



FACULTAD DE INGENIERIA

Universidad de Buenos Aires

Agosto 2019

Tobías Ignacio LIAUDAT

tobiasliaudat@gmail.com

Desarrollo de Métodos *Sparse* de Separación Ciega de Fuentes para Señales de Gran Tamaño

Tesis de Grado de Ingeniería Electrónica

Departamento de Electrónica, Facultad de Ingeniería, Universidad de Buenos Aires

Supervisor: Jérôme BOBIN

Co-Director del laboratorio CosmoStat

Co-Supervisora: Cecilia G. GALARZA

Profesora en la FIUBA

Laboratorios: CosmoStat, CEA-Saclay, Francia

LPSC, FIUBA, Argentina

© 2019 - ***TOBÍAS IGNACIO LIAUDAT***

ALL RIGHTS RESERVED.

UNIVERSIDAD DE BUENOS AIRES
CIUDAD AUTÓNOMA DE BUENOS AIRES, ARGENTINA

CEA-SACLAY
91191 GIF-SUR-YVETTE, FRANCE

Desarrollo de Métodos *Sparse* de Separación Ciega de Fuentes para Señales de Gran Tamaño

RESUMEN

La separación ciega de fuentes (o *BSS* por sus siglas en inglés) es un problema inverso mal planteado (*ill-posed*) que busca extraer ciertas fuentes desconocidas a partir de observaciones ruidosas que son el resultado de una mezcla de las fuentes mencionadas por una función desconocida. Este problema se encuentra en varias áreas del conocimiento; como en la astrofísica en la estimación de la radiación de fondo de microondas (o *CMB* por sus siglas en inglés) y la reconstrucción de imágenes provenientes de radio-interferómetros; o en química con la técnica de análisis de cromatografía líquida y espectrometría de masa.

Los recientes avances de instrumentos masivos, como el radio-interferómetro continental SKA (*Square Kilometer Array*), van a traer consigo una gran cantidad de datos que deberán ser analizados. A pesar de que existan varios métodos que abordan el problema de *BSS*, estos no se encuentran bien adaptados para trabajar con semejantes ordenes de magnitud. El principal objetivo de esta tesis consistirá en el abordaje del problema *BSS* en el caso donde las señales a tratar son de gran tamaño, es decir que no es posible tratar todas las observación en simultaneo.

El método propuesto, llamado DGMCA (*Distributed Generalized Morphological Component Analysis*), utiliza los conceptos de señales *sparse*, o ralas, que se comprobó que funcionan como una regularización eficiente para la resolución de problemas inversos mal planteados. Su construcción se baso en el algoritmo GMCA que resuelve un problema de optimización no-convexa. Este último algoritmo provee un marco robusto para la *sparse BSS*. La estrategia desarrollada que es distribuida depende en gran parte de la agregación de estimadores y en ese punto es donde el Centro de Masa Riemanniano (CMR) entra en juego. Al tener en cuenta la geometría de los datos considerados, el CMR provee una técnica de agregación robusta. Para mantener el nivel de performance y robustez de su predecesor, DGMA incluye nuevas heurísticas distribuidas que resultan ser efectivas en varias pruebas.

Finalmente, varias pruebas numéricas en datos simulados y en imágenes reales, con el uso de una transformada en onditas apropiada, validaron el enfoque distribuido propuesto alcanzando el mismo nivel de performance en un menor tiempo que el algoritmo GMCA, que se utilizó como referencia.

Keywords: BSS, Sparsity, Factorización de matrices, GMCA, Optimización multi-convexa, Cálculo paralelo, Centro de Masa Riemanniano, Optimización en Variedades de Riemann, Onditas

Desarrollo de Métodos Sparse de Separación Ciega de Fuentes para Señales de Gran Tamaño

ABSTRACT

Blind Source Separation is an ill-posed inverse problem that aims at extracting several unknown sources from noisy observations where these sources are mixed by an unknown function. The mentioned problem arises in many areas of knowledge like astrophysics in the estimation of CMB (Cosmic Microwave Background) maps and the reconstruction of radio-interferometric images, or in chemistry with the LC/MS (Liquid Chromatography / Mass Spectrometry) analysis technique.

The recent development of massive instruments like the continental-size radio-interferometer SKA (Square Kilometer Array) will generate a deluge of data to be treated. Although there are several existing methods to address the BSS problem, they are not well adapted to this new exorbitant orders of magnitude. The main focus of this work is to tackle the large-size BSS problem where it is not possible to treat all the observations at once.

The proposed method, coined DGMCA (Distributed Generalized Morphological Component Analysis), involves sparsity as it has proven to be an efficient regularization to the ill-posed problem and is constructed upon the GMCA algorithm which tackles a non-convex optimization problem. This last algorithm provides a robust framework for sparse BSS. The developed distribution scheme will depend in part of the aggregation of estimators and that's where the Fréchet Mean comes into play. By taking into part the geometry of the data involved, it provides a robust aggregation technique. To maintain the robustness of its predecessor, the DGMCA develops new distributed heuristics that have proved their effectiveness in several tests.

Finally, numerical experiments on simulated data and on real images, with the use of an appropriate wavelet transform, have validated the proposed distributed approach reaching the same level of performance as the GMCA algorithm that is used as benchmark.

Keywords: BSS, Sparsity, Matrix Factorization, GMCA, Multi-convex proximal optimization, Parallel Computing, Fréchet Mean, Optimization in Riemannian Manifolds, Wavelets

Contents

RESUMEN	iii
ABSTRACT	iv
LIST OF FIGURES	viii
AGRADECIMIENTOS	x
NOTATIONS AND ACRONYMS	xii
RESUMEN EXTENDIDO	xiv
0.1 Contexto	xiv
0.2 Introducción	xv
0.3 Desarrollo	xvi
0.3.1 Motivación	xvi
0.3.2 GMCA	xvi
0.3.3 Distributed-GMCA	xviii
0.4 Contribuciones principales	xxiv
1 INTRODUCTION	1
1.1 Motivation	1
1.2 Context	2
1.3 Organization of the manuscript	3
2 SPARSITY AND BLIND SOURCE SEPARATION	5
2.1 Blind Source Separation	5
2.1.1 Blind Source Separation Model	5
2.1.2 The Linear Mixture Model	6
2.1.3 Applications for the BSS	7
2.1.4 Methods for BSS	9
2.2 Sparsity	11
2.2.1 Principles	11

2.2.2	Sparse Blind Source Separation	12
3	ALGORITHMIC FRAMEWORK FOR SPARSE BLIND SOURCE SEPARATION	15
3.1	Proximal Algorithms	15
3.1.1	Proximal Operator	16
3.2	Block Coordinate algorithms	18
3.3	Generalized Morphological Component Analysis	21
3.3.1	Morphological Diversity	22
3.3.2	Morphological Component Analysis	22
3.3.3	Generalized Morphological Component Analysis	23
4	DISTRIBUTED SPARSE BLIND SOURCE SEPARATION	26
4.1	Motivation	26
4.1.1	State-of-the-art methods for batch matrix factorization	27
4.2	Distributed Generalized Morphological Component Analysis	30
4.3	The Fréchet Mean	32
4.3.1	Optimization in the sphere	35
4.4	Heuristics for DGMCA	38
4.4.1	Adaptative thresholding	38
4.4.2	Estimating the optimum α_r parameter - Fitting a GG Distribution	39
4.4.3	Weighting strategies for the Fréchet Mean	41
4.4.4	Correcting permutations of the mixing matrices	42
4.5	Complexity analysis	42
4.6	Enhancing DGMCA	43
4.7	Segmenting lines and columns	44
4.8	The pseudo-algorithms	47
4.9	Numerical experiments	49
4.9.1	Input data	49
4.9.2	Results	50
5	DISTRIBUTED SPARSE BSS IN A TRANSFORMED DOMAIN	56
5.1	DGMCA in a transformed domain	56
5.1.1	Isotropic Undecimated Wavelets: The Starlets	58
5.2	Numerical experiments	60
5.2.1	Input data	60
5.2.2	Results	61
6	CONCLUSION AND PERSPECTIVES	65
6.1	Conclusions and Contributions	65
6.2	Perspectives	67
6.3	Extension to the Joint Deconvolution and BSS	68

6.3.1	Background	68
6.3.2	DBSS - Deconvolution Blind Source Separation	71
6.3.3	DecDGMCA - Distributed Joint Deconvolution and BSS	72
APPENDIX		76
A	ADDITIONAL NUMERICAL EXPERIMENTS	77
A.1	Numerical experiments	77
A.1.1	Test: Thresholding strategy	77
A.1.2	Test: Level of sparsity	79
A.1.3	Test: Number of epochs	79
A.1.4	Test: Number of sources	81
B	SPARS CONFERENCE ARTICLE	83
BIBLIOGRAPHY		88

List of Figures

0.3.1 Esquema ilustrativo del uso de la FM aplicado a las columnas de las distintas $\hat{\mathbf{A}}_j$ correspondientes a la primera señal fuente.	xx
2.1.1 Illustration of the LMM structure from Ref [9].	6
2.1.2 Illustration of the Planck data LMM model from Ref [9].	8
2.1.3 Illustration of the Mars Express data allowing to identify ice water and ice CO_2 from Ref [9] and Ref [26].	9
2.1.4 Simple illustration of the LC/MS data from Ref [9] and Ref [10].	10
2.2.1 Toy example to illustrate sparsity. Representing a linear combination of sines in the direct and the Fourier domain.	11
2.2.2 Realistic example to illustrate a domain transformation to get an approximatively sparse signal. The image in the top left, when considered as a flattened vector is not an approximatively sparse signal. When a wavelet transform is used as in the right images the signal (the first level wavelet decomposition coefficients) becomes approximatively sparse in this new representation. The last plot illustrates the energetic contribution (with an unitary normalization) in function of the number of largest entries used (represented as a percentage).	14
3.1.1 Hard and Soft thresholdings illustration from Ref [10].	17
3.2.1 Illustration to show the optimization steps (in red) over each variable (x and y) in the BC type of optimization.	20
3.3.1 Toy example illustrating how the MDP helps to separate the source in the BSS problem.	22
3.3.2 Scatter plot of the approximatively sparse sources S_1 against S_2 illustrating the ℓ_1 ball for the correctly unmixed sources (green line) and the one for the badly unmixed sources (red line).	23
4.2.1 Basic distribution scheme for the optimization problem.	32
4.3.1 Illustration scheme of the use of the Fréchet mean applied to the first column of the mixing matrix estimations after their alignment as seen in section 4.4.4.	34
4.3.2 Illustration of a sphere \mathbb{S}^{m-1} and its tangent space at point x_0 taken from [18]. As the \mathbb{S}^{m-1} is a manifold embedded in \mathbb{R}^n the tangent plane $T_{x_0}\mathbb{S}^{m-1}$ can be seen as a hyperplane in \mathbb{R}^n tangent to the sphere in x_0 . The tangent map can be defined as the elements orthogonal to x_0 , as $T_{x_0}\mathbb{S}^{m-1} = \{z \in \mathbb{R}^n : x_0^T z = 0\}$	35

4.3.3 Illustration containing the elements used to calculate the Exp map and the Log map on the sphere \mathbb{S}^{m-1} . The image seen is the plane generated by \mathbf{z}_0 and $\hat{\mathbf{z}}_1$ (or \mathbf{z}_2). Then, $\gamma(t)$ is the intersection of the plane with \mathbb{S}^{m-1} , and $L(s)$ is the intersection of the plane with $T_{\mathbf{z}_0}\mathbb{S}^{m-1}$.	36
4.7.1 Scanning and estimating scheme for the DGMCA when the segmentation is done in the columns and the lines.	45
4.9.1 GG distribution densities for different β parameters.	49
4.9.2 Simulations under the GG scheme used for testing the algorithms performance.	50
4.9.3 Performance comparison of the GMCA, the ODL (Online Dictionary Learning [32]), and the DGMCA and its variants. Note that the parameters of the ODL algorithm have been optimized for this experiment by an exhaustive search.	51
4.9.4 Computational time gain between the parallelized DGMCA and the GMCA algorithms against the data reduction ratio which is calculated as the problem total size, t , divided by the size of the mini-batch, $t/t_j = J$. Each point on the figure represents the mean over 10 problems.	52
4.9.5 Performance result comparing the original algorithm GMCA, the DGMCA segmenting only the columns and the DGMCA segmenting the columns and the lines.	53
4.9.6 Convergence of the estimation of $\hat{\alpha}_r$ and $\hat{\beta}$, and its corresponding DGMCA performance using the current estimation.	55
5.1.1 Analysis scheme for one observational channel using an undecimated wavelet transform with 3 levels of decomposition.	58
5.1.2 Starlet transform of an astrophysical image. The addition of the last four figures, from (b) to (e), reproduce exactly the first image (a).	60
5.2.1 Images taken from the Hubble Space Telescope used to test the DGMCA algorithm in a transformed domain.	61
5.2.2 Examples of observation channels chosen randomly between the test set.	62
5.2.3 Performance of the GMCA, DGMCA and xDGMCA algorithms when varying the batch size and the line group size.	63
5.2.4 Performance comparison, while varying α_r^o , of the original GMCA algorithm (with α_r^o fixed to 0.5), the DGMCA segmenting only the columns with and without the estimation of the α_r . The metric used is the $C_{A,mean}$ and the input data is the one seen in figure 5.2.1.	64
6.3.1 Images of the LOFAR project taken from https://www.skatelescope.org	69
6.3.2 Illustration of the different coordinate systems given a pair of antennas. Image taken from [35].	70
6.3.3 Distribution of uv spatial frequencies for the LOFAR instrument given an observational time of 1 hour of a source at the zenith in narrow band. Each correlation between a pair of antennas is represented by a single point in an instant. However, due to the movement of the Earth in the hour of observation, the points move to trace segments of ellipses. The axes are graded in units of wavelength with respect to the center phase of the instrument. Image taken from [33].	71
A.1.1 Performance comparison of two thresholding strategies for different batch sizes.	78
A.1.2 Performance test for different levels of sparsity.	80

A.1.3 Performance level in function of the number of epochs used on the algorithms.	81
A.1.4 Performance level for different number of sources and different sizes of batches.	82

Agradecimientos

Para comenzar, me gustaría agradecer a mi familia por todos los principios y valores que me inculcaron a lo largo de mi vida. También desearía agradecer a mis amigos, a ambos lados del Atlántico, que tanto vivimos juntos.

Continuando, estoy muy agradecido con el gobierno francés y con el programa de becas Eiffel que me dio la oportunidad de realizar este gran intercambio.

Las dos instituciones responsables de mi formación, la Universidad de Buenos Aires y la escuela de ingeniería francesa *IMT Atlantique* se merecen mi agradecimiento por todos los conocimientos que me proporcionaron.

Finalmente, le agradezco a mis dos supervisores, Jérôme Bobin y Cecilia Galarza, por todos los consejos y la ayuda que me dieron.

Notations and Acronyms

Abbreviations

- BSS: Blind Source Separation
- WGN: White Gaussian Noise
- LMM: Linear Mixture Model
- ICA: Independent Component Analysis
- NMF: Non-negative Matrix Factorization
- SBSS: Sparse Blind Source Separation
- CMB: Cosmic Microwave Background
- LS/MS: Liquid Chromatography/Mass Spectrometry
- GMCA: Generalized Morphological Component Analysis
- MCA: Morphological Component Analysis
- AMCA: Adaptative Morphological Component Analysis
- DGMCA: Distributed Generalized Morphological Component Analysis
- xDGMCA: extended Distributed Generalized Morphological Component Analysis
- BC: Block Coordinate methods
- BCD: Block Coordinate Descend
- ALS: Alternating Least Squares
- NMF: Non-negative Matrix Factorization

- MDP: Morphological Diversity Principle
- FM: Fréchet Mean (L^P Riemannian Center of Mass)
- SNR: Signal to Noise Ratio
- GGD: Generalized Gaussian Distribution
- MRI: Magnetic Resonance Imaging
- LOFAR: LOw Frequency ARray
- SKA: Square Kilometer Array
- PSF: Point Spread Function
- CS: Compressed Sensing
- DBSS: Deconvolution Blind Source Separation
- DecDGMCA: Deconvolution Distributed Generalized Morphological Analysis

Operators

- Vectors are written in bold lowercase \mathbf{x} .
- Matrices are written in bold uppercase \mathbf{X} . The k line of the matrix would be \mathbf{x}^k and the t column would be \mathbf{x}_t . A single element of the matrix would be $X_{k,t}$.
- The Fourier transform will be noted as: $\tilde{\mathbf{x}} = \mathcal{F}(\mathbf{x})$
- $\text{supp}(\mathbf{x})$: support of the \mathbf{x} vector (set of non-zero coefficients)
- $\text{card}(C)$: cardinal of the set C (number of elements in C)
- $\text{sign}(x)$: It is the sign function that returns 1 if x is positive or zero and -1 otherwise.
- $\text{diag}(\mathbf{x})$: Is a diagonal matrix containing the vector \mathbf{x} in its diagonal.
- \odot : Hadamard product (element-wise multiplication of matrices)
- $(\mathbf{X})^\dagger$: Moore-Penrose pseudo-inverse that can be calculated as $(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$

Resumen Extendido

0.1 CONTEXTO

Tobías Liaudat¹ comenzó sus estudios en la Facultad de Ingeniería de la Universidad de Buenos Aires (FIUBA)². En el 2016 obtuvo la beca de excelencia Eiffel³ dentro del marco de un acuerdo de doble titulación entre la FIUBA y la escuela de ingeniería francesa IMT Atlantique⁴. El acuerdo consiste en hacer los últimos dos años de estudios en la institución francesa y al final obtener el diploma de ambas casas de estudio.

Dentro de esta experiencia, se desarrolló la tesis de grado para la obtención del diploma francés en el laboratorio CosmoStat⁵. Este se encuentra dentro del centro de investigación CEA-Saclay⁶, ubicado en Francia, en la región Parisina, y pertenece a dos de sus departamentos, al Departamento de Astrofísica (*DAP: Département d'AstroPhysique*) y al Departamento de Electrónica, Detectores e Informática para la Física (*DEDIP: Département d'électronique, des détecteurs et d'informatique pour la physique*).

El laboratorio se especializa en el tratamiento de datos para la astrofísica, y en particular para la cosmología. Esta compuesto en una medida similar por físicos y por profesionales especializados en procesamiento de señales y estadística, con un número aproximado de 30 personas entre investigadores permanentes, post-doctorandos, doctorandos y tesistas de grado.

El tutor dentro del laboratorio fue Jérôme Bobin⁷ quien es el líder de la parte de procesamiento de señales. Durante el desarrollo de la tesis también se trabajo en colaboración con Christophe Kervazo⁸ que realiza su doctorado en el laboratorio en temas relacionados al de esta tesis. El trabajo

¹<https://www.linkedin.com/in/tobias-liaudat/>

²<http://fi.uba.ar/>

³<https://www.campusfrance.org/fr/le-programme-de-bourses-d-excellence-eiffel>

⁴<https://www.imt-atlantique.fr/fr>

⁵<http://www.cosmostat.org/>

⁶<http://www.cea.fr/>

⁷<http://jbobin.cosmostat.org/>

⁸<https://www.cosmostat.org/people/christophe-kervazo>

realizado en dicho laboratorio fue remunerado por lo que durante seis meses se cumplió el horario laboral completo de 35 horas por semana siendo este el número de horas legal para Francia, aunque en la práctica la cantidad de horas era mayor.

La motivación de esta tesis reside en el advenimiento de nuevos instrumentos para realizar observaciones del universo como el ambicioso proyecto SKA⁹ (*Square Kilometer Array*) que se encuentra actualmente en construcción. Este consiste en varios radiointerferómetros localizados en Sudáfrica y Australia, que mediante métodos de apertura sintética va a lograr sintetizar imágenes con una resolución angular muy pequeña. El aluvión de datos que se deberán procesar en este tipo de instrumentos es enorme y representa un gran desafío en cuanto al tiempo y la calidad del procesamiento.

0.2 INTRODUCCIÓN

El problema que se abordará en esta tesis es el de *Blind Source Separation* (BSS), un problema inverso donde el objetivo es tratar de estimar unas señales fuente a partir de observaciones que son el resultado de aplicar una función desconocida al conjunto de señales fuente. Por lo general se considera que la observación tiene presente un ruido aditivo. Dentro de este problema vamos a interesarnos en el caso donde las funciones de mezcla son lineales, y el ruido aditivo es Gaussiano y blanco. El modelo del problema inverso sobre el cual se trabajará es el siguiente:

$$\mathbf{X} = \mathbf{AS} + \mathbf{N}, \quad (1)$$

donde en general $\mathbf{X}, \mathbf{N} \in \mathbb{R}^{m \times t}$, $\mathbf{A} \in \mathbb{R}^{m \times n}$ y $\mathbf{S} \in \mathbb{R}^{n \times t}$. Las observaciones están representadas por \mathbf{X} , la matriz de mezcla por \mathbf{A} , la matriz de fuentes por \mathbf{S} y la matriz de ruido por \mathbf{N} .

Este problema no es nuevo y ya se han realizado varios trabajos sobre el mismo [27]. La gran diferencia entre los distintos métodos son las hipótesis que se toman sobre la matriz de fuentes. Al ser un problema inverso mal planteado (*ill-posed*) existen varias soluciones que verifican las observaciones y es necesario regularizar el problema para poder elegir cuál dentro de todas las posibles soluciones será la que nos interese. La hipótesis que vamos a elegir consiste en que las señales fuente van a ser *sparse* en alguna base de representación. Las señales *sparse* se caracterizan por poder representarse por una pequeña cantidad de coeficientes no nulos.

