) an increased noise level in the source domain, and ii) the mixtures are closer to co-linearity, which makes the sources harder to distinguish.

In these experiments, the noise level is fixed to 40 dB and the sparsity level is $\rho = 0.1$. The number of observations is set to 20, the number of sources to 5 and the number of samples per sources to 10000. Figure 3 shows the evolution of the mixing matrix criterion C_A as a function of the mini-batch size t_b for two values of the condition numbers: left panel 2.5 and right panel 7. As expected, the quality of the separation results of all methods decrease when the condition number increases. Similarly to the tests performed in the previous section, the dGMCA algorithm has better results for relatively small mini-batch sizes (but when the Fréchet mean is used, it eventually deteriorates for $t_b < 25$, cf. Fig. 3). The use of small batches along with the robust Fréchet mean leads to an improvement for $t_b < 25$, which becomes more significant when the condition number increases up to a gain of about one order of magnitude. Similarly, when the mini-batch size decreases, the discrepancy between the two methods increases.

3.2. Implementation and computation time

We now conclude this experimental section with the computation time of dGMCA, which depends both on the complexity of one epoch and the number of required epochs.



(a) Condition number of 2.5

(b) Condition number of 7

Figure 3: Evolution of the mixing matrix criterion as a function of the mini-batch size for two distinct values of the mixing matrix condition number.

Complexity of a single epoch

Each iteration of the GMCA algorithm has a complexity of $\mathcal{O}(t(mn + n^2 + m))$. The complexity of one epoch of dGMCA is similar, once the cost of the Frechet mean has been taken into account:

$$\mathcal{O}(b(mn + n^2 + m) + \frac{t}{b}nmK), \tag{14}$$

where the last term correspond to the Fréchet mean and K corresponds to the number of iterations required for its computation. As such, except for very small mini-batches, the linear gain of using dGMCA over GMCA dominates. This is experimentally confirmed (*cf.* Fig. 4): in practice the computation time for a given number of iteration does not deviate much from linearity (in particular, the transfer costs between the nodes are negligible in comparison to the computation time).

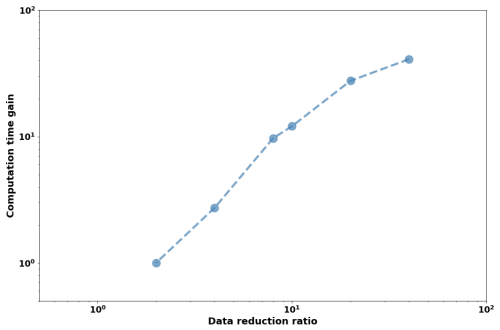


Figure 4: Computation time gain of the dGMCA algorithm with respect to the GMCA algorithm as a function of the reduction gain t/B. The dGMCA algorithm has been run on a PC equipped with 8 Amd CPUs, each one has 6 cores Istanbul Opteron 8431 at $2,4$Ghz. The parallelization of the code has been carried out via python/C++ wrappers with OpenMP.

3.3. Application to γ-ray spectroscopy realistic simulations: sparse case

In this section, the behavior of the dGMCA algorithm is evaluated in the context of γ-ray spectroscopy. This is one of the main methods used for measuring the activity concentrations of radionuclides in environmental samples. It particularly plays a key role to monitor the radiological environment or perform radioecology studies and nuclear incident preparedness. A γ-ray spectrum is the histogram of the number of detected γ-ray photons in the detectors. In this context, an observation is formed by the linear combination of the contribution from various radionuclides. Each one is described by a signature in energy that is composed of one or several emission lines to which a compton continuum is associated as displayed in Figure 5. The goal of this experiment is to jointly estimate the activity of each radionuclide (*a.k.a.* the

mixing matrix) as well as their signature (*a.k.a.* the sources) from several observations. These simulations are composed of 5 radionuclides: ^{7}Be, ^{22}Na, ^{40}K, ^{137}Cs, ^{210}Pb, which are representative of aerosol samples [34] and featured in Figure 5. The number of observations is fixed to $m = 20$ and the number of samples per source is equal to 16940.

These data are particularly interesting as they allow to evaluate the performances of dGMCA when the samples are clearly non-stationary, highly sparse and with a large dynamic range (the source samples basically span 2 to 3 orders of magnitude). The sources are modeled in the wavelet domain: γ-ray observations are first decomposed into an undecimated uni-dimensional wavelet frame [35] before applying any BSS method. The number of scales is fixed to 5, which yields a number of wavelet coefficients that is equal to 81200; these are obviously not large-scale data but it already allows to highlight some remarkable results.

Figure 6 shows the reconstructed solution with GMCA, ODL and dGMCA equipped with the robust Fréchet mean with $t_b = 10$; it also displays the estimation error in transparent solide line. This figure first shows that dGMCA provides a very good reconstruction of $^{2}_{2}$Na signature, while both GMCA and ODL exhibit clear leakage from other sources. The estimation error of the dGMCA solution does not present any structure and is mainly dominated by noise.

Figure 7 features the evolution of the mixing matrix criterion as a function of the mini-batch size for two different levels of the signal-to-noise ratio: 40 and 80 dB. These values might seem large but it has to be recalled that the dynamic range is very large; a small amount of noise might already erase

a significant part of the compton continuum while leaving only the photon peaks. This experiment first shows that GMCA, ODL and dGMCA with the standard Fréchet mean performs rather poorly. In agreement with the results of the previous subsection studying the impact of the sparsity level ρ, the use of the robust aggregation makes the dGMCA algorithm largely outperforms these methods, especially when the mini-batch size is smaller than $t_b = 50$. Further randomizing the mini-batches entails an extra improvement, especially for middle-size mini-batches for $100 < t_b < 1000$. The gain is particularly large when the noise level is small.

To explain the results of such a setting, we advocate the fact that the sources are by a large extent dominated by few photon peaks, which is likely to create spurious critical points that are hard to escape. This might explain why neither the GMCA algorithm nor the dGMCA algorithm without the robust aggregation are able to perform correctly. The use of the robust Fréchet mean makes the algorithm less prone to be stuck in sharp critical points. This will be discussed in the next section.

3.4. Discussion - robustness and implicit regularization

3.4.1. Summing-up the experimental results

From these experiments, two distinct regimes can be highlighted:

- **Mildly sparse sources:** in this setting, the robust dGMCA and GMCA algorithms perform similarly; going distributed comes at almost no cost as soon as the mini-batch size is large enough. For small mini-batch sizes (typically smaller than $t_b = 100$ and even for very small mini-batch sizes), the dGMCA algorithm with the robust aggregation leads to performances that are very close to GMCA. This shows

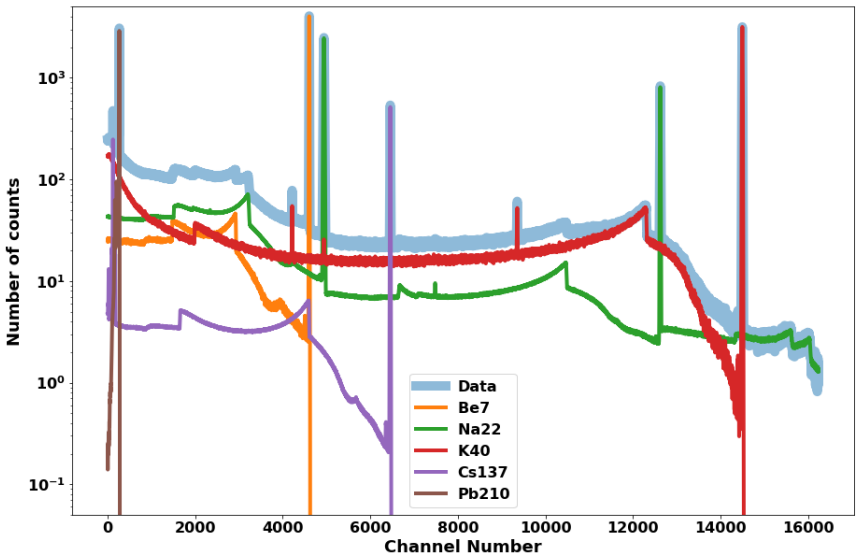


Figure 5: γ**-ray spectroscopy:** example of a single observation and the contribution of each of the radionuclide sources.