El problema recién descrito fue abordado por el tutor J. Bobin en varios desarrollos [1][2][3] con el método GMCA (*Generalized Morphological Component Analysis*) que dio buenos resultados. Sin embargo, el método es viable siempre y cuando el tamaño de los datos a tratar no sea demasiado grande. Con esta descripción ambigua, nos referimos a que no pueda ser guardada en memoria para

⁹<https://www.skatelescope.org/>

un uso completo de las matrices. Es decir, que no se puede trabajar con la matriz en su totalidad simultáneamente. Este fue el puntapié inicial de la tesis.

0.3 DESARROLLO

0.3.1 MOTIVACIÓN

Considerando el éxito que tuvo el método GMCA, y que en nuevos proyectos en desarrollo, el tamaño de las observaciones es cada vez más grande, se decidió emprender en el desarrollo de un método que:

- Pueda resolver el problema de BSS sin poder tener acceso a las matrices en su totalidad en simultaneo, sino que solo sea posible trabajar con porciones de las matrices.
- Además de resolver el problema de BSS mantenga el mismo nivel de performance que GMCA.
- Sea capaz de acelerar el tiempo de cálculo considerando matrices de gran tamaño.

Para dar una idea de los ordenes de magnitud con los que se puede llegar a trabajar, el numero de píxeles puede llegar a ser $t \sim 10^9$, el número de canales de observación $m \sim 10^4$ y el número de fuentes $n \sim 1$.

0.3.2 GMCA

La solución al problema de *sparse* BSS se puede escribir como el problema de optimización siguiente:

$$\hat{\mathbf{A}}, \hat{\mathbf{S}} = \arg \min_{\mathbf{A}, \mathbf{S}} \frac{1}{2} \|\mathbf{X} - \mathbf{AS}\|_F^2 + \left\| \Lambda \odot \mathbf{S} \Phi^{tr} \right\|_p + i_{\mathbf{Y}: \|\mathbf{Y}_k\|_2=1, \forall k}(\mathbf{A}). \quad (2)$$

El primer término se encuentra para que la solución hallada se adecue a las observaciones considerando que el ruido es aditivo blanco y Gaussiano. El segundo término permite regularizar la solución al elegir la más *sparse*, donde $p \in \{0, 1\}$, Λ es la matriz de coeficientes de regularización y Φ^{tr} una matriz de cambio de base puesto que las señales fuente deben de ser *sparse* pero no necesariamente deben serlo en la base de representación de las observaciones. El tercer y último término consta de una función indicatriz (vale 0 si la condición que presenta se verifica y vale ∞ si la condición no se verifica), y sirve para evitar que las soluciones se degeneren debido a la indeterminación de amplitud. Como estamos tratando con un factorización de matrices, puede existir una constante multiplicativa que multiplique a una matriz y que su inversa multiplique a la otra, como por ejemplo $\hat{\mathbf{A}} = \alpha \mathbf{A}$, $\hat{\mathbf{S}} = \frac{1}{\alpha} \mathbf{S}$. En tal caso, la matriz de mezcla \mathbf{A} podría tender a infinito y la matriz

de señales fuente \mathbf{S} a cero. Con este último término imponemos que las columnas de la matriz de mezcla \mathbf{A} tengan norma unitaria.

Este problema de optimización no es simple ya que no es convexo ni diferenciable. Si observamos bien, este problema es multi-convexo, es decir, que si fijamos una variable, el problema es convexo respecto a la otra y viceversa. Sin embargo, para tratar este tipo de problemas se pueden usar técnicas de *Block Coordinate Descend* (BCD) donde se optimiza respecto a una sola de las variables a la vez. Entonces podemos usar métodos de optimización convexa [29] y métodos de algoritmos proximales (*proximal algorithms*) [30] para tratar con la no diferenciabilidad de la función objetivo.

Existen varios métodos de tipo BCD para resolver problemas de optimización [11]. En el caso de GMCA, se utilizó un método llamado PALS (*Projected Alternating Least Squares*). Como se puede observar en el algoritmo 1 consta de una etapa de resolución de mínimos cuadrados para la parte diferenciable y luego se utiliza el operador proximal para la etapa de optimización del término no diferenciable. En cada iteración se optimiza con respecto a una variable y luego se sigue con la otra. Las principales razones de la elección de esta estrategia de optimización es por su bajo tiempo de cálculo y por el hecho de que al complementarlo con una estrategia de elección del parámetro Λ del umbral, se logra una robustez con respecto del punto de inicialización del algoritmo. Por más que bajo este método no existan pruebas de convergencia, en la práctica el algoritmo se estabiliza en un valor y lo hace a la solución esperada. En cambio, otros métodos como PALM (*Proximal Alternating Linearized Minimization*) [14] que tienen pruebas de convergencia hacia un mínimo local, son muy sensibles al punto de inicialización y por lo tanto no robustos.

Vale la pena destacar la importancia de la robustez al punto de inicialización ya que permite lograr una buena estimación de las matrices \mathbf{A} y \mathbf{S} . Una vez que se tiene una buena aproximación de las matrices se puede lanzar una etapa de refinamiento de la solución como en [7] utilizando PALM. Otra opción es utilizar algoritmos como FISTA (*Fast Iterative Shrinkage-Thresholding Algorithm*) [13] para calcular \mathbf{S} considerando que la matriz \mathbf{A} calculada previamente es correcta.

En el algoritmo 1 pueden verse las expresiones de los pasos principales de GMCA, mientras que en el algoritmo 2 pueden verse las expresiones cerradas de los cálculos de GMCA. Uno de los éxitos del algoritmo GMCA es la elección del parámetro de Λ que se calcula automáticamente a cada iteración de manera decreciente para poder adaptarse a las señales tratadas. Primero se utilizan solo las señales más discriminantes para luego ir abarcando cada vez una porción mas grande de la señal y poder llegar a una estimación más precisa de la matriz de mezcla. El valor final del parametro depende del nivel de ruido estimado y se utiliza el estimador MAD (*Median Absolute Deviation*) para

Algorithm 1 [Simple Version] GMCA

```
1: procedure GMCA( $\mathbf{X}$ , parameters)
2:   Initialize :  $\hat{\mathbf{A}}, \hat{\mathbf{S}}$ 
3:
4:   for  $k = 0, \dots, N_{epoch}$  do
5:     (I) fit term:  $\tilde{\mathbf{S}}_{(k+1)} \leftarrow \arg \min_{\mathbf{S}} \frac{1}{2} \|\mathbf{X} - \hat{\mathbf{A}}_{(k)} \mathbf{S}\|_F^2$ 
6:
7:     (II) Reg. term:  $\hat{\mathbf{S}}_{(k+1)} \leftarrow \text{prox}_{(\|\Lambda \odot \mathbf{S}\|_p)}(\tilde{\mathbf{S}}_{(k+1)})$ 
8:
9:     (III) Data fit term:  $\tilde{\mathbf{A}}_{(k+1)} \leftarrow \arg \min_{\mathbf{A}} \frac{1}{2} \|\mathbf{X} - \mathbf{A} \hat{\mathbf{S}}_{(k+1)}\|_F^2$ 
10:    (IV) Reg. term:  $\hat{\mathbf{A}}_{(k+1)} \leftarrow \text{prox}_{i_{\mathbf{Y}: \|\mathbf{Y}^k\|_2=1, \forall k}(\mathbf{A})}(\tilde{\mathbf{A}}_{(k+1)})$ 
11:   return  $\mathbf{A}, \mathbf{S}$ 
```

Algorithm 2 [Algorithmic version] GMCA

```
1: procedure GMCA( $\mathbf{X}$ , parameters)
2:   Initialize :  $\hat{\mathbf{A}}, \hat{\mathbf{S}}$ 
3:
4:   for  $k = 0, \dots, N_{epoch}$  do
5:     Estimation:  $\tilde{\mathbf{S}}_{(k+1)} \leftarrow \hat{\mathbf{A}}_{(k)}^\dagger \mathbf{X}$ 
6:     Thresholding:  $\hat{\mathbf{S}}_{(k+1)} \leftarrow Th_\Lambda(\tilde{\mathbf{S}}_{(k+1)})$ 
7:     Estimation:  $\hat{\mathbf{A}}_{(k+1)} \leftarrow \mathbf{X} \hat{\mathbf{S}}_{(k+1)}^\dagger$ 
8:     Projection:  $\hat{\mathbf{A}}_{(k+1)}^j \leftarrow \frac{\hat{\mathbf{A}}_{(k+1)}^j}{\|\hat{\mathbf{A}}_{(k+1)}^j\|_2}, \forall j \in \{1, \dots, N_{cols}\}$ 
9:   return  $\mathbf{A}, \mathbf{S}$ 
```

calcularlo.

0.3.3 DISTRIBUTED-GMCA

PLANTEO

La idea propuesta consiste en dividir el problema en subproblema segmentando la dimensión más problemática que es t . Si se observa el modelo lineal considerado, al segmentar en la dimensión t (columnas de \mathbf{X} y \mathbf{S}), obtenemos problemas que pueden ser resueltos independientemente. Sin embargo, todos los subproblemas tienen algo en común y es que comparten la misma matriz de mezcla. Estos pueden formularse de la siguiente manera:

$$\text{Subproblem } j : \mathbf{X}_j = \mathbf{A} \mathbf{S}_j + \mathbf{N}_j, \quad (3)$$

donde $\mathbf{N}_j, \mathbf{X}_j \in \mathbb{R}^{m \times t_j}$, $\mathbf{A}_{(j)} \in \mathbb{R}^{m \times n}$ y $\mathbf{S}_j \in \mathbb{R}^{n \times t_j}$. Los t_j es el número de columnas tomado en cada subproblema tal que $t = \sum_j t_j$.

Siguiendo el planteo podemos reescribir el problema de optimización que queremos resolver:

$$\hat{\mathbf{A}}, \hat{\mathbf{S}} = \arg \min_{\mathbf{A}, \mathbf{S}} \sum_{j=1}^T \frac{1}{2} \|\mathbf{X}_j - \mathbf{A}\mathbf{S}_j\|_F^2 + \|\Lambda_j \odot \mathbf{S}_j\|_p + i_{\mathbf{Y}: \|\mathbf{Y}_k\|_2=1, \forall k}(\mathbf{A}) \quad (4)$$

Continuamos con los métodos de tipo BCD y escribimos los dos problemas de optimización que resultado de solo optimizar una de las variables:

$$\hat{\mathbf{S}}_j = \arg \min_{\mathbf{S}_j} \frac{1}{2} \|\mathbf{X}_j - \mathbf{A}\mathbf{S}_j\|_F^2 + \|\Lambda_j \odot \mathbf{S}_j\|_p, \quad \forall j = 1, \dots, T \quad (5)$$

$$\hat{\mathbf{A}} = \arg \min_{\mathbf{A}} \sum_{j=1}^T \frac{1}{2} \|\mathbf{X}_j - \mathbf{A}\mathbf{S}_j\|_F^2 + i_{\mathbf{Y}: \|\mathbf{Y}_k\|_2=1, \forall k}(\mathbf{A}) \quad (6)$$

El primer problema de la ecuación 5 puede ser resuelto de manera similar a la de GMCA. Sin embargo, la estimación de \mathbf{A} resulta ser más compleja ya que involucra a toda la matriz \mathbf{X} y \mathbf{S} .

Para resolver este dilema, vamos a proponer realizar estimaciones independiente de la matriz de mezcla, una para cada subproblema, como se observa a continuación:

$$\hat{\mathbf{A}}_{(j)} = \arg \min_{\mathbf{A}} \frac{1}{2} \|\mathbf{X}_j - \mathbf{A}\mathbf{S}_j\|_F^2 + i_{\mathbf{Y}: \|\mathbf{Y}_k\|_2=1, \forall k}(\mathbf{A}), \quad (7)$$

para luego agregar estas distintas estimaciones en una única que sea más robusta y pueda aprovechar el hecho de que la matriz de mezcla estimada es la misma para todos los subproblemas.

MÉTODO DE AGREGACIÓN: RCM

Para agregar las matrices se propone utilizar el centro de masa Riemanniano (*RCM: Riemannian Center of Mass*). Si se pone atención al término de la función objetivo que impone la norma unitaria para las columnas de \mathbf{A} , se puede ver que las columnas pertenecen a una variedad Riemanniana, que es la esfera unitaria de \mathbb{R}^m que se escribe \mathbb{S}^{m-1} . Al utilizar una geometría no Euclidiana podemos sacar ventaja del conocimiento que tenemos sobre las columnas de la matriz de mezcla y la variedad Riemanniana en la que habitan.

Para agregar las matrices se buscara ir agregando columna a columna calculando el RCM o

Fréchet Mean (FM) de cada grupo de columnas que corresponde a la misma señal fuente.

El RCM del conjunto de datos $\{\mathbf{a}_i\}_{i=1}^T \in \mathbb{S}^{m-1}$ donde $\mathbf{a}_i \in \mathbb{R}^m$ y \mathbb{S}^{m-1} siendo la esfera unitaria ℓ_2 de dimensión $m - 1$ se calcula como el argumento que minimiza la siguiente función:

$$f_P(\mathbf{x}) = \frac{1}{P} \sum_{i=1}^T \omega_i d^P(\mathbf{x}, \mathbf{a}_i), \quad 1 \leq P \leq \infty \quad (8)$$

teniendo pesos de ponderación fijos $0 \leq \omega_i \leq 1$ ($\sum_{i=1}^T \omega_i = 1$) y una distancia geodésica $d^P(\cdot, \cdot)$. En nuestro trabajo vamos a utilizar $P = 2$. Guiándonos por los trabajos de Afsari [16] [17], podemos asegurarnos de la existencia y de la convergencia del algoritmo de optimización elegido que consiste en realizar un descenso por gradiente (*gradient descent*) de la función objetivo. Es de menester destacar que se pueden calcular las expresiones explícitas necesarias para el cálculo, como la distancia geodésica, el *log-map* y el *exp-map*.

Continuando, el esquema de agregación de la matriz de mezcla se puede observar en la figura 0.3.1. Antes de agregar las matrices de mezcla es necesario alinearlas para que las columnas que se agreguen correspondan a la misma señal fuente. El pseudo-algoritmo 9 muestra una versión simplificada de la idea de DGMCA.

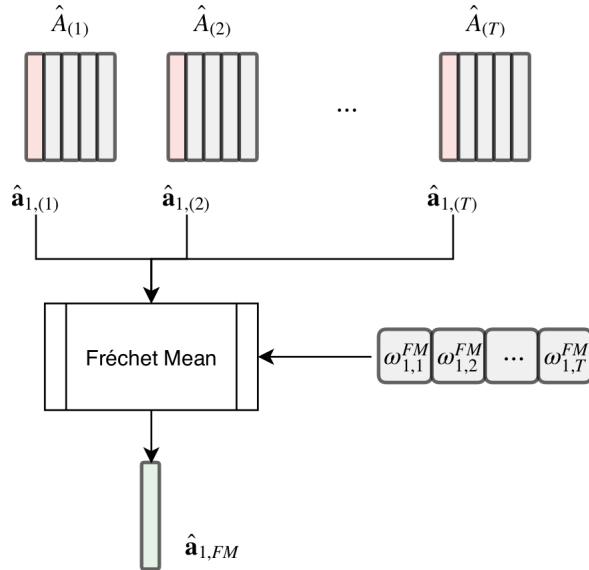


Figure 0.3.1: Esquema ilustrativo del uso de la FM aplicado a las columnas de las distintas $\hat{\mathbf{A}}_j$ correspondientes a la primera señal fuente.

Algorithm 3 [Simple Version] DGMCA

```

1:  $(\cdot)_j$  : Batch  $j$ 
2:  $(\cdot)_{(j)}$  : Estimated with the batch  $j$ 
3:  $(\cdot)^{(k)}$  : Epoch number  $k$ 
4:  $N_{epoch}$  : Maximal number of epochs
5:  $T$  : Total number of batches (group of columns)
6:  $\mathbf{a}_i$  : Column  $i$  from the  $\mathbf{A}$  matrix
7:  $Th_{\Lambda_j^{(k)}}$  : Threshold operator, hard or soft, with  $\Lambda_j^{(k)}$  as thresholding value
8:  $\mathbf{W}_{FM}^{(k)}$  : Weight matrix for the weighted Fréchet Mean
9:
10: procedure DGMCA( $\mathbf{X}$ , parameters)
11:   Initialize :  $\hat{\mathbf{A}}, \hat{\mathbf{S}}$ 
12:
13:   for  $k = 0, \dots, N_{epoch}$  do
14:     Calculation of:  $\Lambda^{(k)}, \mathbf{W}_{FM}^{(k)}$ 
15:     for  $j = 0, \dots, T$  do
16:       Estimation:  $\tilde{\mathbf{S}}_j^{(k+1)} \leftarrow (\hat{\mathbf{A}}_{FM}^{(k)})^\dagger \mathbf{X}_j$ 
17:       Thresholding:  $\hat{\mathbf{S}}_j^{(k+1)} \leftarrow Th_{\Lambda_j^{(k)}}(\tilde{\mathbf{S}}_j^{(k+1)})$ 
18:       Estimation:  $\hat{\mathbf{A}}_{(j)}^{(k+1)} \leftarrow \mathbf{X}_j (\hat{\mathbf{S}}_j^{(k+1)})^\dagger$ 
19:       Projection:  $\hat{\mathbf{a}}_{i, (j)}^{(k+1)} \leftarrow \frac{\hat{\mathbf{a}}_{i, (j)}^{(k+1)}}{\|\hat{\mathbf{a}}_{i, (j)}^{(k+1)}\|_2}, \forall i \in \{1, \dots, n\}$ 
20:     Fusion:  $\hat{\mathbf{A}}_{FM}^{(k+1)} \leftarrow \text{FrechetMean}\left(\hat{\mathbf{A}}_{(0)}^{(k+1)}, \dots, \hat{\mathbf{A}}_{(j)}^{(k+1)}, \dots, \hat{\mathbf{A}}_{(T)}^{(k+1)}, \mathbf{W}_{FM}^{(k)}\right)$ 
21:   return  $\hat{\mathbf{A}}_{FM}^{(k+1)}, \hat{\mathbf{S}}^{(k+1)}$ 
  
```

HEURÍSTICAS

Para que el algoritmo 3 propuesto, que será llamado DGMCA (*Distributed Generalized Morphological Component Analysis*) tenga un buen desempeño y sea robusto frente al ruido y señales fuente con características aproximadamente *sparse* es necesario utilizar ciertas heurísticas. Las señales aproximadamente *sparse* hacen referencia a señales en donde la mayor parte de su información esta agrupada en un pequeño número de coeficientes sin necesidad de que los demás sean nulos.

Umbral adaptativo

Siguiendo la idea de GMCA, vamos a proponer un esquema que vaya decreciendo y dependa de características de la señal tratada. Se debe considerar que seguimos bajo la hipótesis de no poder

utilizar simultáneamente toda la matriz \mathbf{S} y \mathbf{X} . La proposición es la siguiente:

$$\lambda_r = K_\sigma \sigma_{N,r} + (\|\mathbf{S}^r\|_\infty - K_\sigma \sigma_{N,r}) e^{-k\alpha_r}, \quad (9)$$

donde λ_r es el nivel de umbral para la señal fuente r , $\sigma_{N,r}$ es una estimación del ruido para la señal fuente r que se obtiene utilizando el estimador MAD, α_r es el parámetro que controla el nivel de decaimiento de la función exponencial, k es el número de la iteración y K_σ es un multiplicador que se setea en función del nivel final de umbral deseado. En general, usando $K_\sigma = 3$ se obtiene un buen compromiso entre el *denoising* y el sesgo en la estimación. Al utilizar la norma infinito tenemos una característica de las señales fuente que pueden calcularse de manera distribuida. Es decir, que podemos calcular la norma en cada subproblema y luego calcular la norma a estos resultados.

Es fundamental el uso de la estrategia del umbral adaptativo ya que brinda una gran robustez al punto de inicialización del algoritmo de optimización. Al utilizar sólo los valores más grandes de las señales fuentes estimadas para estimar la matriz de mezcla \mathbf{A} se logra orientar la optimización hacia el mínimo global y evitar quedar atrapado en mínimos locales. Con el curso de las iteraciones se va afinando la optimización al ir relajando el umbral. Este va abarcando señales fuentes de menor amplitud hasta un cierto valor que depende del piso de ruido estimado.

Estrategia de ponderación en el RCM

Para esto, vamos a considerar una elección basada en una estimación de la relación señal a ruido de las estimaciones realizadas (*SNR: Signal-to-Noise-Ratio*). Trabajando este concepto se llega a la siguiente expresión:

$$\omega_{r,j}^{FM} = \frac{\|\hat{\mathbf{s}}_j^r\|_2^2}{\sigma_{X_j} \|\hat{\mathbf{A}}_{FM}^\dagger\|_F^2}, \quad \forall j \in \{1, \dots, T\}, \forall r \in \{1, \dots, n\} \quad (10)$$

donde σ_{X_j} es una estimación del ruido en el batch considerado de la matriz de observaciones que puede ser conocido de antemano o puede calcularse. El denominador se compone de una estimación del ruido considerando la matriz de mezcla que se le aplica.

Esta estrategia permite penalizar las estimaciones proveniente de subproblemas que utilicen las observaciones mas ruidosas y a la vez destacar las estimaciones que contengan un gran contenido de señal útil, siendo así más discriminante.

DOMINIO DE REPRESENTACIÓN TRANSFORMADO

Hasta ahora se había considerado que el dominio de representación donde las señales fuente son *sparse* era el mismo que el dominio de representación de las observaciones. Sin embargo, este no

es siempre el caso.