that the proposed dGMCA is an efficient approach for separation of large-scale mildly sparse signals, which are typical of natural images with a multiscale representation such as wavelets.

- **Sparse to very sparse sources:** As expected, small mini-batch sizes lead to more heterogeneous mini-batches, which are more likely to be seen as outliers. In this case, the proposed robust Fréchet mean provides superior and more robust separation performances with respect to the standard Fréchet mean. Interestingly, using small mini-batch sizes with dGMCA in its robust version leads to significantly enhanced

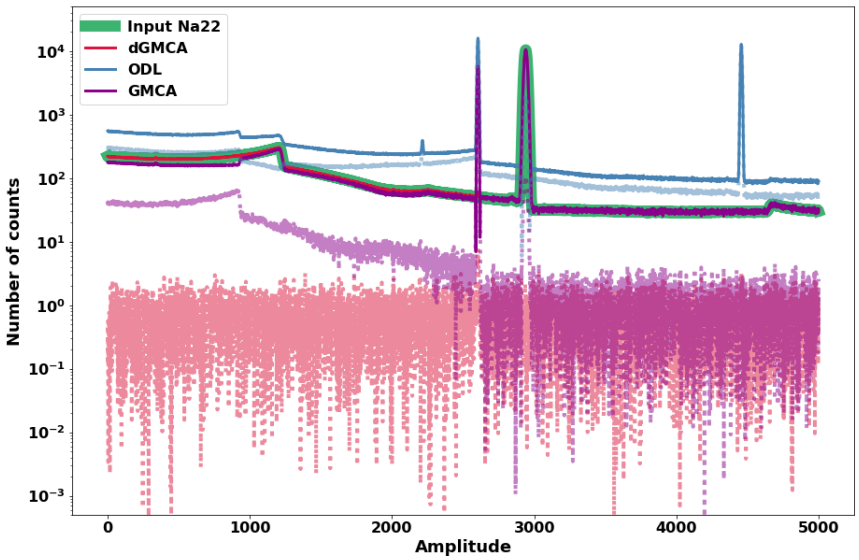
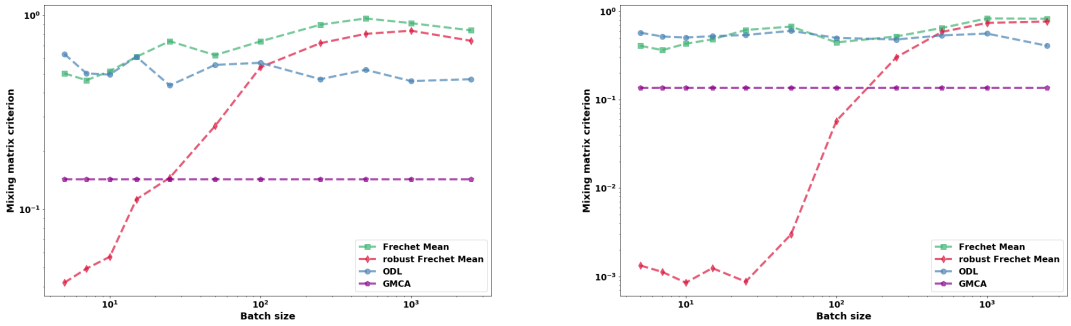


Figure 6: **γ-ray spectroscopy:** estimated ^{22}Na radionuclide with GMCA, ODL and dGMCA equipped with the robust Fréchet mean. Errors with respect to the input spectrum are displayed in transparent solid lines.

results with respect to the GMCA algorithm, which can also be observed for ill-conditioned mixtures. We now give insights concerning this astonishing phenomenon.