Para sobrelevar este tema considerando que trabajamos con matrices de gran tamaño, se propone utilizar una transformada en onditas que sea localizada, para poder ir transformando de a secciones de las imágenes con las que trabajamos.

Considerando la bibliografía existente sobre distintas bases de representación [19], se eligieron las *starlets* (o también llamadas *Isotropic Undecimated Wavelets*) debido a su simple implementación y a que son localizadas. Además, al ser isotrópicas, se encuentran bien adaptadas a muchas de las estructuras que se observan en las imágenes de la astrofísica que se tratan en el laboratorio CosmoStat.

La estrategia adoptada es la de transformar las observaciones de su dominio de representación original, hacia el dominio donde las señales fuente son *sparse* (o aproximadamente *sparse*) y trabajar directamente con el algoritmo DGMCA descrito previamente en ese dominio. La otra opción que fue descartada es de transformar la estimación de las fuentes en cada iteración ya que el término de adecuación a las observaciones y el término que impone la *sparsity* de la solución no tienen el mismo dominio de representación. El problema de optimización puede expresarse de la siguiente manera:

$$\hat{\mathbf{A}}, \hat{\mathbf{S}}_\Phi = \arg \min_{\mathbf{A}, \mathbf{S}_\Phi} \frac{1}{2} \|\mathbf{X}_\Phi - \mathbf{AS}_\Phi\|_F^2 + \|\Lambda \odot \mathbf{S}_\Phi\|_p + i_{\mathbf{Y}: \|\mathbf{Y}^k\|_2=1, \forall k}(\mathbf{A}). \quad (11)$$

Una vez resuelto el problema de optimización, solo falta utilizar el mismo esquema de transformación pero utilizando la transformada inversa para así volver al dominio de observación.

RESULTADOS

Los principales resultados se pueden dividir en tres:

- Se logró resolver el problema de BSS planteado para matrices de gran tamaño.
- Se mantuvo el nivel de performance de separación comparando con GMCA, siendo también robusto contra el ruido y a distintos tipos de señales fuente.
- El cálculo de complejidad algorítmica arroja una ganancia lineal en tiempo de cálculo con el número de subproblemas que se utilice. Este resultado fue confirmado por experiencias numéricas.

El algoritmo fue probado con señales simuladas y con señales realistas, donde los dominios de representación no coinciden, arrojando muy buenos resultados en ambos casos. Algunos resultados pueden observarse en el artículo corto que se adjunta en el anexo B al final de esta tesis.

0.4 CONTRIBUCIONES PRINCIPALES

Las contribuciones principales fueron:

- El desarrollo teórico del algoritmo DGMCA que no solo logró mantener el mismo nivel de performance de separación que GMCA sino que también tiene una aceleración lineal respecto del mismo.
- La implementación completa del algoritmo DGMCA en python que se encuentra disponible en github¹⁰. Esta versión no es una implementación que trabaja en paralelo, pero simula la implementación distribuida. Sirvió para realizar las experiencias numéricas y como maqueta para la versión distribuida.
- Se implementó una versión en C++ utilizando la librería Boost que trabaja en paralelo en los distintos cores de un CPU. Para la experiencia del tiempo de cálculo se utilizó el cluster de cálculo del laboratorio donde se paralelizó en 40 cores. Esta versión del algoritmo todavía no se hizo pública y el contribuyente principal fue Jérôme Bobin.
- Se publicó un artículo en la conferencia europea SPARS¹¹ 2019 (*Signal Processing with Adaptive Sparse Structured Representations*) donde se presentó un poster del trabajo realizado.
- El trabajo realizado sirvió como puntapié para seguir investigando sobre este tipo de métodos dado los buenos resultados a los que se llegaron. Al realizar modificaciones a la definición de la función objetivo en el cálculo del RCM y a la distancia utilizada, se pueden lograr performances superiores a GMCA con ganancias lineales en el tiempo de cálculo. En esta línea se encuentra trabajando principalmente Jérôme Bobin, con colaboraciones de Christophe Kervazo y Tobías Liaudat, el autor de esta tesis. Se espera una primera versión de un artículo para Agosto del 2019.
- Se dejan las puertas abiertas para seguir con el desarrollo para encarar el problema que consiste en la BSS que presenta una convolución a la mezcla. Este problema de deconvolución y BSS puede tratarse en simultáneo [8], y el método DGMCA puede servir para poder tratar el problema mencionado cuando se trabaje con matrices de gran tamaño.

¹⁰<https://github.com/tobias-liaudat/DGMCA>

¹¹<http://www.spars-workshop.org/en/index.html>

*“Los hermanos sean unidos
Porque esa es la ley primera
Porque si entre ellos se pelean,
Los devoran los de afuera.”*

El gaucho Martín Fierro - José Hernández

1

Introduction

1.1 MOTIVATION

Forthcoming astrophysical observations of our Universe in the radio wavelengths will be provided by forthcoming continental-size radio-interferometers, such as the SKA¹. These radio-interferometers are composed of hundreds or thousands of antennas, which produce dramatically large amounts of data. Such a deluge of data requires the development of dedicated, computationally efficient, algorithms to analyze them. In this context, blind source separation is one of the key tools to analyze multi-wavelength radio-interferometric data, which is composed of different measurements of the sky taken in different wavelength bands. Each measurement contains an unknown mixture of unknown sources. The objective is to be able to estimate the sources and the mixing function.

Being the BSS a matrix factorization problem that is severely ill-posed, it is necessary to add a *priori* information in order to select from all the possible solutions of this particular inverse problem. The regularization used is the imposition of the sparsity of the sources. Thanks to all the work done with different wavelets transforms and signal representations [19] it is possible to intelligently choose a suitable transform to sparsely represent the signal.

The CosmoStat² laboratory has much worked in BSS methods with sparse regularization mostly

¹<https://www.skatelescope.org/>

²<http://www.cosmostat.org/>

based on Jérôme Bobin, my supervisor, avant-garde work [1] [2] [3]. The proposed methods started evolving to cover more applications while giving promising results. It has been used for the CMB map estimation from Plank and WMAP³ data, hyperspectral unmixing, sparse non-negative matrix factorization and robust sparse BSS, just to mention a few [4] [6] [7] [8] [9] [10].

The existing methods give promising results in terms of separation performance but they are not especially conceived to work with large scale data. The data wall has to be demolished to keep up with the progress in the following years. *That will be the target of this work.*

To accomplish the goal we will make our way through convex and non-convex optimization, and the use of proximal algorithms. This is a consequence of the variational approach considered to solve the inverse problem, as it makes use of a *multi-convex* cost function. In order to cope with large scale data the proposed approach needs to aggregate estimators and that is where the *Fréchet mean* (also known as the L_P Riemannian center of mass) will play its role. Its calculation will be done using an optimization technique over a Riemannian manifold that is issued by a previously defined unit norm constraint. During the internship several heuristics have been developed and complementary methods applied in order to give the needed robustness to the algorithm.

1.2 CONTEXT

The CEA (*Commissariat à l'Energie Atomique et aux Energies Alternatives*) is a reknown french public research establishment which plays a leading role in a wide range of fields: low carbon energies (nuclear and renewable), technological research for the industry, fundamental research and, defense and security. It is perhaps best characterized by the level of excellence of its scientific research and by the strength of its link with the industry. *The CEA was identified in 2017 by Thomson-Reuters / Clarivate as the most innovative public research organization in Europe.*

The organism is divided into a number of Directions, each representing one of its different missions. In particular, the DRF (*Direction de la Recherche Fondamental* - Fundamental Research Division) is the branch of the CEA that conducts intensive research in fundamental physics, chemistry and biology. The division is in charge of the fundamental research in several domains like biotechnology, materials, space, health, physics and nanosciences. Nowadays the division has more than 6000 personnel working, from permanent researchers to PhD students, interns and post-docs. The DRF is made up of 11 research institutes covering all the domains already mentioned.

The *Institut de Recherche sur les lois Fondamentales de l'Univers* (Irfu - Institute of Research on the Fundamental Laws of the Universe) is one of the 11 institutes of the DRF and specializes in the fields of astrophysics, particle physics and nuclear physics. The strength of this world-renowned institute lies in its unique set of skills which ranges from leading theoretical physics to cutting-edge

³Planck and WMAP are the two latest full-sky microwave surveys.

engineering. This allows the Irfu to apply its expertise to any step of the conception, realization and exploitation of the most ambitious physics experiments of our time.

The Irfu is organised into 7 departments, 3 of them are oriented towards physics:

- DAP: *Département d'AstroPhysique* (Astrophysics Department)
- DPHP: *Département de PHysique des Particules* (Particle Physics Department)
- DPHN: *Département de PHysique Nucléaire* (Nuclear Physics Department)

and the remaining 4 are specialized in the several fields of engineering required for the conception of the scientific equipment used in the experiments:

- DEDIP: *Département d'électronique, des détecteurs et d'informatique pour la physique* (Department of Electronics, Detectors and Computer Science for Physics)
- DIS: *Département d'ingénierie des systèmes* (Systems Engineering Department)
- DACM: *Département des Accélérateurs, de Cryogénie et de Magnétisme* (Accelerators, Cryogenics and Magnetism Department)
- GANIL: *Grand Accélérateur National d'Ions Lourds* (Large National Heavy Ion Accelerator)

The present work was done in the CosmoStat laboratory (LCS) that is composed of both cosmologists and computer scientists working together to develop new methods of statistics, signal processing, and apply them to cosmological data sets. The LCS is involved several international projects like PLANCK and Euclid and belongs to the DAP, one of the 7 divisions of Irfu.

The DAP is deeply involved in a number of international collaborations and is at the forefront of astrophysics research. This department is part of the AIM laboratory, which is a mixed research unit regrouping teams from the DAP, DEDIP, Paris Diderot University and CNRS (*Centre National de la Recherche Scientifique* - French National Center for Scientific Research). The strength of this laboratory is to gather all the multi-disciplinary skills necessary to the fast and efficient development and exploitation of instruments for astrophysical research. This unique organization is a good example of Irfu's interdisciplinary synergy.

1.3 ORGANIZATION OF THE MANUSCRIPT

The current chapter introduces the motivations and the context of this work. The rest is organized as follows. In chapter 2 we will give some necessary background on sparsity, blind source separation and how the use of the first one can be useful to tackle the second one. The chapter 3 consist of notions in

optimization and the GMCA algorithm that is the basis from we will build the new method. In chapter 4 a complete description of the DGMCA method can be found including numerical experiments in the last part. The chapter 5 consist of the DGMCA algorithm when the observations are not sparse so the domain of representation will have to be transformed. The last chapter will include the conclusions of this work but also the perspectives that include an extension of the developed algorithm for the joint deconvolution and BSS which is able to work in compressed sensing frameworks like radio-interferometers and MRI imaging.

*“Nace el hombre con la astucia
Que ha de servirle de guía
Sin ella sucumbiría,
Pero sigún mi experiencia
Se vuelve en unos prudencia
Y en los otros picardía.”*

El gaucho Martín Fierro - José Hernández

2

Sparsity and Blind Source Separation

SPARSITY EVOKE THE CONCEPT THAT A SIGNAL CAN BE REPRESENTED WITH A SMALL NUMBER OF COEFFICIENTS FROM A GIVEN DICTIONARY. The arising and exploitation of this concept has brought several advances in many inverse problems such as denoising, impainting, blind source separation, deconvolution, just to name a few examples. One of the main advantages is that the knowledge that a signal is sparse in a given dictionary can be used as a prior in order to regularize an ill-posed inverse problem. In simple terms this accounts to give insight to be able to choose a specific solution from all the ones that verify the original inverse problem.

The blind source separation problem will be described afterwards as well as the conditions and hypothesis that will be used during this work. Finally, some theoretical insight will be provided on how the sparsity can be used to tackle the BSS.

2.1 BLIND SOURCE SEPARATION

2.1.1 BLIND SOURCE SEPARATION MODEL

The Blind Source Separation (BSS) can be simply described as the estimation of a reduced number of signals that are called "sources" and of the mixing function from noisy observations of a mixture of these signals. It is in general a hard problem to tackle as the the estimation has to be done simulta-

neously of the sources and of the mixing function leading to an ill-posed inverse problem. This justify the necessity of using some type of *prior* to be able to converge to an exploitable solution. Some of the most known and studied priors are statistical independence, non-negativity or sparsity.

The nature of the noise involved will depend on the physical characteristics of the problem addressed. *From now on the noise will be considered as to be White Gaussian Noise (WGN)*. Other type of noises for example: Poisson noise contamination issued from the counting of events, that are Poisson processes, need specific considerations.

The basic equation describing the BSS problem is the following one:

$$\mathbf{X} = f(\mathbf{S}) + \mathbf{N} \quad (2.1)$$

where in general $\mathbf{X}, \mathbf{N} \in \mathbb{R}^{m \times t}$ and $f : \mathbb{R}^{n \times l} \rightarrow \mathbb{R}^{m \times t}$.

2.1.2 THE LINEAR MIXTURE MODEL

The model that will be studied in the Blind Source Separation problem is the Linear Mixture Model (LMM). It is a special case of the one in equation (2.1) where the observations are a linear combination of the sources with additive noise. The equation describing the LMM is the following one:

$$\mathbf{X} = \mathbf{AS} + \mathbf{N} \quad (2.2)$$

where in general $\mathbf{X}, \mathbf{N} \in \mathbb{R}^{m \times t}$, $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{S} \in \mathbb{R}^{n \times t}$.

In this notation n is the number of sources present in the mixture, m is the number of channels or observations of the system and t is the number of samples of each observation.

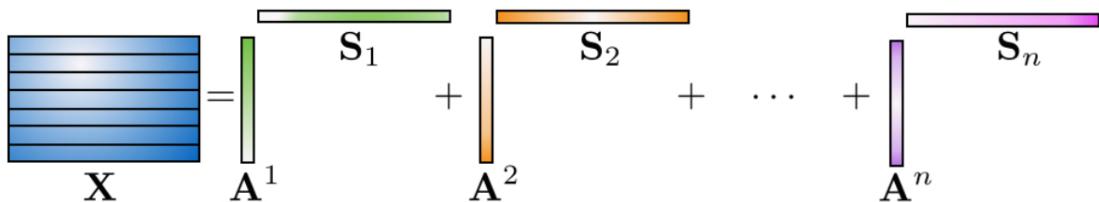


Figure 2.1.1: Illustration of the LMM structure from Ref [9].

During the whole work, the hypothesis that the number of observations is greater or equal than the number of sources will hold in order not to have an under-determined system.

Although the proposed model might be simplistic in some cases where the non-linearity of the data is known, it is still useful as it can be seen as a model for dimensionality reduction. For example, this

is the case for NMF, Non-negative Matrix Factorization, that can be considered a BSS problem with the prior of non negativity in the underlying matrices to estimate. Considering the dimensions of the data used, the input \mathbf{X} matrix has a $m \times t$ whereas the \mathbf{A} matrix $m \times n$ and the \mathbf{S} matrix $n \times t$. So in the matrix factorization we are going from a mt dimension to a $n(t + m)$ where the orders of magnitude can scale up to 10^9 for t , 10^4 for m and 10^1 for n . To end this example the dimension reduction is done when going from a data representation of 10^{13} dimensions to one of 10^{10} dimensions.

The formulation of the problem has many applications and the physical interpretation of the constituent elements vary for each one. In this work, the ultimate application will be to deal with radio-interferometric data that will add other constraints and difficulties.

2.1.3 APPLICATIONS FOR THE BSS

During this work we will be using *multivariate data*, that consist of several observations of the same process, phenomena, region, etc. with different but coherent points of view. The different observations and the techniques studied will allow the separation of the data in order to reconstruct the underlying structure following the LMM we will be using.

Nowadays there are many examples for the application of BSS methods on multivariate data. We will briefly describe three of them.

Estimation of the CMB map and foreground emissions

The CMB (Cosmological Microwave Background) is the oldest observable of our Universe, according to the Big-Bang theory, and is of much importance for cosmologist as it contains crucial information for the understanding of the origin and the evolution of the universe.

The Planck satellite was launched by the European Spatial Agency, operated from 2009 till 2013, with the mission of obtaining a more precise and more revealing CMB map. The satellite did multi-spectral observations in the microwave range, this accounts for observations of the sky at 9 different frequencies.

The observations issued from the mission are a mixture of the CMB map and different observable emissions from other phenomena like the synchrotron, free-free, thermal and spinning dust emissions. The recovery of a clear CMB map is done by the identification of the underlying contributions of each emission by a separation process, more precisely a Blind Source Separation.

Considering a LMM model as in the figure (2.1.2), each column of the mixing matrix (\mathbf{A}) is the frequency response or spectrum of a given source. Meanwhile, each line of the source matrix (\mathbf{S}) is the spatial distribution of the source in the observed sky. Although in this representation the sources are seen each one as an observation in 1D, they are in general a 2D image that has been flattened into a 1D array.

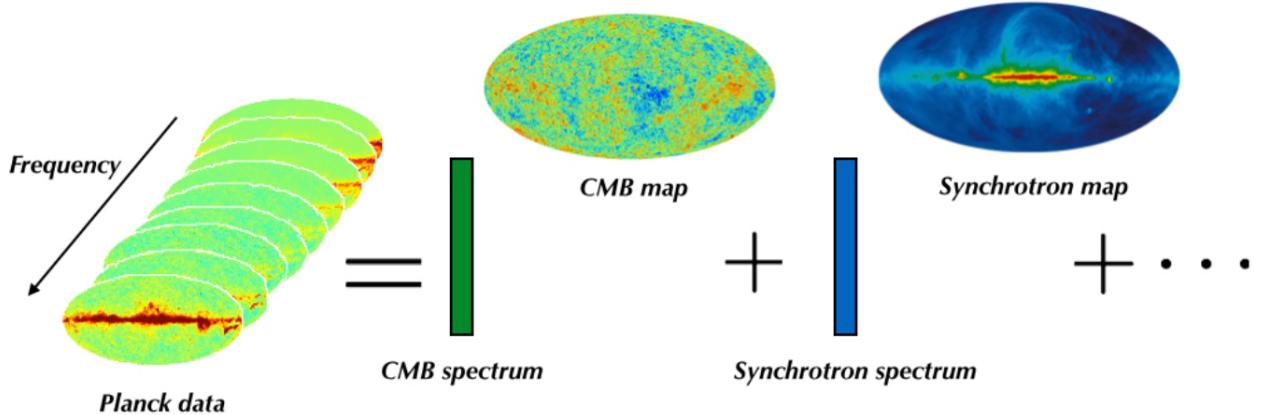


Figure 2.1.2: Illustration of the Planck data LMM model from Ref [9].

Hyperspectral unmixing

An application that has gained appreciation during the last years has been the hyperspectral unmixing. One of the main reasons is the fast development of hyperspectral sensors that can make measurements (around 100) at many frequencies in a defined spectral gap. The BSS has been of great help to exploit all the spectral information present in the observations and to move forward the field of remote sensing.

Considering a particular application, Mars Express is a space exploration mission conducted by the European Space Agency launched in 2003. The satellite is equipped with 128 channels OMEGA spectrometer from which BSS algorithms have been used to unmix the observations and identify solid H_2O and solid CO_2 in the red planet's surface as presented in the Ref [26] and in the figure (2.1.3).

LC/MS data analysis

Liquid Chromatography / Mass Spectrometry (LC/MS) is a powerful analytical chemistry technique that can identify and characterize many types of chemical compounds by coupling the two techniques together. LC helps to separate mixtures with multiple components while MS provides the structural characterization of these components with high sensitivity. The first gives the elution time and the second one the mass-to-charge ratio as it can be seen in the Figure (2.1.4).

There is a double separation taking place in the coupled method, a time separation thanks to the LC and a mass-to-charge ratio separation thanks to the MS. This shows the widespread of the

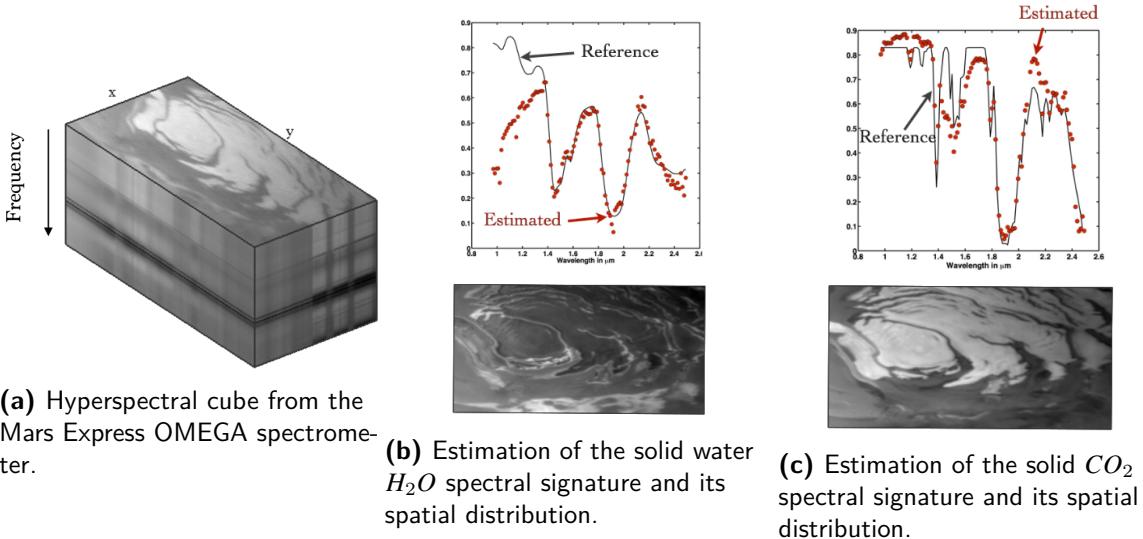


Figure 2.1.3: Illustration of the Mars Express data allowing to identify ice water and ice CO_2 from Ref [9] and Ref [26].

applications of these methods as in the precedent applications the separation was made in the spectral and spatial domain.