3.4.2. Stochasticity of mini-batch optimization and implicit regularization

Connections with mini-batch GD :. A first remark is that either for the update of the mixing matrix or for the sources – and ignoring the projection step – a least-square update is virtually equivalent to a Newton iteration with gradient path length equal to 1. As such, our explanation will be based on links between Gradient Descent (GD) and ALS. More precisely, mini-batch



(a) SNR = 40dB (b) SNR = 80dB

Figure 7: **γ-ray spectroscopy:** evolution of the mixing matrix criterion as a function of the mini-batch size for SNR 40dB and 80dB.

optimization has been a widespread strategy in minimization methods relying on GD[6].

Mini-batch GD induces implicit regularization :. Understanding the impact of optimization in learning parameters from matrix factorization problems [36] or (deep) neural networks [37] has attracted a lot of interest during the last few years. More specifically, it has been emphasized that a specific optimization method might enable some *implicit regularization*, which tends to favor the convergence towards critical points with specific properties. Such a phenomenon is well known in the context of machine learning, where it has

[6]Note that in GD, mini-batch optimization can be implemented in a rather natural way as it requires no aggregation step; the computation of the gradient naturally accumulates information from the different mini-batches.

been noticed for long that using small stochastic mini-batches along with GD can improve the results over full batch methods [38, 39, 40]. A common interpretation is that using small size mini-batches is important as it injects noise in the optimization process, which is essential to escape certain types of critical points. More specifically, it has been empirically shown in [41] that using stochastic mini-batches enables to explore broader areas further away from the initial point. The authors also argue that the stochasticity introduced by small size mini-batches through a structured noise strongly favors flat minimizers that are akin to generalize better [42].

As for BSS, the factorization problem at play is furthermore determined or over-determined, which means that the optimization landscape is likely to be largely different and probably less smooth, leading to the presence of spurious local critical points. In this setting, regularization, either being explicit or implicit, is of paramount importance. To that respect, the case of highly sparse sources is particularly illustrative (*e.g.* the γ-spectroscopy application): dGMCA performs very well as the mini-batch size decreases, while GMCA does not. As such, a similar implicit regularization is very likely to be at play within dGMCA, for which the generalization notion would translate into minimizers that are less sensitive to a given realization of the sources. Indeed, as shown by the γ-ray spectroscopy example, the algorithm performs much better in the highly sparse case, where spurious critical points tend to be sharper and more difficult to escape. The stochasticity induced by mini-batches is likely to prevent the dGMCA algorithm to be stuck in such solutions of the optimization landscape.

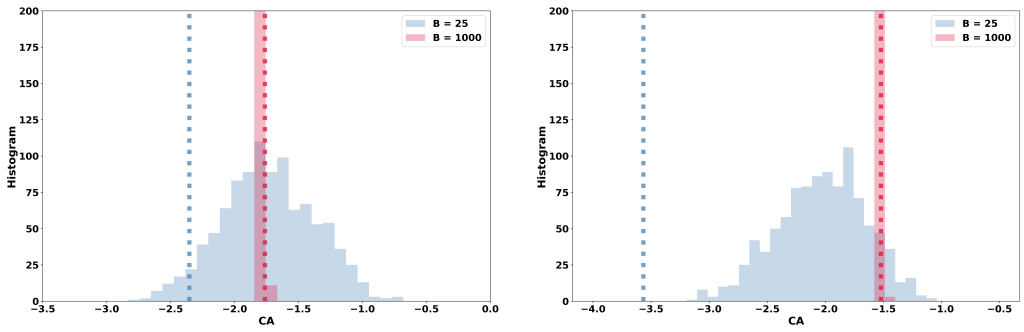
To further illustrate this phenomenon, Figure 8 displays the histograms of the

mixing matrix criterion over the different mini-batches after 1000 iterations of dGMCA (close to "convergence") when the Fréchet mean (left panel) or its robust version (right panel) is used. In this experiment, the number of observations is set to 20, the number of sources to 5, the sparsity level to $\rho = 0.1$, and the condition number of mixing matrix to 1. The number of samples per source is fixed to 100000. In the first case, one can notice that using large size mini-batches provide more stable solutions, with a smaller scatter of the criterion across mini-batches. In contrast, using smaller mini-batches leads to a broader exploration of the parameter space as testified by more widespread values of the mixing matrix criterion. As most of the values are quite poor (close to 10 dB), the aggregated estimate is not satisfactory. Switching to robust aggregation (right panel Figure 8), the observed scatter is very similar but now the aggregated value is much more robust to outlier mini-batches. At first sight, it might look strange to produce an aggregated estimate that is much better than the majority of each individual estimate: it is largely on the left side of the distribution. However, it is likely that the exploration of the optimization landscape goes in random directions, which are not measured in this histogram. It further reveals that the aggregation step is essential to capture an average (and robust) estimate out of this stochastic exploration.

Figure 9 shows the evolution of the mixing matrix criterion when the mini-batch size increases from 10 to 50.10^3 for $B = 10$, 100 and 1000. One can first point out that for large mini-batch size (*i.e.* $t_b = 1000$), the separation accuracy is poor and does not improve when the number of mini-batches increases. In this regime, it is likely that the various estimates of the mixing

matrix do not present enough stochasticity to prevent the algorithm from being stuck in a spurious critical point. For a larger number of mini-batches of smaller sizes, the exploration power highly increases. This define a clearly distinct regime, which suggests some "phase transition" with respect to the mini-batch size for a fixed number of mini-batches.

To sum up, an optimization landscape exploratory phenomenon related to the stochasticity of mini-batch optimization occurs when small mini-batches are considered (typically a few times the number of sources n). In this regime, the separation quality will improve when the number of mini-batches increases. This phenomenon will vanish when the mini-batch size is too large.



(a) Fréchet mean

(b) robust Fréchet mean

Figure 8: Histogram of $\log_{10}$ of the mixing matrix criterion across mini-batches with the Fréchet mean (left) and robust Fréchet mean (right) and $B = 500$ and $B = 10$.

Software

The *dGMCA* algorithm will be made available online as part of the pyGMCALab toolbox (*https://github.com/jbobin/pyGMCALab*).

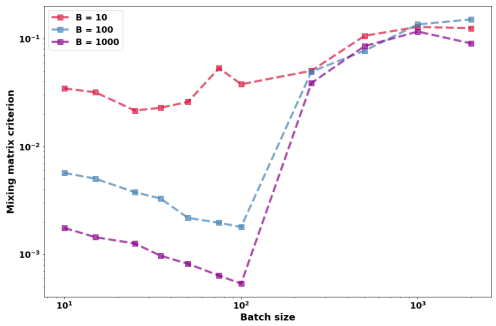


Figure 9: Evolution of the mixing matrix criterion as a function of the mini-batches size for a fixed number of batches of 10, 100 and 1000.

Conclusion

To tackle the large-scale sparse BSS problem, we introduced in this work the dGMCA algorithm, which combines mini-batches with the projected Alternating Least-Square framework. At each iteration, the algorithm separately builds estimators of the mixing matrix from data mini-batches in a distributed manner. These estimates are further aggregated by using two versions of the Fréchet mean, which takes into account the Riemannian geometry of the Oblique constraint. For mildly sparse sources, the dGMCA equipped with the *robust* Fréchet mean leads to a huge gain in computation time *without sacrificing the separation accuracy*. More surprisingly, dGMCA outperforms standard algorithms in challenging blind separation problems, such as ill-conditioned mixtures and/or very sparse sources. Borrowing ideas from the machine learning community, this phenomenon shares similarities with stochastic mini-batch gradient descent, that enables a better exploration of the optimization landscape. Numerical experiments with synthetic

and realistic simulations have been carried out to illustrate the relevance of the approach.