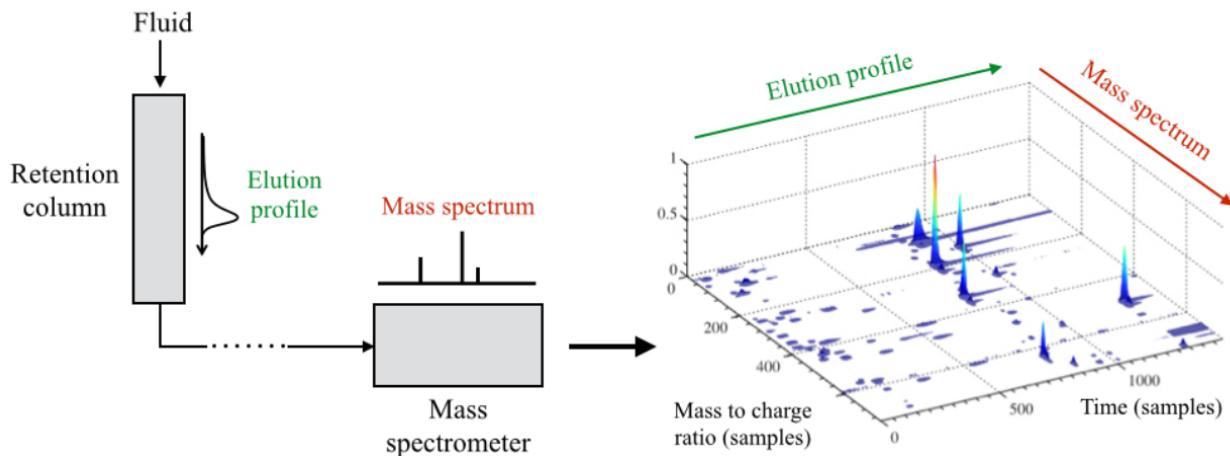
2.1.4 METHODS FOR BSS

The ill posedness of the inverse problem of estimating \mathbf{A} and \mathbf{S} from the equation (2.2) makes it a hard task to accomplish. There is an infinity of pairs $(\hat{\mathbf{A}}, \hat{\mathbf{S}})$ that verify the observations. This obliges each method to impose a prior knowledge to the problem or an hypothesis in order to have a criteria for choosing the desired solution.

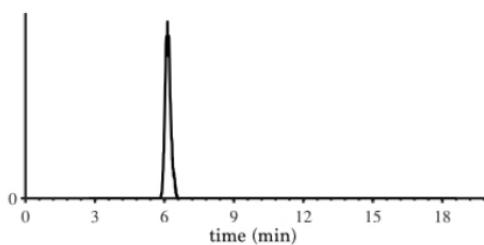
The choice of the criteria depends on the physical constraints of the problem and that is what is going to give sense and exploitability to the solution found. In this section the three main families of methods will be briefly described.

Independent Component Analysis (ICA)

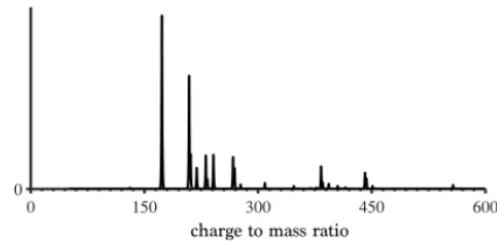
This family of methods is based upon the assumption that the unknown sources are statistically independent and they follow non Gaussian distributions. The method aims at estimating a unmixing matrix \mathbf{B} that will output estimated sources $\tilde{\mathbf{S}} = \mathbf{B}\mathbf{A}\mathbf{S}$ that are statistically independent and equal to the desired sources up to a permutation and a scale indeterminacy. One of the strong points of these methods, based on the Darmois-Linnik theorem, is that if the original sources are statistically independent, at most one of the sources has a Gaussian distribution and the estimated sources are



(a) Hyperspectral cube from the Mars Express OMEGA spectrometer.



(b) Estimation of the solid water H_2O spectral signature and its spatial distribution.



(c) Estimation of the solid CO_2 spectral signature and its spatial distribution.

Figure 2.1.4: Simple illustration of the LC/MS data from Ref [9] and Ref [10].

statistically independent, an unmixing matrix \mathbf{B} can be found that will yield the desired result up to a permutation and a scale factor.

The implementation of a constraint of statistical independence is not an easy task and this is a reason of the many approaches in the ICA based methods. There are several implementations, for example; the minimization of the mutual information using measures like the Kullback-Leibler divergence; and the maximization of the non-Gaussianity using the kurtosis.

An extensive description of these methods with their variants can be found in the Ref [27].

Non-negative Matrix Factorization (NMF)

This family of methods is based on the assumption of non-negativity of the mixing matrix \mathbf{A} and the sources \mathbf{S} . This assumption appears naturally in several type of problems like in LC/MS. Nevertheless, the assumption of non-negativity is in general not strong enough to deal with the ill posedness so it is in general coupled with other criteria like sparsity as in Ref [10].

An extensive description of these methods with their variants can be found in the Ref [28].

Sparse Blind Source Separation (SBSS)

The assumption in these methods is that the sources can be represented sparsely in a given dictionary. The assumption is not as strong as it could seem as the dictionary used for representing the signal can be overcomplete allowing a sparse representation for a wide variety of signals.

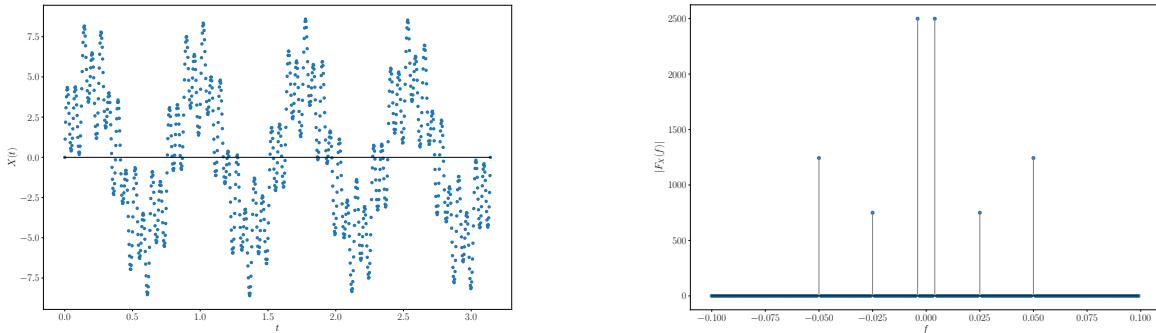
These type of methods will be deepened throughout this entire work and the References [1], [2] deal with the application of sparsity in the BSS problem.

2.2 SPARSITY

2.2.1 PRINCIPLES

Sparsity is the concept that a signal can be represented with a small number of non-zero coefficients. Centering us in the case of discrete signals of finite length, let $\mathbf{x} \in \mathbb{R}^{t \times 1}$, we will say that \mathbf{x} is *sparse* if only a small amount of its values is non-zero. In other words if the number of non-zero elements, $\|\mathbf{x}\|_0$, given by the ℓ_0 *pseudo-norm* happens to be much smaller than the length of the vector $\|\mathbf{x}\|_0 \ll t$.

The concept of sparsity is closely related to the way of representing a signal. An array can be sparse or not depending on the domain used for its representation. This is the case for a linear combination of sines as the one in the figure 2.2.1. The signal is not sparse in its direct domain as in the figure 2.2.1a but if the domain used for its representation is changed to Fourier as in the figure 2.2.1b the signal is said to be sparse.



(a) A linear combination of sines represented in the direct domain.

(b) A linear combination of sines represented in the Fourier domain.

Figure 2.2.1: Toy example to illustrate sparsity. Representing a linear combination of sines in the direct and the Fourier domain.

A signal that is represented by a few significant samples is said to be *approximatively sparse* even though the number of non-zero elements is low. This means that we can make a good approximation of the signal only using a small amount of coefficients.

To show a realistic example, an image of the Bubble Nebula cataloged as NGC 7635, seen in the figure 2.2.2a, taken from the Hubble Heritage team from the NASA will be used. This image is later used as one of the sources to be estimated by the developed algorithm. As it can be observed in the figure 2.2.2c, the plot of a flattened version of the image, the signal is not sparse. Nevertheless, if we change the domain of representation by using a wavelet transform, as the one used in the chapter 5 (*Starlets*: undecimated isotropic wavelets with 1 level of decomposition from Reference [19]), issuing the image in figure 2.2.2c, the coefficients of the first level of decomposition are approximatively sparse as observed in the figure 2.2.2d.

The figure 2.2.2e shows the energetic contribution of the samples when plotting the quotient between the energy of a vector composed of a given percentage of the largest samples and the energy of the whole signal. It can be seen that the coefficients in the wavelet domain assemble most of the energy in a small percentage of entries while in the natural image that is not the case as the energy is more evenly distributed.

2.2.2 SPARSE BLIND SOURCE SEPARATION

The way to proceed for estimating the sources \mathbf{S} and the mixing matrix \mathbf{A} will be a variational approach. This means that we will define an objective function in such way that the desired solution of the estimating problem will be its minimal value. In the section (3.3.3) why the solution of the proposed equation corresponds to the desired sources and mixing matrix.

To start with, the estimated matrices should describe the observations represented by the matrix \mathbf{X} so the problem can be formulated as follows:

$$\hat{\mathbf{A}}, \hat{\mathbf{S}} = \arg \min_{\mathbf{A}, \mathbf{S}} \frac{1}{2} \|\mathbf{X} - \mathbf{AS}\|_F^2 \quad (2.3)$$

where $\|\cdot\|_F$ is the Frobenius norm that defined as $\|\mathbf{A}\|_F^2 = \text{trace}(\mathbf{A}^T \mathbf{A})$. This problem can be also considered as a *matrix factorization problem* as we want to estimate a specific factorization of the \mathbf{X} matrix. The data fit term is chosen in order to cope with Gaussian noise as it is the case in the problem treated.

Sparsity plays its role as being a prior in the optimization problem that will help to regularize the problem. As it is ill-posed where there is an infinity of solutions that verify the equation 2.3. The sparsity prior will be imposed on the sources, and this is done by adding a sparsity inducing term in the objective function.

$$\hat{\mathbf{A}}, \hat{\mathbf{S}} = \arg \min_{\mathbf{A}, \mathbf{S}} \frac{1}{2} \|\mathbf{X} - \mathbf{AS}\|_F^2 + \|\Lambda \odot \mathbf{S}\|_p \quad (2.4)$$

with $p \in \{0, 1\}$, where Λ denotes the regularization parameters and \odot is the Hadamard product (element-wise multiplication). The Λ matrix allows to manage the weight given to each term, the data fit and the sparsity inducing term, in the optimization problem. The fact of using a matrix with the Hadamard product and not only a constant allows to specify different weight to each source giving more versatility to the objective function. The weight matrix is crucial as it can drastically change the solution of the optimization problem.

In this matrix factorization problem we have an amplitude indetermination as we are dealing with two elements multiplying. There can be a constant that is multiplying one of the matrices and the inverse of that constant multiplying the other matrix (for example $\hat{\mathbf{A}} = \alpha \mathbf{A}$, $\hat{\mathbf{S}} = \frac{1}{\alpha} \mathbf{S}$). To overcome this problem, that might degenerate the solution into one matrix with very high amplitudes and another one with very low, we will impose that the columns of the mixing matrix will have a unity ℓ_2 norm. This term will also give stability when optimizing the objective function.

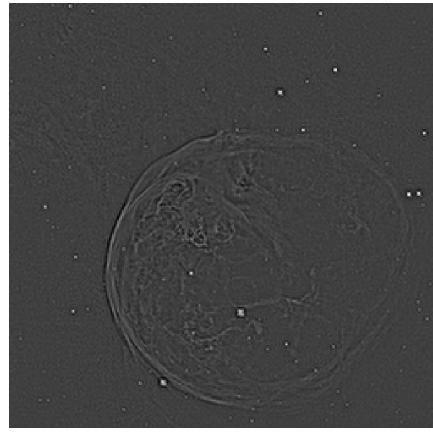
$$\hat{\mathbf{A}}, \hat{\mathbf{S}} = \arg \min_{\mathbf{A}, \mathbf{S}} \frac{1}{2} \|\mathbf{X} - \mathbf{AS}\|_F^2 + \left\| \Lambda \odot \mathbf{S} \Phi^{tr} \right\|_p + i_{\mathbf{Y}: \|\mathbf{Y}_k\|_2=1, \forall k}(\mathbf{A}) \quad (2.5)$$

where \mathbf{Y}_k denotes the k -th column of the matrix. The function $i_{\mathbf{Y}: \|\mathbf{Y}_k\|_2=1, \forall k}(\cdot)$ is an indicator function that returns zero when the constraint is met and infinity when it is not met. In general, unless otherwise stated it will be used $p = 0$. An interesting discussion on the choice of p can be found in [10].

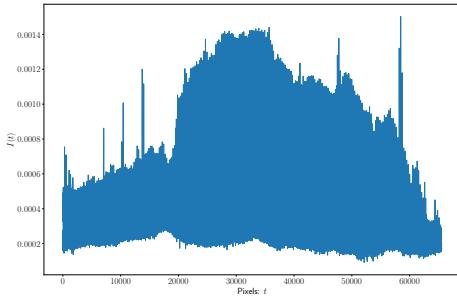
As the matrices we are trying to estimate are multiplying each other and as the two regularization terms are not differentiable we are facing a non-convex non-differentiable optimization problem. This means that the method used for the optimization will have an important impact on the solution found as it has to avoid local minima. The next chapter will deal with the choice of the optimization method to solve the variational problem of equation 2.5.



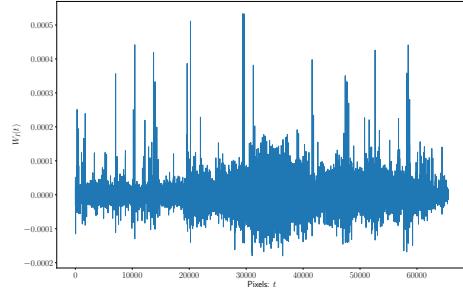
(a) One channel of the natural image of the Bubble Nebula (NGC 7635).



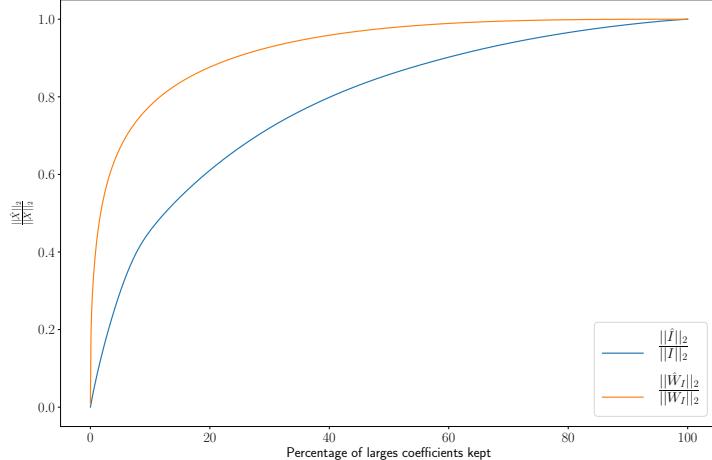
(b) The plot of a flattened version of the natural image of the Bubble Nebula. Not an approximatively sparse signal.



(c) The first level of wavelet decomposition of the Bubble Nebula image.



(d) The plot of a flattened version of the first level coefficients of the transformed image of the Bubble Nebula. An approximatively sparse signal.



(e) Energetic contribution of the samples in both representations.

Figure 2.2.2: Realistic example to illustrate a domain transformation to get an approximatively sparse signal. The image in the top left, when considered as a flattened vector is not an approximatively sparse signal. When a wavelet transform is used as in the right images the signal (the first level wavelet decomposition coefficients) becomes approximatively sparse in this new representation. The last plot illustrates the energetic contribution (with an unitary normalization) in function of the number of largest entries used (represented as a percentage). 14

“Hay hombres que de su ciencia
 Tienen la cabeza llena;
 Hay sabios de todas menas,
 Mas digo sin ser muy ducho
 Es mejor que aprender mucho
 El aprender cosas buenas.”

El gaucho Martín Fierro - José Hernández

3

Algorithmic Framework for Sparse Blind Source Separation

THE OPTIMIZATION METHOD EMPLOYED TO SOLVE THE PROBLEM OF EQUATION 2.5 IS CRUCIAL to determine the right sources as the objective function is non-convex. Nevertheless, it is *multi-convex*, that means that if we fix \mathbf{A} (resp. \mathbf{S}) and optimize with respect to \mathbf{S} (resp. \mathbf{A}) the optimization problem becomes convex. An extensive description of convex optimization can be found in Ref [29].

Another difficulty is added by the fact that the regularizing terms are not differentiable. This excludes the utilization of standard optimization methods such as the gradient descend. In order to cope with the optimization of this type of functions we will introduce proximal operators where a good description of them can be found in Ref [30].

Finally a description of the GMCA (Generalized Morphological Component Analysis) will be given, being this algorithm one of the basis from which we will be constructing the next chapter.

3.1 PROXIMAL ALGORITHMS

Before introducing the proximal operator we will remind some definitions. Let f be a function from \mathbb{R}^n to $\mathbb{R} \cup \{-\infty, +\infty\}$ we can define the following terms:

- **Effective domain:** It is the domain where the function do not reaches $-\infty$. It can be written as $\text{dom}(f) = \{\mathbf{x} \in \mathbb{R}^n : f(\mathbf{x}) < +\infty\}$.
- **Convex set:** A set $\mathcal{C} \subset \mathbb{R}^n$ is said to be convex if $\forall (\mathbf{x}, \mathbf{y}) \in \mathcal{C}^2$ and $\forall \lambda \in [0, 1]$, $\lambda \mathbf{x} + (1 - \lambda) \mathbf{y} \in \mathcal{C}$.
- **Convex function:** A function is said to be convex if its effective domain is convex and $\forall (\mathbf{x}, \mathbf{y}) \in \text{dom}(f)^2$ and $\forall \lambda \in [0, 1]$ we have the following inequality: $f(\lambda \mathbf{x} + (1 - \lambda) \mathbf{y}) \leq \lambda f(\mathbf{x}) + (1 - \lambda) f(\mathbf{y})$
- **Proper function:** A function is said to be proper if it does not reaches $-\infty$ which has a non empty effective domain.
- **Lower semi-continuous function:** A function is said to be lower semi-continuous if the ensembles $\{\mathbf{x} \in \mathbb{R}^n : f(\mathbf{x}) \leq a\}$ are closed for every $a \in \mathbb{R}$.

3.1.1 PROXIMAL OPERATOR

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a convex, proper and lower semi-continuous function. The functions that meet these conditions are said to be *proximable*.

The proximal operator $\text{prox}_f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is defined for proximable functions and is given by the following relation:

$$\text{prox}_f(\mathbf{x}) = \arg \min_{\mathbf{y}} \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 + f(\mathbf{y}) \quad (3.1)$$

where f is a proximable function. This operator can be seen a local minimizer of the function f in the neighborhood of \mathbf{x} .

There are some functions that allow an explicit formula for the operator while other ones require an iterative optimization algorithm. Some usual functions used as regularization terms admit an explicit solution.

Proximal operator of sparsity inducing terms

The sparsity inducing term with norm ℓ_1 admits the following operator which is named *soft thresholding*:

$$g_1(\mathbf{s}) = \lambda \|\mathbf{s}\|_1, \quad \text{prox}_{g_1}(\mathbf{x}) = \text{sign}(\mathbf{x}) \odot \max(0, |\mathbf{x}| - \lambda) \quad (3.2)$$

Another possibility for the sparsity inducing term is to use the pseudo-norm ℓ_0 that can be defined as follows:

$$\|\mathbf{x}\|_0 = \text{card}(\text{supp}(\mathbf{x})) \quad (3.3)$$

Its proximal operator can be seen as the hard thresholding, defined in equation 3.4, even though if the pseudo-norm ℓ_0 is not strictly a *proximable* as it is not convex.

$$g_0(\mathbf{s}) = \lambda \|\mathbf{s}\|_0, \text{ prox}_{g_0}(\mathbf{x}) = \begin{cases} 0 & : |x_i| < \lambda \\ x_i & : \text{otherwise} \end{cases}, \forall i = 1, \dots, n \quad (3.4)$$

The illustration of both thresholdings can be seen in the figure 3.1.1.

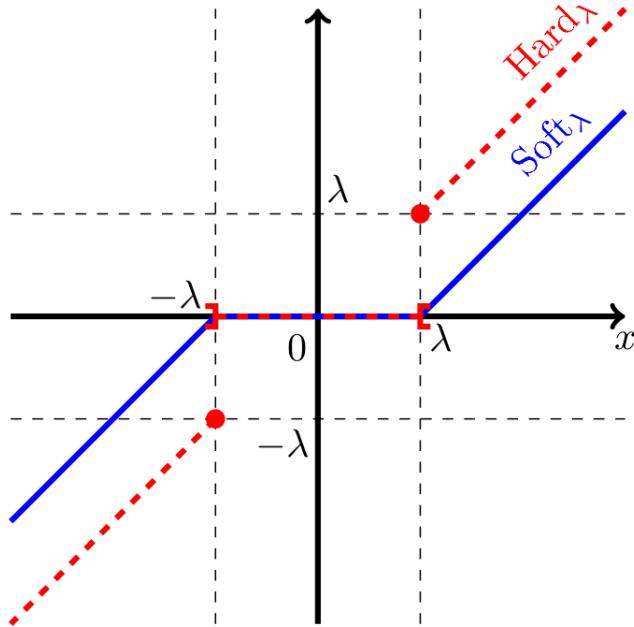


Figure 3.1.1: Hard and Soft thresholdings illustration from Ref [10].