Acknowledgement

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

References

[1] E. Vincent, M. G. Jafari, S. A. Abdallah, M. D. Plumbley, M. E. Davies, Probabilistic modeling paradigms for audio source separation, in: Machine Audition: Principles, Algorithms and Systems, IGI Global, 2011, pp. 162–185.

[2] C. Févotte, N. Bertin, J.-L. Durrieu, Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis, Neural computation 21 (3) (2009) 793–830.

[3] F. Negro, S. Muceli, A. M. Castronovo, A. Holobar, D. Farina, Multi-channel intramuscular and surface EMG decomposition by convolutive blind source separation, Journal of neural engineering 13 (2) (2016) 026027.

[4] J. Bobin, F. Sureau, J.-L. Starck, A. Rassat, P. Paykari, Joint planck and wmap cmb map reconstruction, Astronomy and Astrophysics 563 (A105).

[5] P. Comon, C. Jutten, Handbook of Blind Source Separation: Independent component analysis and applications, Academic Press, 2010.

[6] N. Gillis, F. Glineur, Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization, Neural Computation 24 (4) (2012) 1085–1105.

[7] N. Gillis, Successive nonnegative projection algorithm for robust nonnegative blind source separation, SIAM Journal on Imaging Sciences 7 (2) (2014) 1420–1450.

[8] N. Gillis, The why and how of nonnegative matrix factorization, Regularization, Optimization, Kernels, and Support Vector Machines 12 (257) (2014) 257–291.

[9] A. Vandaele, N. Gillis, F. Glineur, D. Tuyttens, Heuristics for exact nonnegative matrix factorization, Journal of Global Optimization 65 (2) (2016) 369–400.

[10] M. Zibulevsky, B. A. Pearlmutter, Blind source separation by sparse decomposition in a signal dictionary, Neural computation 13 (4) (2001) 863–882.

[11] J. Bobin, J.-L. Starck, J. M. Fadili, Y. Moudden, Sparsity and morphological diversity in blind source separation, IEEE Transactions on Image Processing 16 (11) (2007) 2662–2674.

[12] J. Le Roux, F. J. Weninger, J. R. Hershey, Sparse nmf–half-baked or well done?, Mitsubishi Electric Research Labs (MERL), Cambridge, MA, USA, Tech. Rep., no. TR2015-023.

[13] J. Bobin, J.-L. Starck, Y. Moudden, M. J. Fadili, Blind source separation: The sparsity revolution, Advances in Imaging and Electron Physics 152 (1) (2008) 221–302.

[14] A. Picquenot, F. Acero, J. Bobin, P. Maggi, J. Ballet, G. W. Pratt, Novel method for component separation of extended sources in x-ray astronomy, Astronomy and Astrophysics A139 (626).

[15] J. Bobin, F. Sureau, J.-L. Starck, A. Rassat, P. Paykari, Joint planck and wmap cmb map reconstruction, Astronomy and Astrophysics 563 (2014) A105.

[16] E. Chapman, al., The scale of the problem: recovering images of reionization with generalized morphological component analysis, Monthly Notices of the Royal Astronomical Society 429 (1) (2013) 165–176.

[17] J. Bobin, J. Rapin, A. Larue, J.-L. Starck, Sparsity and adaptivity for the blind separation of partially correlated sources., IEEE Transanctions on Signal Processing 63 (5) (2015) 1199–1213.

[18] Y. Xu, W. Yin, A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion, SIAM Journal on imaging sciences 6 (3) (2013) 1758–1789.

[19] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online learning for matrix factorization and sparse coding, Journal of Machine Learning Research 11 (1) (2010) 19–60.

[20] A. Mensch, J. Mairal, B. Thirion, G. Varoquaux, Stochastic subsampling for factorizing huge matrices, IEEE Transactions on Signal Processing 66 (1) (2018) 113–128.