Proximal operator of indicator functions

The indicator functions are defined as follows:

$$i_{\mathcal{C}} : \mathbf{x} \rightarrow \begin{cases} 0 & : \mathbf{x} \in \mathcal{C} \\ +\infty & : \text{otherwise} \end{cases} \quad (3.5)$$

where \mathcal{C} is the set where the function is defined on.

An interesting case is when the set \mathcal{C} of the indicator function is a convex as the function becomes *proximable*. The proximal operator has an explicit formulation and it is simply the orthogonal projection

of the vector into the convex set \mathcal{C} . For example, the proximal operator of the indicator function of the vectors of \mathbb{R}^n having unit norm ℓ_2 is simply the normalization of the vector as follows:

$$\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_2 = 1\} \quad (3.6a)$$

$$\text{prox}_{i_{\mathcal{C}}}(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|_2} \quad (3.6b)$$

3.2 BLOCK COORDINATE ALGORITHMS

The sensibility and the non convexity of the previously defined objective function 2.5 gives the optimization algorithm an important role to play in the BSS problem solving.

Its characteristics makes the *Block Coordinate methods* (BC) a natural choice to deal with the optimization problem as it is adapted for *block multi-convex* problems. The general objective function used by the BC is the following one:

$$h(\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n) = f(\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n) + \sum_{i=1}^n g_i(\mathbf{X}_i) \quad (3.7)$$

where we the goal is to estimate the n real blocks $\{\mathbf{X}_i\}_{i=1,\dots,n}$ with $\mathbf{X}_i \in \mathbb{R}^{m_i \times t_i}$. When making the link to the BSS problem and the objective function defined in the equation (2.5) it is possible to identify the f function in equation (3.7) as the data fit term and the functions $\{g_i\}_{i=1,\dots,n}$ as the different regularizing terms.

Let \mathbf{X}_i^k denote the value of \mathbf{X}_i after its k th update. What characterizes the BC methods is that they optimize the function h by cyclically minimizing it with respect to each block (\mathbf{X}_i) while fixing the remaining ones $(\mathbf{X}_1^k, \dots, \mathbf{X}_{i-1}^k, \mathbf{X}_{i+1}^{k-1}, \dots, \mathbf{X}_n^{k-1})$ at their last updated value. Let

$$f_i^k(\mathbf{X}_i) \triangleq f(\mathbf{X}_1^k, \dots, \mathbf{X}_{i-1}^k, \mathbf{X}_i, \mathbf{X}_{i+1}^{k-1}, \dots, \mathbf{X}_n^{k-1}) \quad \forall i, \forall k. \quad (3.8)$$

Block Coordinate Descend

The basic Block Coordinate Descend (BCD) method consists of solving exactly the subproblem:

$$\mathbf{X}_i^k = \arg \min_{\mathbf{X}_i} f_i^k(\mathbf{X}_i) + g_i(\mathbf{X}_i) \quad (3.9)$$

for each block as presented in algorithm (4). There are several strategies for solving each subproblem as it has been studied in the Reference [11]. This algorithm can be proved to converge

towards a stationary point of the original cost function h under mild conditions in References [11] and [15] (Theorem 4.1).

The Figure 3.2.1 illustrates with a toy example the optimization path of a BCD strategy to solve a simple function.

Algorithm 4 Block Coordinate Descend

```

1: procedure BCD( $\mathbf{X}_1^0, \mathbf{X}_2^0, \dots, \mathbf{X}_n^0$ )
2:   while do not converge do
3:     for  $i = 1, \dots, n$  do
4:        $\mathbf{X}_i^k \leftarrow \arg \min_{\mathbf{x}_i} f_i^k(\mathbf{X}_i) + g_i(\mathbf{X}_i)$ 
5:      $k \leftarrow k + 1$ 
6:   return  $\mathbf{X}_1^k, \mathbf{X}_2^k, \dots, \mathbf{X}_n^k$ 
```

There are many variants of the BC algorithms where they main difference resides on the subproblem (eq. (3.9)) solving technique. In many practical cases we are facing a non differentiable convex problem that is adapted for proximal optimization techniques as we are going to see with the next two algorithms PALS¹ and PALM². It is important to note that it is not always in the best interest to solve exactly each subproblem (eq. (3.9)) as the minimization is done over only one variable and not on the original objective function. In other words, a big optimization step on one variable can, depending on the problem faced, take us away from the optimal optimization path.

Proximal Alternating Linearized Minimization

The PALM method aims to solve non-convex non-smooth problems which is exactly the case for the h cost function in equation (3.7) as the f function is non-convex and the $\{g_i\}_{i=1,\dots,n}$ functions are non-smooth considering the sparsity inducing term. The algorithm is studied in the Reference [14] with a proof of its convergence to a critical point under some conditions (Theorem 3.1).

It consists of updating alternatively each block as in BC methods. Each subproblem update is done by minimizing a prox-linearization. This accounts to do a gradient step over the smooth f function and then applying the proximal operator of the non-smooth function g_i as it can be seen in the algorithm

¹Proximal Alternating Least Squares

²Proximal Alternating Linearized Minimization

³Generated using the code of: Nicoguaro (https://commons.wikimedia.org/wiki/File:Coordinate_descent.svg), <https://creativecommons.org/licenses/by/4.0/legalcode>

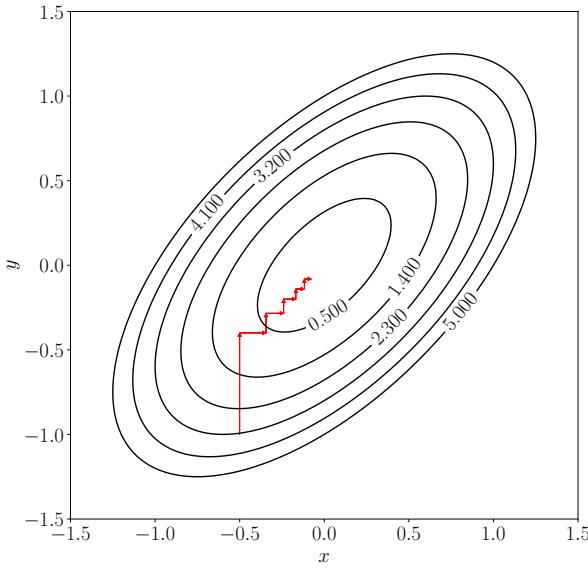


Figure 3.2.1: Illustration to show the optimization steps (in red) over each variable (x and y) in the BC type of optimization.³

(5). The subproblem update can be written as follows:

$$\tilde{\mathbf{X}}_i^{k+1} = \mathbf{X}_i^k - \frac{1}{L_k^i} \nabla f_i^k (\mathbf{X}_i^k) \quad (3.10a)$$

$$\mathbf{X}_i^{k+1} = \text{prox}_{\frac{1}{L_k^i} g_i} (\tilde{\mathbf{X}}_i^{k+1}) \quad (3.10b)$$

where $\frac{1}{L_k^i}$ denotes a constant for which we will take the Lipschitz constant of ∇f_i^k .

Algorithm 5 PALM algorithm

```

1: procedure PALM( $\mathbf{X}_1^0, \mathbf{X}_2^0, \dots, \mathbf{X}_n^0$ )
2:   while do not converge do
3:     for  $i = 1, \dots, n$  do
4:       Compute the Lipschitz constant  $\frac{1}{L_k^i}$  of  $\nabla f_i^k$ 
5:        $\mathbf{X}_i^{k+1} \leftarrow \text{prox}_{\frac{1}{L_k^i} g_i} \left( \mathbf{X}_i^k - \frac{1}{L_k^i} \nabla f_i^k (\mathbf{X}_i^k) \right)$ 
6:      $k \leftarrow k + 1$ 
7:   return  $\mathbf{X}_1^k, \mathbf{X}_2^k, \dots, \mathbf{X}_n^k$ 

```

Projected Alternating Least Squares

The last method introduced updates each component in a two step procedure. The first one consists of solving the minimization exactly but for only the differentiable term that in the notation of equation (3.9) would be $f_i^k(\mathbf{X}_i)$ and the result of this optimization is used as an input for the minimization of the non differentiable function $g_i(\mathbf{X}_i)$ using a proximal operator. This algorithm is mostly interesting when the proximal operators have an explicit formulation as it makes it simple and fast. A general formulation of the algorithm can be found in algorithm (6).

The method was firstly introduced in the context of NMF in Reference [12] and has not yet any proof of convergence. The minimization of each subproblem (equation 3.9) is not properly done. The sequential minimization of each one of its terms does not guarantee the minimization of the entire cost function.

Nevertheless, it is still an interesting algorithm due to its simplicity and rapidity and it is the base of the optimization algorithm of the GMCA (Generalized Morphological Component Analysis) method for BSS.

The name least squares from PALS comes from the fact that in many cases the function f optimized in the first step is a quadratic loss term using the ℓ_2 norm or the Frobenius norm. The minimization of this function is done with a least square estimation using the Moore-Penrose pseudoinverse. The projected part comes from the fact that the least square solution is projected on to the constraint convex set which is the proximal operator of the indicator function over a convex set.

Algorithm 6 PALS algorithm

```
1: procedure PALS( $\mathbf{X}_1^0, \mathbf{X}_2^0, \dots, \mathbf{X}_n^0$ )
2:   for  $k = 1, \dots, K$  do
3:     for  $i = 1, \dots, n$  do
4:        $\mathbf{X}_i^k \leftarrow \arg \min_{\mathbf{x}_i} f_i^k(\mathbf{X}_i)$ 
5:        $\mathbf{X}_i^k \leftarrow \text{prox}_{g_i}(\mathbf{X}_i^k)$ 
6:   return  $\mathbf{X}_1^k, \mathbf{X}_2^k, \dots, \mathbf{X}_n^k$ 
```

3.3 GENERALIZED MORPHOLOGICAL COMPONENT ANALYSIS

The Generalized Morphological Component Analysis (GMCA) is a PALS-optimization-based method that aims to robustly solve a BSS problem. It was developed in References [1] and [2], and its study continued in [4], [9] and [10]. The separation power of the algorithm is founded on the *sparsity and the morphological diversity* of the sources in contrast to the statistical independence used in the ICA method.

3.3.1 MORPHOLOGICAL DIVERSITY

Morphological diversity accounts for the concept where an image (or more general a signal) is modeled as a sum of components where each one has a morphological signature. This means that each component is more sparsely represented in a specific dictionary.

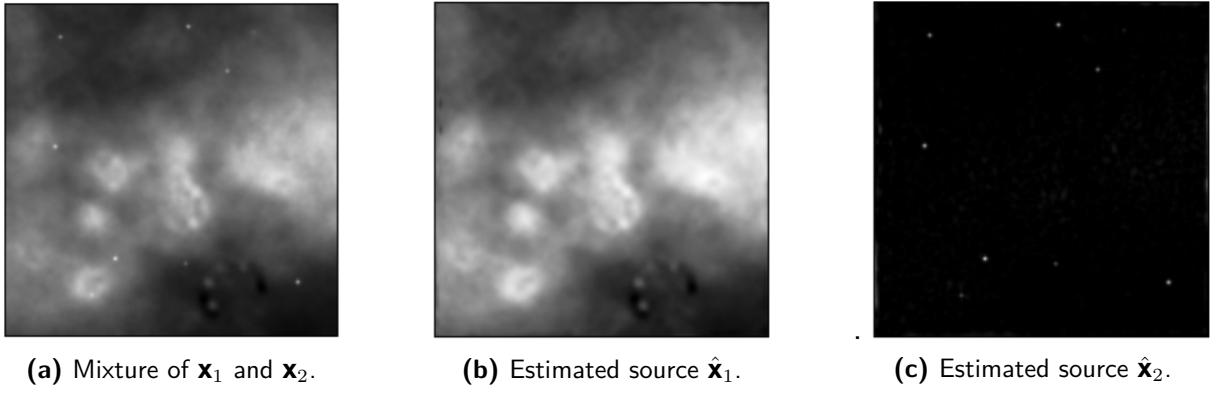


Figure 3.3.1: Toy example illustrating how the MDP helps to separate the source in the BSS problem.

In the mixture in figure 3.3.1a the two sources have different morphologies, thus respecting the Morphological Diversity Principle (MDP). The point source emission in figure (3.3.1c) has already a sparse representation in the direct domain whereas the emission in figure (3.3.1b) is not sparsely represented as it is distributed all over the image. Nevertheless, it can be sparsely distributed in a specific wavelet domain. The fact that each source is more sparsely represented in its own dictionary makes them respect the MDP, thus showing diversity in their morphologies.

3.3.2 MORPHOLOGICAL COMPONENT ANALYSIS

The Morphological Component Analysis (MCA) is a technique of BSS to retrieve the components \mathbf{x}_1 and \mathbf{x}_2 from the mixture \mathbf{x} by looking for their sparsest decompositions. The theoretical support of the foud solution is provided always as long as the mixture respects the MDP. In that case, the sparsest decompositions will correspond to the components looked for. The MCA algorithm intends to solve the following optimization problem:

$$\operatorname{argmin}_{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n} \left\| \mathbf{x} - \sum_{i=1}^n \mathbf{x}_i \right\|_2^2 + \sum_{i=1}^n \lambda_i \left\| \mathbf{x}_i \Phi_i^T \right\|_0 \quad (3.11)$$

where the Φ_i^T is the analysis operator of a specific dictionary. The equation (3.11) allows to identify firstly a data fidelity term and then a sparsity inducing term. Further information about this method can be found in [3]. Some clarifying words from *Jérôme Bobin* in [1]:

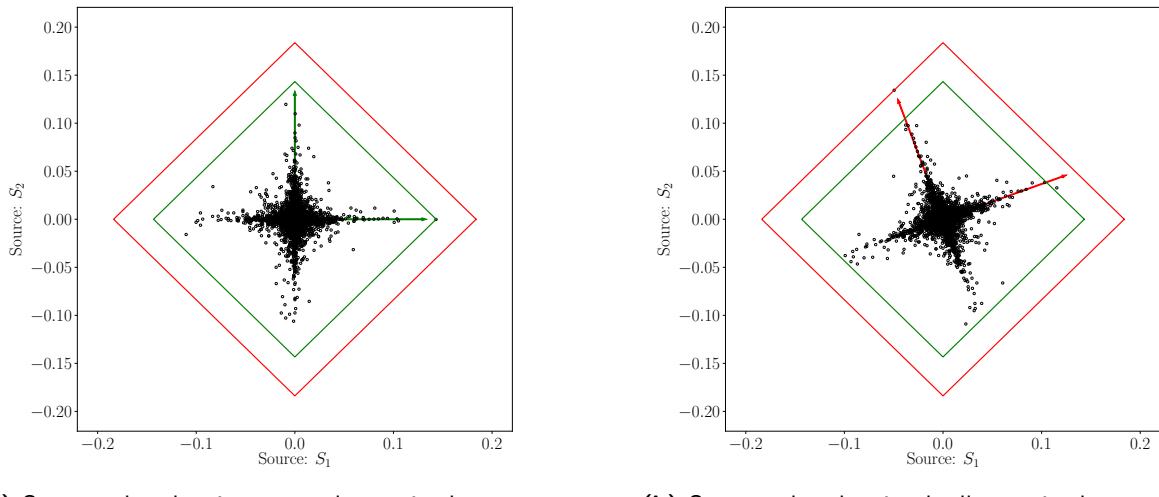
"In other words, MCA then relies on the incoherence between the subdictionaries to estimate the morphological components".

3.3.3 GENERALIZED MORPHOLOGICAL COMPONENT ANALYSIS

The dictionaries used to represent the data have to be known *a priori* in the described setup. It can be interesting to create a new dictionary $\mathcal{D} = [\Phi_1 \Phi_2 \dots \Phi_m]$ made up from the concatenation of many different dictionaries in order to allow the sparse representation of an important number of signals.

It is possible to extend the MDP to the case where the signals are represented in one dictionary and exploit it for tackling the BSS problem. The new dictionary \mathcal{D} will allow representations of the sources with their most representative active entries active at different positions if the MDP is respected. The disjoint supports of the signal representations can be exploited to separate them. This fact allows to identify the sources from the mixture by looking for the sparsest representation of the data, thus imposing sparsity. This is the basic concept of the GMCA as if the sources are not correctly unmixed they will not display their sparsest representation and it will exist another unmixing matrix allowing sparsest sources maintaining the fidelity on the observed data.

The toy example in figure 3.3.2 shows how the smallest ℓ_1 ball indicates the good unmixing of the sources. The arrows indicate the remaining mixing matrix present after unmixing. The left figure (3.3.2a) arrows are identical with the axes whereas in the right figure (3.3.2b) it is possible to see that the arrows are not aligned with the figure axes showing an incorrect unmixing.



(a) Scatter plot showing correctly unmixed sources.

(b) Scatter plot showing badly unmixed sources.

Figure 3.3.2: Scatter plot of the approximatively sparse sources S_1 against S_2 illustrating the ℓ_1 ball for the correctly unmixed sources (green line) and the one for the badly unmixed sources (red line).

The previous scatter plots in figure (3.3.2) show the morphological diversity, as the more the shape of a + sign the easier to separate the sources and the more morphologically diverse they are. This means that when the coefficient of one source is active, have a high amplitude, the coefficients of the other sources are inactive, have a small amplitude, and *vice-versa*. The separation problem amounts to align the mixing matrix (\mathbf{A}) columns to each one of the axis of the scatter plot. When the diversity principle is hold, the solution with the most sparse representation will coincide with the one of the real sources. It exist a difficult scenario when different sources are active at the same time, not respecting the principle, and therefore the most sparse solution will not correspond with the real sources. The work [4] deals with a method to overcome with spatially correlated sources in the GMCA that will be used afterwards in this work.

Optimization method

Optimizing the problem (2.5) is not an easy task as it is globally non-convex and will make the solution depend strongly on the initialization point and the algorithm used to optimize. Nevertheless, the problem is multi-convex, that means that if we fix one of the variables \mathbf{A} or \mathbf{S} and optimize the problem with respect to the other one we are in a convex optimization problem. The GMCA optimization scheme is based on the PALS procedure specified in the algorithm (5). The basis steps of GMCA are presented in the algorithm (7).

Algorithm 7 [Simple Version] GMCA

```

1: procedure GMCA( $\mathbf{X}$ , parameters)
2:   Initialize :  $\hat{\mathbf{A}}, \hat{\mathbf{S}}$ 
3:
4:   for  $k = 0, \dots, N_{epoch}$  do
5:     (I) fit term:  $\tilde{\mathbf{S}}_{(k+1)} \leftarrow \arg \min_{\mathbf{S}} \frac{1}{2} \|\mathbf{X} - \hat{\mathbf{A}}_{(k)} \mathbf{S}\|_F^2$ 
6:
7:     (II) Reg. term:  $\hat{\mathbf{S}}_{(k+1)} \leftarrow \text{prox}_{(\|\Lambda \odot \mathbf{S}\|_p)}(\tilde{\mathbf{S}}_{(k+1)})$ 
8:
9:     (III) Data fit term:  $\tilde{\mathbf{A}}_{(k+1)} \leftarrow \arg \min_{\mathbf{A}} \frac{1}{2} \|\mathbf{X} - \mathbf{A} \hat{\mathbf{S}}_{(k+1)}\|_F^2$ 
10:    (IV) Reg. term:  $\hat{\mathbf{A}}_{(k+1)} \leftarrow \text{prox}_{i_{\mathbf{Y}: \|\mathbf{Y}^k\|_2=1, \forall k}(\mathbf{A})}(\tilde{\mathbf{A}}_{(k+1)})$ 
11:   return  $\mathbf{A}, \mathbf{S}$ 

```

First, the optimization is done over \mathbf{S} , where there are two terms, the data fidelity term and the regularization term that is imposing sparsity. The first term being convex and differentiable, the optimization is done using the pseudo-inverse as in a least square problem. The proximal operator is used to optimize the regularization term as it is convex but non differentiable. The operator has an

explicit solution as it can be seen in the algorithm (8) with the thresholding operator. Depending on the norm used to enforce the sparsity in the sources the thresholding will be a soft (norm ℓ_1) or a hard thresholding (pseudo-norm ℓ_0) as in the figure (3.1.1). The optimization in the data fidelity term over \mathbf{A} is also solved using the pseudo-inverse. The proximal operator of the indicator function has also an explicit solution and is simply the projection into the ℓ_2 ball as it can be seen in the algorithm (8).

Algorithm 8 [Algorithmic version] GMCA

```

1: procedure GMCA( $\mathbf{X}$ , parameters)
2:   Initialize :  $\hat{\mathbf{A}}, \hat{\mathbf{S}}$ 
3:
4:   for  $k = 0, \dots, N_{epoch}$  do
5:     Estimation:  $\hat{\mathbf{S}}_{(k+1)} \leftarrow \hat{\mathbf{A}}_{(k)}^\dagger \mathbf{X}$ 
6:     Thresholding:  $\hat{\mathbf{S}}_{(k+1)} \leftarrow Th_\Lambda(\tilde{\mathbf{S}}_{(k+1)})$ 
7:     Estimation:  $\hat{\mathbf{A}}_{(k+1)} \leftarrow \mathbf{X} \hat{\mathbf{S}}_{(k+1)}^\dagger$ 
8:     Projection:  $\hat{\mathbf{A}}_{(k+1)}^j \leftarrow \frac{\hat{\mathbf{A}}_{(k+1)}^j}{\|\hat{\mathbf{A}}_{(k+1)}^j\|_2}, \forall j \in \{1, \dots, N_{cols}\}$ 
9:   return  $\mathbf{A}, \mathbf{S}$ 

```

Choice of the regularization parameter

The parameter Λ is essential in the optimization problem as its result will depend strongly on it. The authors in [1] propose an adaptive thresholding strategy where the parameter should have a high value at first and then go relaxing during the iterations until it converges into an ultimate value. This ultimate value should depend on the value of the noise so it is based on the mad (Median Absolute Deviation) estimator from equation (3.12). It is a robust estimator of the noise standard deviation that can be calculated as $\sigma_N \approx 1.48 \text{ mad}(\mathbf{x})$. In practice, the final value is fixed to $3\sigma_N$ for each source estimated using the mad. This strategy is based on the idea of using the entries with high amplitude at first as they are the most discriminant ones and then go taking in more entries to get a more precise estimation. The final thresholding value serves for denoising.

$$\text{mad}(\mathbf{x}) = \text{median}_i(|x_i - \text{median}_i(x_i)|) \quad (3.12)$$

*“Mi gloria es vivir tan libre,
Como el pájaro del cielo,
No hago nido en este suelo,
Ande hay tanto que sufrir;
Y naides me ha de seguir
Cuando yo remuento el vuelo.”*

El gaucho Martín Fierro - José Hernández

4

Distributed Sparse Blind Source Separation

ASTROPHYSICS DID NOT ESCAPE FROM THE “DATA BOOM” experienced over the last years throughout the most diverse disciplines. In this chapter a novel method developed for Blind Source Separation in the presence of vast amounts of data will be presented, described and analyzed. The basis of the method is to segment the *big* problem into *small* problems resembling a *divide and conquer* approach. There is a new problem arising with this choice, the different subproblems will issue each one estimators of the same variable and hard-to-tackle task will be to combine them into one robust estimator. The key element of the method is the aggregation of estimators that is done using a weighted *Fréchet Mean* calculated as an optimization problem on a Riemannian manifold which in our case will be the unit sphere \mathbb{S}^{m-1} as we are imposing the unit ℓ_2 norm for the columns of the mixing matrix.

4.1 MOTIVATION

Distributed BSS accounts for solving a large scale problem where the data involved is so massive that cannot be held in memory. As this is becoming the new standard with new space telescopes

such as Euclid¹, radio-interferometers such as SKA² and the explosion of data, it is natural to take the development path of a method capable of both:

- Solving a BSS problem where the data can not be accessed entirely at once but only to scan parts of it and keeping up the separation performance
- Speed up the resolution of the heavy calculations involved by the use of distributed architectures.

In order to have an idea of the magnitudes, the number of pixels are approx. 10^9 and the number of channels approx. 10^4 having in general a small number of sources approx. 10^1 .

4.1.1 STATE-OF-THE-ART METHODS FOR BATCH MATRIX FACTORIZATION

Given the data boom in the last years and the subsequent rise of machine learning algorithms capable of coping with such orders of magnitude some attractive articles were written. However, in general the methods do not tackle the large-scale SBSS problem specifically and are not simply adapted to our case.

Distributed Optimization via the Alternating Direction Method of Multipliers

The well-known article [31] from S. Boyd gives a powerful framework for performing distributed optimization. In fact, it is possible to develop a distributed algorithm to solve the blind source separation problem from the defined cost function of equation (2.5).

The ADMM intends to blend the decomposability of the dual ascend method with the convergence properties of the method of multipliers. The algorithm allows to split variables by the addition of a dual variable which will be updated at each epoch in a way so that the convergence of the algorithm to a saddle point can be guaranteed under some conditions.

The problem addressed by the ADMM has the following form:

$$\begin{aligned} & \text{minimize } f(x) + g(z) \\ & \text{subject to } \mathbf{Ax} + \mathbf{Bz} = c \end{aligned} \tag{4.1}$$

where $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{p \times n}$, $\mathbf{B} \in \mathbb{R}^{p \times m}$ and $c \in \mathbb{R}^p$. The x variable has been splitted into x and z , modifying the objective function and the constraint. In the so called *scaled form* of the problem,

¹An ESA mission to map the geometry of the dark universe by the investigation of the distance-redshift relationship and the evolution of cosmic structures. It will cover the entire period over which dark energy played a significant role in accelerating the expansion.

²Square Kilometer Array, the world's largest radio telescope with physical presence in South Africa and Australia; and eventually with over a square kilometre (one million square metres) of collecting area.

where ρ is a parameter to be defined, the algorithm can be expressed as:

$$\begin{aligned} x^{(k+1)} &:= \arg \min_x \left(f(x) + \frac{\rho}{2} \left\| \mathbf{A}x + \mathbf{B}z^{(k)} - c + u^{(k)} \right\|_2^2 \right) \\ z^{(k+1)} &:= \arg \min_z \left(g(z) + \frac{\rho}{2} \left\| \mathbf{A}x^{(k+1)} + \mathbf{B}z - c + u^{(k)} \right\|_2^2 \right) \\ u^{(k+1)} &:= u^{(k)} + \mathbf{A}x^{(k)} + \mathbf{B}z^{(k)} - c \end{aligned} \quad (4.2)$$

The framework can be adapted to the objective function of equation (2.5). Calling $g_\Lambda(\cdot)$ and $h(\cdot)$, the sparsity inducing term and the mixing matrix constraint term respectively we can define the following problem:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2} \|\mathbf{X} - \mathbf{C}\|_F^2 + g_\Lambda(\mathbf{D}) + h(\mathbf{B}) \\ &\text{subject to} \quad \mathbf{B} - \mathbf{A} = 0 \\ &\quad \mathbf{C} - \mathbf{A}\mathbf{S} = 0 \\ &\quad \mathbf{D} - \mathbf{S} = 0 \end{aligned} \quad (4.3)$$

The updates can be expressed as:

$$\begin{aligned} \mathbf{S}^{(k+1)} &:= \arg \min_{\mathbf{S}} \left(\frac{\rho \mathbf{S}}{2} \left\| \mathbf{C}^{(k)} - \mathbf{A}^{(k)}\mathbf{S} + \mathbf{D}^{(k)} - \mathbf{S} + \mathbf{B}^{(k)} - \mathbf{A}^{(k)} + \mathbf{U}_C^{(k)} + \mathbf{U}_D^{(k)} + \mathbf{U}_B^{(k)} \right\|_F^2 \right) \\ \mathbf{C}^{(k+1)} &:= \arg \min_{\mathbf{C}} \left(\frac{1}{2} \|\mathbf{X} - \mathbf{C}\|_F^2 + \frac{\rho \mathbf{C}}{2} \left\| \mathbf{C} - \mathbf{A}^{(k)}\mathbf{S}^{(k+1)} + \mathbf{D}^{(k)} - \mathbf{S}^{(k+1)} + \mathbf{B}^{(k)} - \mathbf{A}^{(k)} + \mathbf{U}_C^{(k)} + \mathbf{U}_D^{(k)} + \mathbf{U}_B^{(k)} \right\|_F^2 \right) \\ \mathbf{D}^{(k+1)} &:= \arg \min_{\mathbf{D}} \left(g_\Lambda(\mathbf{D}) + \frac{\rho \mathbf{D}}{2} \left\| \mathbf{C}^{(k+1)} - \mathbf{A}^{(k)}\mathbf{S}^{(k+1)} + \mathbf{D} - \mathbf{S}^{(k+1)} + \mathbf{B}^{(k)} - \mathbf{A}^{(k)} + \mathbf{U}_C^{(k)} + \mathbf{U}_D^{(k)} + \mathbf{U}_B^{(k)} \right\|_F^2 \right) \\ \mathbf{B}^{(k+1)} &:= \arg \min_{\mathbf{B}} \left(h(\mathbf{B}) + \frac{\rho \mathbf{B}}{2} \left\| \mathbf{C}^{(k+1)} - \mathbf{A}^{(k)}\mathbf{S}^{(k+1)} + \mathbf{D}^{(k+1)} - \mathbf{S}^{(k+1)} + \mathbf{B} - \mathbf{A}^{(k)} + \mathbf{U}_C^{(k)} + \mathbf{U}_D^{(k)} + \mathbf{U}_B^{(k)} \right\|_F^2 \right) \\ \mathbf{A}^{(k+1)} &:= \arg \min_{\mathbf{A}} \left(\frac{\rho \mathbf{A}}{2} \left\| \mathbf{C}^{(k+1)} - \mathbf{A}\mathbf{S}^{(k+1)} + \mathbf{D}^{(k+1)} - \mathbf{S}^{(k+1)} + \mathbf{B}^{(k+1)} - \mathbf{A} + \mathbf{U}_C^{(k)} + \mathbf{U}_D^{(k)} + \mathbf{U}_B^{(k)} \right\|_F^2 \right) \\ \mathbf{U}_B^{(k+1)} &:= \mathbf{U}_B^{(k)} + \mathbf{B}^{(k+1)} - \mathbf{A}^{(k+1)} \\ \mathbf{U}_C^{(k+1)} &:= \mathbf{U}_C^{(k)} + \mathbf{C}^{(k+1)} - \mathbf{A}^{(k+1)}\mathbf{S}^{(k+1)} \\ \mathbf{U}_D^{(k+1)} &:= \mathbf{U}_D^{(k)} + \mathbf{D}^{(k+1)} - \mathbf{S}^{(k+1)} \end{aligned} \quad (4.4)$$

The usefulness of this method resides on the splitting of the source matrix optimization in the data

fidelity term and the sparsity inducing term while having explicit proximal operators. That is the motivation of the definition of the new \mathbf{D} , \mathbf{C} and \mathbf{B} variables. This formulation of the ADMM optimization allows explicit formulations of the various minimizations allowing as well a parallelization scheme.

Nevertheless, not everything is rosy, this approach suffers some drawbacks which are the cause of its non-use. The algorithm is very sensible to the choice of parameters, namely the different ρ and Λ . They depend on many factors like the initial point, thus giving diverse results. The way they are found in the equations make it hard to deduce heuristics in order to give the algorithm robustness. Although it is an interesting method it lacks robustness for its practical use as it is one of the essential characteristics ordered to the algorithm.

Online Learning for Matrix Factorization

Mairal proposes an online method for matrix factorization, mostly dictionary learning, in [32]. The method calculates a dictionary \mathbf{D}_T that allows the sparsest representation α_t of a series of observations \mathbf{x}_t . The method does not require to store all the observations and their sparse representation, but only two matrices, that will store the necessary statistics, being $\mathbf{A}_t = \sum_{i=1}^t \alpha_i \alpha_i^{tr}$ and $\mathbf{B}_t = \sum_{i=1}^t \mathbf{x}_i \alpha_i^{tr}$.

Both matrices are updated each time a new observation arrives. Afterwards the dictionary \mathbf{D}_t is computed using \mathbf{D}_{t-1} as a warm restart. The algorithm performing the dictionary update does it column by column independently for each iteration until convergence where the cost function being optimized is the following one:

$$\begin{aligned}\mathbf{D}_t &= \arg \min_{\mathbf{D} \in \mathcal{C}} \frac{1}{t} \sum_{i=1}^t \left(\frac{1}{2} \|\mathbf{x}_i - \mathbf{D}\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1 \right) \\ &= \arg \min_{\mathbf{D} \in \mathcal{C}} \frac{1}{t} \left(\frac{1}{2} \text{Tr}(\mathbf{D}^{tr} \mathbf{D} \mathbf{A}_t) - \text{Tr}(\mathbf{D}^{tr} \mathbf{B}_t) \right)\end{aligned}\tag{4.5}$$

where $\text{Tr}(\cdot)$ is the matrix trace operator and \mathcal{C} is a convex set. The method can be still optimized to deal with very large data sets. In section 5 of [32] the author proposes several extensions of the online algorithm when changing the regularizer for α , the constraint set \mathcal{C} for \mathbf{D} , adding non negative constraints, and more.

The proposed method shows itself as powerful for many tasks like sparse component analysis or inpainting in large-scale processing. However, when specifying in BSS the method does not perform as expected. The main reason is that in the dictionary learning tasks the objective as well as the evaluation of the matrix factorization consist on how useful the factorization was to accomplish a given task. In BSS we are interested in the factorization itself, that is what really matters and the method is not adapted for that case.

Block-GMCA

The recent issued article [7] shows an interesting variant of the GMCA algorithm coined bGMCA (block-GMCA) which is the base for a warm-up stage. The algorithm finishes with a refining stage build upon a PALM-based method as will be mentioned in the section 4.6.

Focusing on the first stage, the author proposes a stochastic block update strategy maintaining the successful GMCA heuristics. Each iteration will consist of an update of a randomly chosen group of *columns* of \mathbf{A} and *lines* of \mathbf{S} ; \mathbf{A}_I and \mathbf{S}_I respectively. In simple words it accounts for updating only some sources. Calling I the selected group and I^C its complement (elements outside the chosen group), the chosen groups are updated using a residual term $\mathbf{R}_I = \mathbf{X} - \mathbf{A}_{I^C}^{(k)} \mathbf{S}_{I^C}^{(k)}$ that expresses the contribution of the group I to the observation matrix \mathbf{X} in the iteration k . The update step can be written as:

$$\begin{aligned}\mathbf{S}_I^{(k)} &= \text{prox}_{g(\cdot)}\left(\mathbf{A}_I^{(k-1)\dagger} \mathbf{R}_I\right) \\ \mathbf{A}_I^{(k)} &= \text{prox}_{h(\cdot)}\left(\mathbf{R}_I \mathbf{S}_I^{(k)\dagger}\right)\end{aligned}\tag{4.6}$$

where the g and the h functions can be the sparsity inducing term and the convex set indicator term respectively. The proposed approach is interesting as it shows superior performance than GMCA for some block sizes as well as a faster convergence. The method does not use the entire \mathbf{A} or \mathbf{S} matrices and it is possible to update the blocks independently always that the groups chosen are disjoint. The framework given is suitable for a parallel implementation. Nevertheless, each core needs the entire observation matrix \mathbf{X} , as it is needed to calculate the residual matrices \mathbf{R}_I and they share the same dimension, making it not well adapted for large scale problems.

4.2 DISTRIBUTED GENERALIZED MORPHOLOGICAL COMPONENT ANALYSIS

The first approach to tackle the problem is to focus on the most significant dimension which is the number of pixels t , this is the columns of \mathbf{X} and \mathbf{S} , the observation matrix and the source matrix respectively.

As already stated we will decompose the *big* problem into several BSS subproblems that can be managed with the machine's memory constraint as in equation (4.7). The observation matrix will be decomposed into groups of columns \mathbf{X}_j that will make up, each one, a new BSS subproblem from where an entire mixing matrix \mathbf{A} and a group of columns of the sources matrix \mathbf{S}_j will be estimated.

The subproblems can be written as follows:

$$\text{Subproblem } j : \mathbf{X}_j = \mathbf{A} \mathbf{S}_j + \mathbf{N}_j\tag{4.7}$$

where $\mathbf{N}_j, \mathbf{X}_j \in \mathbb{R}^{m \times t_j}$, $\mathbf{A}_{(j)} \in \mathbb{R}^{m \times n}$ and $\mathbf{S}_j \in \mathbb{R}^{n \times t_j}$. The t_j are the number of columns taken in each subproblem so that $t = \sum_j t_j$.

It is interesting to recall the objective function in equation (2.5) in terms of the defined subproblems.

$$\hat{\mathbf{A}}, \hat{\mathbf{S}} = \arg \min_{\mathbf{A}, \mathbf{S}} \sum_{j=1}^T \frac{1}{2} \|\mathbf{X}_j - \mathbf{A}\mathbf{S}_j\|_F^2 + \|\Lambda_j \odot \mathbf{S}_j\|_p + i_{\mathbf{Y}: \|\mathbf{Y}_k\|_2=1, \forall k}(\mathbf{A}) \quad (4.8)$$

where $\mathbf{S} = [\mathbf{S}_1, \dots, \mathbf{S}_j, \dots, \mathbf{S}_T]$. Following the spirit of the algorithms presented in the section 3.2 for solving multi-convex problems the problem in equation (4.8) can be divided into two problems, solving for \mathbf{S} leaving \mathbf{A} fixed and *vice-versa*.

Firstly, when solving for the source matrix, the problem is separable and can be solved independently for each block \mathbf{S}_j .

$$\hat{\mathbf{S}}_j = \arg \min_{\mathbf{S}_j} \frac{1}{2} \|\mathbf{X}_j - \mathbf{A}\mathbf{S}_j\|_F^2 + \|\Lambda_j \odot \mathbf{S}_j\|_p, \quad \forall j = 1, \dots, T \quad (4.9)$$

Then the estimation of the complete source matrix is obtained by the concatenation of the different outputs of equation (4.9), $\hat{\mathbf{S}} = [\hat{\mathbf{S}}_1, \dots, \hat{\mathbf{S}}_j, \dots, \hat{\mathbf{S}}_T]$. The solution for when the source matrix is fixed and we optimize with respect to \mathbf{A} is more complex.

$$\hat{\mathbf{A}} = \arg \min_{\mathbf{A}} \sum_{j=1}^T \frac{1}{2} \|\mathbf{X}_j - \mathbf{A}\mathbf{S}_j\|_F^2 + i_{\mathbf{Y}: \|\mathbf{Y}_k\|_2=1, \forall k}(\mathbf{A}) \quad (4.10)$$

The data fidelity term needs to use all of \mathbf{X} and \mathbf{S} if it want to output only one estimation for \mathbf{A} . The dimensions of these variables makes this operation prohibitive we will solve the subproblems independently in parallel, thus having several estimations of the mixing matrix.

$$\hat{\mathbf{A}}_{(j)} = \arg \min_{\mathbf{A}} \frac{1}{2} \|\mathbf{X}_j - \mathbf{A}\mathbf{S}_j\|_F^2 + i_{\mathbf{Y}: \|\mathbf{Y}_k\|_2=1, \forall k}(\mathbf{A}) \quad (4.11)$$

Coming back to the algorithm, each *epoch* (iteration of the main loop k in the optimization algorithm) consist of going through each group of columns. In other words it is going over j in \mathbf{X}_j estimating several $\hat{\mathbf{A}}_{(j)}$ and $\hat{\mathbf{S}}_j$ as in the figure (4.2.1).

The output of a whole loop in j gives an entire estimation of \mathbf{S} and T estimations of \mathbf{A} . These different estimators of the mixing matrices have to be aggregated into only one estimator \mathbf{A}_{FM} before starting a new epoch. To do this the Fréchet mean, which is described in section (4.3) will be used. Regarding our application, it is a method for doing a weighted mean of m dimensional vectors with a unit ℓ_2 norm constraint. Once the matrices are aligned - having the columns corresponding to each

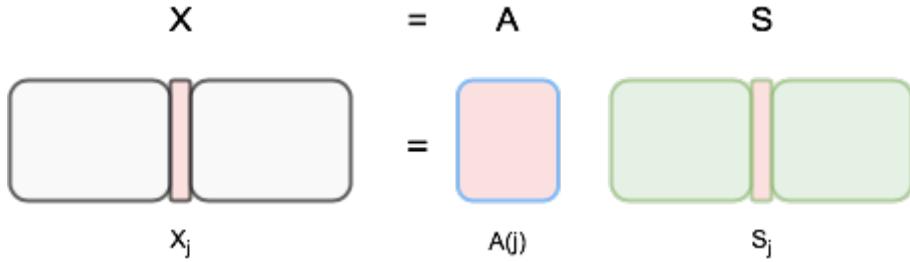


Figure 4.2.1: Basic distribution scheme for the optimization problem.

source in the right position - it is possible to fusion them. In an independent way, each group of columns corresponding to only one source will be the input of the Fréchet Mean algorithm that will output a new single column also living in the unit ℓ_2 sphere \mathbb{S}^{m-1} .

After the estimation of the mean mixing matrix \mathbf{A}_{FM} the epoch starts again following the PALS optimization strategy until convergence or until the maximum number of iterations is reached. A reduced version of the procedure can be seen in algorithm (9).

The fact that we cannot access entirely the data not only is obliging us to use a method to fusion the different estimators but also re-define the calculation of the threshold level and the weighting strategy for the Fréchet mean. They will be treated in section 4.4.

An important hypothesis taken is that *each source follows the same distribution on all the batches (group of columns)*. This implies that the batches must be big enough to contain representative statistics of the different sources and that if there is a detail that is only present in one batch but not in the other ones it will be considered as an outlier and therefore neglected. *A possible extension of the algorithm is to deal with this cases.*

It is possible to do a random permutation of the columns of the \mathbf{X} matrix before each iteration in the optimization algorithm to enforce the batches with a more similar distribution and be tight to the hypothesis taken. It is possible to do so, as the essential estimation is the one of the mixing matrix. The reason is because if this matrix is precisely estimated consequently the sources can be estimated easily.