[21] L. Bottou, Large-scale machine learning with stochastic gradient descent, in: Proceedings of COMPSTAT'2010, Springer, 2010, pp. 177–186.

[22] D. Davis, B. Edmunds, M. Udell, The sound of apalm clapping: Faster nonsmooth nonconvex optimization with stochastic asynchronous palm, in: Advances in Neural Information Processing Systems, 2016, pp. 226–234.

[23] P.-A. Thouvenin, N. Dobigeon, J.-Y. Tourneret, Partially asynchronous distributed unmixing of hyperspectral images, IEEE Transactions on Geoscience and Remote Sensing 4 (57) (2019) 2009–2021.

[24] C. Kervazo, J. Bobin, C. Chenot, F. Sureau, Use of palm for ℓ_1 sparse matrix factorization: Difficulty and rationalization of an heuristic approach., Digital Signal Processing 97.

[25] P. Paatero, U. Tapper, Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values, Environmetrics (1994) 111—126.

[26] N. Parikh, S. Boyd, et al., Proximal algorithms, Foundations and Trends® in Optimization 1 (3) (2014) 127–239.

[27] T. Liaudat, C. Kervazo, J. Bobin, Distributed sparse bss for large-scale datasets, in: SPARS 2019 conference, 2019.

[28] P.-A. Absil, R. Mahony, R. Sepulchre, Optimization algorithms on matrix manifolds, Princeton University Press, 2009.

[29] B. Asfari, R. Tron, R. Vidal, On the convergence of gradient descent for finding the riemannian center of mass, SIAM J. Control Optim. 51 (3) (2013) 2230–2260.

[30] P. T. Fletcher, S. Venkatasubramanian, S. Joshi, Robust statistics on riemannian manifolds via the geometric median, in: 2008 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.

[31] M. Arnaudon, F. Barbaresco, L. Yang, Medians and means in riemannian geometry: existence, uniqueness and computation, in: Matrix Information Geometry, Springer, 2013, pp. 169–197.

[32] Y. Nesterov, Smooth minimization of non-smooth functions, Mathematical programming 103 (1) (2005) 127–152.

[33] S. Becker, J. Bobin, E. Candes, Nesta: a fast and accurate first-order method for sparse recovery, SIAM Journal on Imaging Science 4 (11).

[34] J. Xu, J. Bobin, A. de Vismes Ott, C. Bobin, Sparse spectral unmixing for activity estimation in γ-ray spectrometry applied to environmental measurements, Applied Radiation and Isotopes, in press.

[35] J.-L. Starck, J. Fadili, F. Murtagh, The undecimated wavelet decomposition and its reconstruction, IEEE Transactions on Image Processing 16 (2) (2007) 297–309.

[36] G. Gunasekar, B. Woodworth, S. Bhojanapalli, B. Neyshabur, N. Srebro, Implicit regularization in matrix factorization, in: Proceedings NIPS 2017, 2017.

[37] B. Neyshabur, Implicit regularization in deep learning, Ph.D. thesis, Toyota Technological Institute Chicago, https://arxiv.org/abs/1709.01953 (2017).

[38] Y. A. LeCun, L. Bottou, G. B. Orr, K.-R. Müller, Efficient backprop, in: Neural networks: Tricks of the trade, Springer, 2012, pp. 9–48.

[39] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, P. T. P. Tang, On large-batch training for deep learning: Generalization gap and sharp minima, arXiv preprint arXiv:1609.04836.

[40] M. Hardt, B. Recht, Y. Singer, Train faster, generalize better: Stability of stochastic gradient descent, arXiv preprint arXiv:1509.01240.

[41] C. Xing, D. Arpit, C. Tsirigotis, Y. Bengio, A walk with sgd, arXiv preprint, arXiv:1802.08770.

[42] S. Hochreiter, J. Schmidhuber, Flat minima, Neural Computation 9 (1) (1997) 1–42.