4.3 THE FRÉCHET MEAN

In the need of an aggregation of the different \mathbf{A} estimations it is possible to exploit the fact that the columns of the mixing matrix have unit ℓ_2 norm (as it is fixed as a constraint for the possible solutions for \mathbf{A}). This means that they live in the unit sphere \mathbb{S}^{m-1} , a Riemannian manifold. Therefore, it is an insight that using the Riemannian center of mass would be appropriate for combining these estimators. The result will still lie in the sphere while if a simple euclidean mean is used, it will not be the same

Algorithm 9 [Simple Version] DGMCA

```

1:  $(\cdot)_j$  : Batch  $j$ 
2:  $(\cdot)_{(j)}$  : Estimated with the batch  $j$ 
3:  $(\cdot)^{(k)}$  : Epoch number  $k$ 
4:  $N_{epoch}$  : Maximal number of epochs
5:  $T$  : Total number of batches (group of columns)
6:  $\mathbf{a}_i$  : Column  $i$  from the  $\mathbf{A}$  matrix
7:  $Th_{\Lambda_j^{(k)}}$  : Threshold operator, hard or soft, with  $\Lambda_j^{(k)}$  as thresholding value
8:  $\mathbf{W}_{FM}^{(k)}$  : Weight matrix for the weighted Fréchet Mean
9:
10: procedure DGMCA( $\mathbf{X}$ , parameters)
11:   Initialize :  $\hat{\mathbf{A}}, \hat{\mathbf{S}}$ 
12:
13:   for  $k = 0, \dots, N_{epoch}$  do
14:     Calculation of:  $\Lambda^{(k)}, \mathbf{W}_{FM}^{(k)}$ 
15:     for  $j = 0, \dots, T$  do
16:       Estimation:  $\tilde{\mathbf{S}}_j^{(k+1)} \leftarrow (\hat{\mathbf{A}}_{FM}^{(k)})^\dagger \mathbf{X}_j$ 
17:       Thresholding:  $\hat{\mathbf{S}}_j^{(k+1)} \leftarrow Th_{\Lambda_j^{(k)}}(\tilde{\mathbf{S}}_j^{(k+1)})$ 
18:       Estimation:  $\hat{\mathbf{A}}_{(j)}^{(k+1)} \leftarrow \mathbf{X}_j (\hat{\mathbf{S}}_j^{(k+1)})^\dagger$ 
19:       Projection:  $\hat{\mathbf{a}}_{i, (j)}^{(k+1)} \leftarrow \frac{\hat{\mathbf{a}}_{i, (j)}^{(k+1)}}{\|\hat{\mathbf{a}}_{i, (j)}^{(k+1)}\|_2}, \forall i \in \{1, \dots, n\}$ 
20:     Fusion:  $\hat{\mathbf{A}}_{FM}^{(k+1)} \leftarrow \text{FrechetMean}\left(\hat{\mathbf{A}}_{(0)}^{(k+1)}, \dots, \hat{\mathbf{A}}_{(j)}^{(k+1)}, \dots, \hat{\mathbf{A}}_{(T)}^{(k+1)}, \mathbf{W}_{FM}^{(k)}\right)$ 
21:   return  $\hat{\mathbf{A}}_{FM}^{(k+1)}, \hat{\mathbf{S}}^{(k+1)}$ 

```

case. The reader is invited to see [18] for more information about optimization on manifolds.

The central idea is to fusion all the estimators of a given source. When speaking of the mixing matrix \mathbf{A} , each column corresponds to one source. Having as input all the different estimations of a single mixing matrix column $\hat{\mathbf{a}}_{1, (j)}$ and their corresponding weight $\omega_{1,j}^{FM}$ the Fréchet mean issues a new column $\hat{\mathbf{a}}_{1, FM}$.

In his works, *B. Asfari*, proves the existence and uniqueness of the L_P Riemannian center of mass in the Theorem 2.6 (based on his work [17]) and the convergence of a gradient-descent method to compute it in Theorem 2.10 from [16]. The Riemannian center of mass of a set of data points $\{\mathbf{a}_i\}_{i=1}^T$ in a Riemannian manifold \mathcal{M} is defined as the set of points which minimize the sum of squares of geodesic distances to the data points.

In order to add some intuition to the concept, the calculation of the Riemannian center of mass over

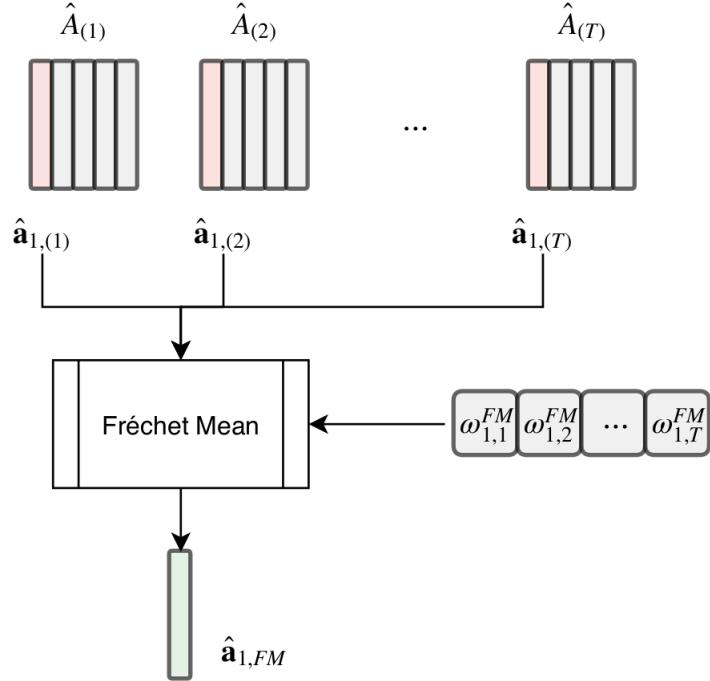


Figure 4.3.1: Illustration scheme of the use of the Fréchet mean applied to the first column of the mixing matrix estimations after their alignment as seen in section 4.4.4.

the sphere can be considered roughly as doing an angular mean with the elements considering them as vectors from the origin to the surface of the sphere. This is under the hypothesis that the elements are not so different from each other.

The Fréchet mean of the data set $\{\mathbf{a}_i\}_{i=1}^T \in \mathbb{S}^{m-1}$ where $\mathbf{a}_i \in \mathbb{R}^m$ and \mathbb{S}^{m-1} being the unit ℓ_2 sphere of dimension $m - 1$ is calculated as the minimizer of the following function:

$$f_P(\mathbf{x}) = \frac{1}{P} \sum_{i=1}^T \omega_i d^P(\mathbf{x}, \mathbf{a}_i), \quad 1 \leq P \leq \infty \quad (4.12)$$

having the fixed weights $0 \leq \omega_i \leq 1$ ($\sum_{i=1}^T \omega_i = 1$) and the geodesic distance $d^P(\cdot, \cdot)$. In our work the algorithm will be using $P = 2$.

The optimization on the manifold will be done following [16] and [18], using a simple line-search method with the following iteration:

$$\mathbf{x}^{k+1} = \exp_{\mathbf{x}^k}(-t_k \nabla f_P(\mathbf{x}^k)) \quad (4.13)$$

where $\exp_{\mathbf{x}^k}(\cdot) : T_{\mathbf{x}^k} \rightarrow \mathbb{S}^{m-1}$ is the exponential map of \mathbb{S}^{m-1} at $\mathbf{x}^k \in \mathbb{S}^{m-1}$ that maps the tangent space at $\mathbf{x}^k \in \mathbb{S}^{m-1}$ to the manifold \mathbb{S}^{m-1} , and $t_k \in \mathbb{R}$ is the step size that in practice will be fixed. The

tangent space allows to use euclidean elements in the neighborhood of its intersection with the \mathbb{S}^{m-1} manifold. The algorithm (10) shows the iterative scheme to obtain the Riemannian L_2 center of mass or the Fréchet mean in the particular case where the manifold is the sphere.

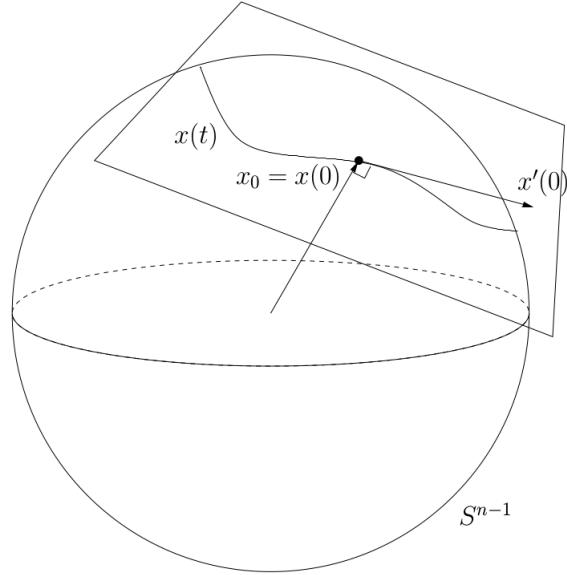


Figure 4.3.2: Illustration of a sphere \mathbb{S}^{m-1} and its tangent space at point x_0 taken from [18]. As the \mathbb{S}^{m-1} is a manifold embedded in \mathbb{R}^n the tangent plane $T_{x_0}\mathbb{S}^{m-1}$ can be seen as a hyperplane in \mathbb{R}^n tangent to the sphere in x_0 . The tangent map can be defined as the elements orthogonal to x_0 , as $T_{x_0}\mathbb{S}^{m-1} = \{z \in \mathbb{R}^n : x_0^T z = 0\}$.

In order to clarify the optimization iteration in equation (4.13), it can be thought firstly, as a projection from the manifold to the tangent space $T_{\mathbf{x}^k} \rightarrow \mathbb{S}^{m-1}$ of all the data set. This space is defined by the optimization element \mathbf{x}^k that always belongs to the manifold and naturally by the manifold itself. Then, it is followed by a weighted average in the tangent space represented by $\nabla f_P(\mathbf{x}^k)$. Finally, this new element in $T_{\mathbf{x}^k} \rightarrow \mathbb{S}^{m-1}$ is projected back to the manifold using the $\exp_{\mathbf{x}^k}(\cdot)$ and issuing the new element of the iteration \mathbf{x}^{k+1} .

4.3.1 OPTIMIZATION IN THE SPHERE

In the case of the unit sphere \mathbb{S}^{m-1} , the geodesic distance function becomes the angle between the two vectors, resulting in:

$$d(\mathbf{x}, \mathbf{a}_i) = \arccos(\mathbf{x}^T \mathbf{a}_i) \quad (4.14)$$

In order to optimize the cost function (4.12) we need its gradient as we are following a line search

method. The gradient of the cost function for the L_2 case is the following one:

$$\nabla f_2(\mathbf{x}^k) = - \left(\sum_{i=1}^T \omega_i d^{P-2}(\mathbf{x}^k, \mathbf{a}_i) \exp_{\mathbf{x}^k}^{-1}(\mathbf{a}_i) \right) \Big|_{P=2} = - \sum_{i=1}^T \omega_i \exp_{\mathbf{x}^k}^{-1}(\mathbf{a}_i) \quad (4.15)$$

where $\exp_{\mathbf{x}^k}^{-1}(\mathbf{a}_i)$ is the inverse of the exponential map (aka the log map) and it maps elements from the manifold to the tangent space at \mathbf{x}^k .

On the calculation of the Exp Map

The exponential map has an explicit calculation which in the case of the unit \mathbb{S}^{m-1} sphere. First we will define $\mathbf{z}_0 \in \mathbb{S}^{m-1}$ a point in the $m-1$ dimensional unit sphere, $\hat{\mathbf{z}}_1 \in T_{\mathbf{z}_0} \mathbb{S}^{m-1}$ a point in the tangent plane of the unit sphere at \mathbf{z}_0 and $\dot{\mathbf{z}}_0 = \hat{\mathbf{z}}_1 - \mathbf{z}_0$. The goal will be to deduce a formula to calculate $\mathbf{z}_2 = \exp_{\mathbf{z}_0}(\hat{\mathbf{z}}_1)$ following the notation introduced and the one seen in figure 4.3.3.

Even though we are working in a high dimensional space, the formulation of the exponential map can be brought back into a 2 dimensional space, in the case of \mathbb{S}^{m-1} . We are only interested in the space generated by the reference point \mathbf{z}_0 and by the element of the manifold $\hat{\mathbf{z}}_1$. The subspace spanned by these two elements will be called from now on $\Pi_{\{\mathbf{z}_0, \hat{\mathbf{z}}_1\}}$, that is the equivalent as $\Pi_{\{\mathbf{z}_0, \dot{\mathbf{z}}_0\}}$.

The use of a two dimensional plane is a consequence of the geodesic $\gamma(t)$ between \mathbf{z}_0 and \mathbf{z}_2 living in the intersection between \mathbb{S}^{m-1} and $\Pi_{\{\mathbf{z}_0, \dot{\mathbf{z}}_0\}}$. Given the definition of the exponential map we know that the element \mathbf{z}_2 we are looking for has to be contained in the geodesic $\gamma(t)$. In order to follow the geodesic we can rotate the element \mathbf{z}_0 an angle θ until it reaches the desired position of \mathbf{z}_2 . The angle needed for the rotation can be defined as $\theta = \|\hat{\mathbf{z}}_1\|_2$ because the angle between two elements of the unit sphere is its distance (geodesic).

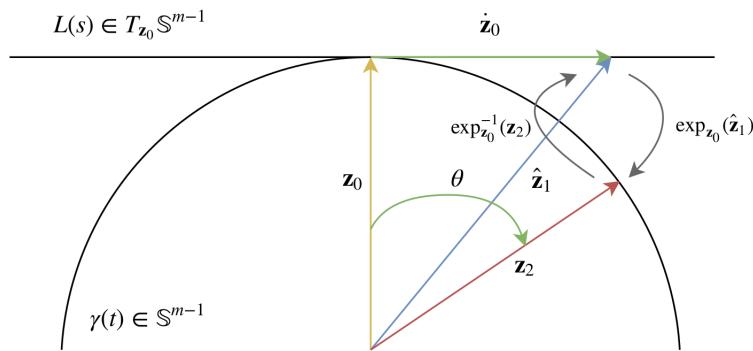


Figure 4.3.3: Illustration containing the elements used to calculate the Exp map and the Log map on the sphere \mathbb{S}^{m-1} . The image seen is the plane generated by \mathbf{z}_0 and $\hat{\mathbf{z}}_1$ (or \mathbf{z}_2). Then, $\gamma(t)$ is the intersection of the plane with \mathbb{S}^{m-1} , and $L(s)$ is the intersection of the plane with $T_{\mathbf{z}_0} \mathbb{S}^{m-1}$.

It is possible to define an orthonormal base for the $\Pi_{\{\mathbf{z}_0, \dot{\mathbf{z}}_0\}}$ subspace. It can be expressed as $B_\Pi = \{\mathbf{z}_0, \dot{\mathbf{z}}_0\}$ where $\dot{\mathbf{z}}_0 = \dot{\mathbf{z}}_0 / \|\dot{\mathbf{z}}_0\|_2$. Using the recently defined base we can represent the element \mathbf{z}_0 as $\mathbf{v}_{B_\Pi} = [1, 0]^{tr}$. The element \mathbf{z}_2 will be $\mathbf{v}_{B_\Pi}^\theta = R_\theta \mathbf{v}_{B_\Pi}$ where R_θ is the following two dimensional rotation matrix:

$$R_\theta = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}. \quad (4.16)$$

The rotated vector will be $\mathbf{v}_{B_\Pi}^\theta = [\cos(\theta), \sin(\theta)]^{tr}$ and changing the representation we obtain:

$$\mathbf{v}^\theta = \mathbf{z}_2 = \mathbf{z}_0 \cos(\theta) + \dot{\mathbf{z}}_0 \sin(\theta). \quad (4.17)$$

Finally, rewriting the previous expression we obtain the definition of the exponential map:

$$\exp_{\mathbf{z}_0}(\hat{\mathbf{z}}_1) = \mathbf{z}_0 \cos(\|\dot{\mathbf{z}}_0\|_2) + \frac{\dot{\mathbf{z}}_0}{\|\dot{\mathbf{z}}_0\|_2} \sin(\|\dot{\mathbf{z}}_0\|_2). \quad (4.18)$$

More information about the exponential map and Riemannian Geometry can be found in [18].

On the calculation of the Log Map

The log map or inverse exponential map will be calculated using two terms. The first one will give the direction of the new element and the second one its norm, as we are going from the sphere \mathbb{S}^{m-1} to the tangent plane $T_{\mathbf{z}_0} \mathbb{S}^{m-1}$. Let $\mathbf{z}_0, \mathbf{z}_2 \in \mathbb{S}^{m-1}$:

$$\hat{\mathbf{z}}_1 = \exp_{\mathbf{z}_0}^{-1}(\mathbf{z}_2) = \mathbf{z}_0 + \frac{\mathbf{z}_2 - \cos(d(\mathbf{z}_2, \mathbf{z}_0))\mathbf{z}_0}{\|\mathbf{z}_2 - \cos(d(\mathbf{z}_2, \mathbf{z}_0))\mathbf{z}_0\|_2} d(\mathbf{z}_2, \mathbf{z}_0) \quad (4.19)$$

where the distance function $d(\cdot, \cdot)$ is the one defined in equation (4.14). The first term can be simply understood if we consider a 2 dimensional plane generated by \mathbf{z}_2 and \mathbf{z}_0 that is the one seen in figure 4.3.3 that was called $\Pi_{\{\mathbf{z}_0, \dot{\mathbf{z}}_0\}}$ that is equivalent to $\Pi_{\{\mathbf{z}_0, \mathbf{z}_2\}}$. The element $\hat{\mathbf{z}}_1$ we are looking for will be contained in the one dimensional subspace $L(s)$ that is the intersection of the plane $\Pi_{\{\mathbf{z}_0, \mathbf{z}_2\}}$ with the tangent map $T_{\mathbf{z}_0} \mathbb{S}^{m-1}$.

Then,

$$P_{\text{span}\{\mathbf{z}_0\}}(\mathbf{z}_2) = \cos(d(\mathbf{z}_2, \mathbf{z}_0))\mathbf{z}_0 \quad (4.20)$$

can be thought as the projection of \mathbf{z}_2 into the 1 dimensional subspace generated by \mathbf{z}_0 . So subtracting from \mathbf{z}_2 , $P_{\text{span}\{\mathbf{z}_0\}}(\mathbf{z}_2)$, in equation (4.20) leave us the projection of \mathbf{z}_2 into the orthogonal space of $\text{span}\{\mathbf{z}_0\}$, this is $(\text{span}\{\mathbf{z}_0\})^\perp \in L(s)$. This last subspace is nothing but the tangent plane $T_{\mathbf{z}_0} \mathbb{S}^{m-1}$ given its orthogonality with \mathbf{z}_0 and the fact that it is a $m-1$ dimensional space embedded in \mathbb{R}^m . More

specifically, $\mathbf{z}_2 - \cos(d(\mathbf{z}_2, \mathbf{z}_0))\mathbf{z}_0$, not only belongs to $T_{\mathbf{z}_0}\mathbb{S}^{m-1}$ but to $L(s)$ as it is a linear combination of \mathbf{z}_0 and \mathbf{z}_2 .

Normalizing the direction and giving it the geodesic distance between the points in the sphere finishes the calculation of $\dot{\mathbf{z}}_0$. Then, the Log map equation (4.19) is obtained by just adding the \mathbf{z}_0 element.

Algorithm 10 Riemannian L^2 center of mass calculation over the unit sphere \mathbb{S}^{m-1}

```

1:  $\{\mathbf{x}_i\}_{i=1}^T$  : Group of elements to mean, where  $\mathbf{x}_i \in \mathbb{S}^{m-1}$ ,  $\forall i = 1, \dots, T$ 
2:  $N_{iter}$  : Maximal number of iterations
3:  $T$  : Total number of batches (group of columns)
4:  $\{\omega_i\}_{i=1}^T$  : Weights corresponding to each element verifying ( $0 \leq \omega_i \leq 1$ ,  $\sum_{i=1}^T \omega_i = 1$ )
5:
6: procedure FRECHETMEAN( $\{\mathbf{x}_i\}_{i=1}^T$ ,  $\{\omega_i\}_{i=1}^T$ ,  $t_k$ ,  $tol$ )
7:   Initialize :  $\hat{\mathbf{x}}^0 \in \mathbb{S}^{m-1}$ 
8:
9:   for  $k = 0, \dots, N_{epoch}$  do
10:    Calculation of  $\nabla f_P(\mathbf{x}^k)$  using equations (4.19) (4.15) (4.14)
11:    Calculation of  $\exp_{\mathbf{x}^k}(-t_k \nabla f_P(\mathbf{x}^k))$  using equation (4.18)
12:     $\mathbf{x}^{k+1} \leftarrow \exp_{\mathbf{x}^k}(-t_k \nabla f_P(\mathbf{x}^k))$ 
13:    if  $\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_2 \leq tol$  then
14:      break
15:   return  $\mathbf{x}^{k+1}$ 

```

4.4 HEURISTICS FOR DGMCA

Although one may think of heuristics in a pejorative sens it should not be always the case. In DGMCA the different heuristics have its own rationale and they have empirically proven to give satisfying results. They are in great part responsible for the robustness of the algorithm.

4.4.1 ADAPTATIVE THRESHOLDING

The thresholding parameter takes a vital role in the denoising and unmixing processes. It is important to have an automatic choice of its value providing robustness and simplicity to the algorithm. Algorithms like the Alternating Direction Method of Multipliers (ADMM) are well structured for distributed computing and for solving multi-convex problems as in [31]. Nevertheless, the choice of parameters is a hard task as it is difficult to give them an interpretation and the solution of the optimization algorithm depend strongly on them.

In the case of the PALS based algorithm, the choice of the threshold can be given an interpretation as it is finally applied as a threshold directly onto the sources.

Several strategies were tested using approximatively sparse signals as for example the one in the figure (2.2.2d). Considering this type of signals, which is the one that will be used in practice, there is a scheme that gives good results, that can be simply parallelized, that shows empirical robustness and which is the following one:

$$\lambda_r = K_\sigma \sigma_{N,r} + (\|\mathbf{S}_r^r\|_\infty - K_\sigma \sigma_{N,r}) e^{-k \alpha_r} \quad (4.21)$$

where λ_r is the threshold level for the source r , $\sigma_{N,r}$ is an estimation of the noise for the source r that is obtained with the mad operator from equation (3.12), α_r is a parameter to control the decrease of the exponential function, k is the number of the iteration and K_σ is a multiplier that will set the last level of threshold. Generally, using $K_\sigma = 3$ gives a good compromise between denoising and bias in the estimation. The subsection 4.4.2 details a method for choosing the α_r parameter that will boost the DGMCA performance.

The heuristic proposed is interesting due to several characteristics:

- It is easy to calculate in a distributed way as the signal statistic used is the $\|\mathbf{S}_r\|_\infty$ that is the calculation of the maximal absolute value of the signal. Taking the maximal value on each batch and then taking the maximum over these other results gives us the desired statistic.
- The fact that the threshold level starts with the highest valued elements make it more discriminant for the estimation of the mixing matrix.
- It is an adaptive threshold value as it depends on signals statistics giving its robustness and ensuring that there are always elements that are not being thresholded.
- Finishing with a threshold value that depends on the noise statistics ensure a good denoising for a fine and precise estimation.

In fact, the performance of an heuristic has to be evaluated with the corresponding weighting scheme used for the Fréchet mean. After testing several strategies, the combination of the heuristic and weighting scheme that resulted to be more robust and with better performance is the one presented and the one that will be used.

4.4.2 ESTIMATING THE OPTIMUM α_r PARAMETER - FITTING A GG DISTRIBUTION

After several tests comparing the performance of the algorithm while varying the sparsity level of the input signal and the value of the α_r parameter in the threshold heuristic there was found empirically

that there is an optimum level of this value and that this value is function of the sparsity level of the signal.

In order to estimate the optimal value of α_r , some iterations of the optimization algorithm are run using a default value for α_r . When the estimation of the mixing matrix is good enough it is possible to estimate the sources and use that estimation to fit the model of a Generalized Gaussian Distribution (GGD) function (see figure 4.9.1). After the β parameter of the GGD function is estimated, it will be mapped with a previously defined mapping function that outputs the optimum α_r from the β parameter of the GGD.

The GGD is a family of parametric continuous probability distributions on the real line where the probability density function is the following one:

$$f(x) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-(\frac{|x-\mu|}{\alpha})^\beta} \quad (4.22)$$

where $\Gamma(\cdot)$ denotes the gamma function. This family provides an accurate modeling of wavelet coefficients that are approximatively sparse which will be the case when used in practice. The estimation is done in the same spirit as the authors in [21] that use the GGD to model the marginal distribution of wavelet coefficients for texture retrieval.

To do the estimation there is a simple algorithm based on the maximum-likelihood estimator that will allow us to arrive to good value of this parameter as studied in detail in [20] and as used in [21]. Before giving the algorithm we have to define two functions :

$$\begin{aligned} g(\beta, \mathbf{x}, u) &= 1 + \frac{\psi\left(\frac{1}{\beta}\right)}{\beta} - \frac{\sum_{i=1}^N |x_i - u|^\beta \log |x_i - u|}{\sum_{i=1}^N |x_i - u|^\beta} + \frac{\log\left(\frac{\beta}{N} \sum_{i=1}^N |x_i - u|^\beta\right)}{\beta} \\ g'(\beta, \mathbf{x}, u) &= -\frac{\psi'\left(\frac{1}{\beta}\right)}{\beta^2} - \frac{\psi'\left(\frac{1}{\beta}\right)}{\beta^3} + \frac{1}{\beta^2} - \frac{\sum_{i=1}^N |x_i - u|^\beta (\log |x_i - u|)^2}{\sum_{i=1}^N |x_i - u|^\beta} + \frac{\left(\sum_{i=1}^N |x_i - u|^\beta \log |x_i - u|\right)^2}{\left(\sum_{i=1}^N |x_i - u|^\beta\right)^2} \\ &\quad + \frac{\sum_{i=1}^N |x_i - u|^\beta \log |x_i - u|}{\beta \sum_{i=1}^N |x_i - u|^\beta} - \frac{\log\left(\frac{\beta}{N} \sum_{i=1}^N |x_i - u|^\beta\right)}{\beta^2} \end{aligned} \quad (4.23)$$

Where ψ is the digamma function and the ψ' is the trigamma function.

The algorithm is based on a simple iterative gradient descent with a γ parameter (in practice was fixed to 0.1) and using 100 iterations:

$$\beta_{i+1} = \beta_i - \gamma \frac{g(\beta_i)}{g'(\beta_i)} \quad (4.24)$$

Based on simulations, the model that best maps the β parameter to the α_r is a truncated quadratic one ($\alpha_r(\beta_r) = a_1\beta_r^2 + b_1\beta_r + c_1$) with the parameters in equation (4.25). The result should always be kept between $0.01 \leq \alpha_r(\beta_r) \leq 20$.

$$a_1 = 4.5828, b_1 = -8.5807, c_1 = 5.2610 \quad (4.25)$$

In practice, to have a good estimation of the β parameter, the estimated sources should be denoised, that is done with the thresholding after the first estimation (that is why it is needed to run several iterations of the algorithm, so that the estimation becomes more robust and denoised). Then, each source has to be sorted in a decreasing array taking out the zero values and from that sorted non-zero valued array the β parameter can be estimated using the iterative gradient descend algorithm. It is important to notice that this estimation is done for each source resulting in the same number of sources than parameters.

Although this method provided the α_r that give a better unmixing when used in the DGMCA algorithm it has some drawbacks. In order to have a first *good* estimation of the sources it is needed at least half of the total number of iterations of the DGMCA algorithm. This adds a considerable amount of time to arrive to the solution. The estimator shows a bias with low values of SNR. Nevertheless, it is possible to reformulate the estimator in order to be robust to the type of noise considered.

4.4.3 WEIGHTING STRATEGIES FOR THE FRÉCHET MEAN

The weight given to each estimation is very important as it has to be a measure of reliability we have on the estimation. It should be able to discriminate and give a low weight to batches with a high noise component and to give a high wight to the batches with the most discriminant signals.

The choice taken is to calculate the weight as an estimation of the SNR (Signal to Noise Ratio) of the sources estimated *after the thresholding*. The intuition behind this is that in order to have a good estimation of the mixing matrix one should have a good SNR in the estimated sources. This means that there is signal information from where to estimate a more precise mixing matrix. The weight is calculated for each source for each batch.

For example, a batch where a source is not present at all will give an estimation of the source that will be made of low amplitude values (due to the noise components). In consequence, there will not be entries that pass the thresholding values giving an array full of zeros that will end up with a weight of zero. Finally, that estimation that was composed mostly of noise is not going to be considered in the aggregation with the Fréchet mean as its weight is zero.

The weight calculation can be expressed as follows:

$$\omega_{r,j}^{FM} = \frac{\|\hat{\mathbf{s}}_j^r\|_2^2}{\sigma_{X_j} \|\hat{\mathbf{A}}_{FM}^\dagger\|_F^2}, \quad \forall j \in \{1, \dots, T\}, \forall r \in \{1, \dots, n\} \quad (4.26)$$

where σ_{X_j} is an estimation of the noise in the observation batch that is supposed to be known (or can be estimated using the mad operator) and supposed to be white as well. The denominator is composed of the noise estimation considering the mixing matrix that is applied to the sources.

4.4.4 CORRECTING PERMUTATIONS OF THE MIXING MATRICES

In practice, when estimating the different \mathbf{A} matrices, there exist a difference in each estimation that is that they can be permuted as it will give the same result if $\mathbf{A}' = \mathbf{AP}$ and $\mathbf{S}' = \mathbf{P}^{-1}\mathbf{S}$ as the products $\mathbf{A}'\mathbf{S}' = \mathbf{AS}$ gives the same result, being \mathbf{P} an invertible permutation matrix.

In order to combine the estimations it is important to combine the columns corresponding to the same source (as each column from the mixing matrix corresponds to a single source). To do this, the *hungarian algorithm* (or *Kuhn-Munkres algorithm*) is used. It will not be explained in this work but the reader is referred to [22] and [23]. The number of sources to align for each source is equal to T , the number of batches in the problem. In practical terms it is the number of cores used in the distributed architecture. So the dimensionality of the elements involved will not make this calculation prohibitive in a computational point of view. The distance used to compare the different columns is the euclidean distance.

In an extension to this work, the distance used to compare could be changed into for example, the geodesic between the two elements in the Riemannian manifold, as the columns of the mixing matrix lie on the unit ℓ_2 sphere. It could be interesting as it will provide a better mating between the different columns as well as an elegant way to check for outliers in the estimation of \mathbf{A} .

4.5 COMPLEXITY ANALYSIS

By means of a simple complexity analysis it is possible to show that by following the pseudo-algorithm (11) in a distributed manner the computational cost can be reduced. The parallelization is considered to be in T nodes, the number of blocks used.

The pseudo inverse of the mixing matrix and the block source matrix is done by means of an SVD (Singular Value Decomposition). For example, for \mathbf{A} , the SVD is done of the product $\mathbf{A}^{tr}\mathbf{A}$. Then the singular values are inversed (in practice a threshold is set in the inverse of the singular values to be able to control the condition number of the matrix), and $(\mathbf{A}^{tr}\mathbf{A})^{-1}$ is issued. Finally, a multiplication with \mathbf{A}^{tr} give us the pseudo inverse of \mathbf{A} . The complexity of this calculation is $O(n^3 + n^2m + m^2n)$ as

A is a $m \times n$ matrix.

In the complexity analysis of the GMCA the steps that consume most of the time are the estimations of **S** and of **A**. Nevertheless, the algorithm has a linear complexity in the biggest dimension which is t , the number of pixels. Considering, m the number of observations, lines of **X**, n the number of sources, and N_{ep} as the number of epochs of the algorithm; the complexity of the GMCA algorithm is:

$$\text{GMCA} \sim O(N_{ep} t (mn + n^2 + m)). \quad (4.27)$$

Now we will consider the DGMCA algorithm with a parallelization of T blocks of size t_j and using the same amount of cores. The estimation time is approximatively divided by T although the time of the calculation of the Fréchet Mean has to be added. Finally the complexity is as follows:

$$\text{DGMCA} \sim O(N_{ep} [t_j (mn + n^2 + m) + T^3 + TnmN_{FM}]), \quad (4.28)$$

where N_{FM} is the number of iterations of the optimization algorithm calculating the Fréchet Mean which in practice is set to 100 that guaranteed convergence in the numerical experiments. The number of epochs was set to 100 as well as the algorithm had already stabilized in a value giving no signs of improvement when increasing the epoch number.

The $O(T^3)$ term corresponds to the munkres algorithm for aligning the estimations of the mixing matrices in one epoch. We are conscious that is not an scalable algorithm and new approaches to align the matrices should be explored. The last term $O(TnmN_{FM})$ corresponds to the Fréchet Mean calculation in one epoch. It is interesting to notice that the problematic dimension t does not appear in the algorithm as we are only dealing with **A** matrices.

Finally, we can see that the gain in computational time is approximatively by a factor of T , the number of cores used in the distribution algorithm. It is somehow logic, as the original algorithm is linear in t . The procedures used for the Fréchet Mean have at least one order less of magnitude than the estimation part.

4.6 ENHANCING DGMCA

AMCA weights - Dealing with partially correlated sources

The AMCA method (Adaptative Morphological Component Analysis) is really useful to maintain the Morphological Diversity Principle (MDP) in the sources. A weighting matrix is used to penalize partially correlated sources so that they do not play a role in the unmixing process. These sources will mislead the algorithm into a solution that is not the one we are looking for as the MDP is not respected.

The method is implemented as proposed in [4]. The construction of the mixing matrix depends on a regularized version of the inverse of the ℓ_1 norm done over the columns of \mathbf{S} . The weight elements are constructed as follows:

$$\forall t = 1, \dots, T; \quad w_q[t] = \frac{1}{\|[\mathbf{S}]^t\|_{\ell_1} + \epsilon} \quad (4.29)$$

where $[\mathbf{S}]^t$ is the t -th column of \mathbf{S} and ϵ is a small value, typically 10^{-6} . Briefly, the penalization is done using the sparsity level of the columns of \mathbf{S} . The \mathbf{W} is constructed as follows:

$$\mathbf{W} = \text{diag}(\mathbf{w}_q) \quad (4.30)$$

where $\mathbf{w}_q = [w_q[0], \dots, w_q[T]]^{tr}$. The weight matrix modifies \mathbf{A} estimation (line 18 from algorithm (9)) as it can be seen in the next equation:

$$\hat{\mathbf{A}} = \mathbf{X} \mathbf{W} \mathbf{S}^T \ (\mathbf{S} \mathbf{W} \mathbf{S}^T)^{-1} \quad (4.31)$$

The method has proven to be simple and very effective against partially correlated sources.

PALM refinement step

The adaptive threshold strategy of DGMCA is very efficient, and makes it, most of the times, reliable, robust and fast, but does not yield precision. The complement of this algorithm, is the PALM procedure. A robust first guess of the solution is given by the DGMCA that will be close enough the desired solution that if a PALM refinement step is applied afterwards, it is possible to arrive to the fine solution. This second method is chosen due to its convergence characteristics, as it is proven that it converges into a local minimum. More details about the use of PALM and GMCA can be found on [7] and [6].

In this work the refinement step will not be used as we are focused on the DGMCA procedure. However, it is important to have it in mind when doing an implementation to analyze crucial data.

4.7 SEGMENTING LINES AND COLUMNS

A natural extension of the previous work is to segment not only the columns of the observation matrix but also the lines.

In this version of the algorithm, coined xDGMCA, for each scanned block (\mathbf{X}_j^i) in a precise group of columns j there is an estimation of columns of the source matrix $(\mathbf{S}_j^{(i)})$. If the group of columns j is left constant and we move through all the group of lines we will have several estimations of $\mathbf{S}_j^{(i)}$ where

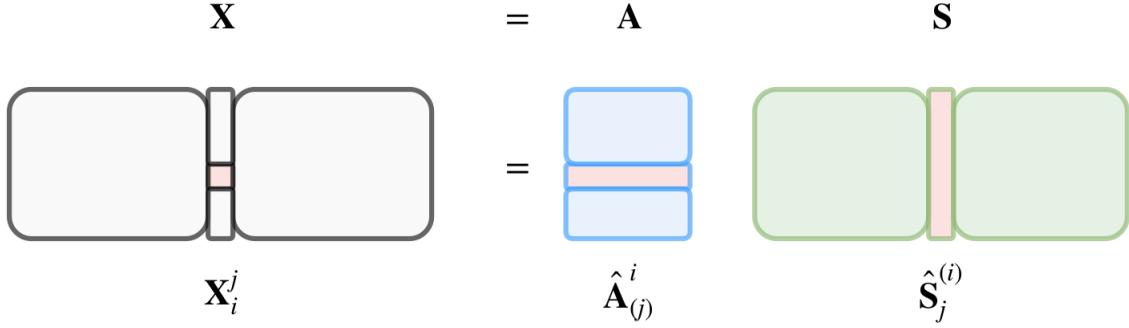


Figure 4.7.1: Scanning and estimating scheme for the DGMCA when the segmentation is done in the columns and the lines.

the difference is which part of the mixing matrix $(\mathbf{A}_{MF}^i)^\dagger$ was used for its estimation. Recalling that the estimation is done in the following way:

$$\hat{\mathbf{s}}_j^{(i)} \leftarrow Th_\lambda \left((\mathbf{A}_{MF}^i)^\dagger \mathbf{X}_j^i \right) \quad (4.32)$$

A need of combining these different estimations of the same group of columns of the source matrix can be identified when aiming to arrive to the same configuration of the algorithm when only the columns are segmented. It is of our interest to profit from the successful scheme proposed before.

In this case, the Fréchet mean cannot be applied as we do not have the unit ℓ_2 norm constraint and we can not impose it as we are dealing with only a part of the sources. If the information of the source would be complete, this is having the entire line of the source matrix a unit ℓ_2 norm constraint could be imposed. In our case, the different scales of the sources are unknown leaving not many options to consider.

The combination method will be a simple weighted average of the different estimators of $\hat{\mathbf{s}}_j^{(i)}$.

$$\hat{\mathbf{s}}_j = \sum_i \alpha_i \hat{\mathbf{s}}_j^{(i)} \quad (4.33)$$

where the weight calculation will follow the same spirit of the previous algorithm making use of the SNR of the estimations *after the thresholding step*:

$$\alpha_i = \frac{\|\hat{\mathbf{s}}_j^{(i)}\|_F^2}{\sigma_{X_j^i} \|(\hat{\mathbf{A}}_{MF}^i)^\dagger\|_F^2} \quad (4.34)$$

The weights are normalized so that they have unit ℓ_1 norm ($\sum_i |\alpha_i| = 1$). After the combination we

arrived to the setup of the previous algorithm where the Fréchet mean can be applied to combine the different estimators of the mixing matrix. The weights used at this step will be calculated as before but using the $\hat{\mathbf{S}}_j$ issued from the estimation in equation (4.33).

The AMCA weights strategy, described in section 4.6, is important for the DGMCA algorithm. This is due to the fact that the algorithm deals with blocks of smaller size making the partial correlation of the sources more harmful for the separation process. It is worth to consider a fusion technique to combine the estimators coming from the different $\mathbf{S}_j^{(i)}$.

A natural way to face the problem is to work with the newly combined estimation $\hat{\mathbf{S}}_j$ as it would have been the original estimation using all the lines of the corresponding group of columns, this is \mathbf{X}_j instead of \mathbf{X}_j^i . Lacking robustness, another method which is based on the median operator is though of. The median operator is more flexible against outliers that can drag the mean towards it.

$$\beta_j^{(i)} = \frac{1}{\left\| [\tilde{\mathbf{S}}_j^{(i)}] \right\|_1 + \epsilon} \in \mathbb{R}^{T_j} \quad (4.35)$$

where T_j is the number of columns of the group j and $\tilde{\mathbf{S}}_j^{(i)} \in \mathbb{R}^{N_{src} \times T_j}$. The combination is done following equation (4.36) using the different AMCA weights of the different estimations $\hat{\mathbf{S}}_j$ as seen in equation (4.35).

$$\beta_j[l] = \text{median}_i(\alpha_i[l]\beta_j^{(i)}[l]) \quad (4.36)$$

4.8 THE PSEUDO-ALGORITHMS

Algorithm 11 Distributed-GMCA (DGMCA)

```

1:  $\mathbf{S}_j$  : Column group  $j$  of  $\mathbf{S}$  ( $\mathbf{S}_j \in \mathbb{R}^{N_{src} \times T_j}$ )
2:  $\hat{\mathbf{A}}_{(j)}$  : Mixing matrix estimated using the  $j$  group of columns ( $\mathbf{X}_j$ )
3:  $\mathbf{W}^{FM} = [\omega_0, \dots, \omega_r, \dots, \omega_{N_{src}}]$  : Weighting factors for the FrechetMean
4:  $\omega_r = [\omega_{0,r}, \dots, \omega_{j,r}, \dots, \omega_{T,r}]^{tr}$ 
5:  $\lambda_r$  : Thresholding parameter for source  $r$ 
6:  $\alpha_r^e$  : Threshold exponential decay parameter for the source  $r$ 
7:  $\sigma_{N,r}$  : Noise standard deviation for source  $r$ 
8:  $K_\sigma$  : End value thresholding  $\sigma N$  multiplier
9:  $\sigma_{X,j} = MAD(\mathbf{X}_j)$ 
10:  $T$  : Number of batchs being used
11:  $N_{src}$  : Number of sources present in the mixture
12:  $\beta_j$  : Weight vector for the AMCA algorithm ( $\beta_j \in \mathbb{R}^{T_j}$ )
13:  $\mathbf{B}$  : Weighting matrix for the AMCA algorithm
14:
15: procedure DGMCA( $\mathbf{X}$ , parameters)
16:   Initialize:  $\mathbf{A}_{init}$ 
17:    $\mathbf{A}_{MF} \leftarrow \mathbf{A}_{init}$ 
18:
19:   for  $k = 0, \dots, N_{epoch}$  do
20:     Shuffle S and X column order.
21:     for  $r = 0, \dots, N_{src}$  do
22:        $\lambda_r \leftarrow K_\sigma \sigma_{N,r} + (\|\hat{\mathbf{S}}^r\|_\infty - K_\sigma \sigma_{N,r}) e^{-k \alpha_r^e}$  (Threshold calculation)
23:
24:     for  $j = 0, \dots, T$  do
25:        $\tilde{\mathbf{S}}_j \leftarrow (\mathbf{A}_{MF})^\dagger \mathbf{X}_j$  (Estimation)
26:        $\beta_j \leftarrow \frac{1}{\|\tilde{\mathbf{S}}_j\|_1 + \epsilon}$  (AMCA weight calculation)
27:
28:     for  $r = 0, \dots, N_{src}$  do
29:        $\hat{\mathbf{S}}_{j,r} \leftarrow Th_{\lambda_r}(\tilde{\mathbf{S}}_{j,r})$  (Thresholding)
30:        $\omega_{j,r}^{FM} \leftarrow \frac{\|\hat{\mathbf{S}}_{j,r}\|_F^2}{\sigma_{X,j} \|\hat{\mathbf{A}}_{MF}^\dagger\|_F^2}$  (FM Weight calculation)
31:
32:      $\hat{\mathbf{A}}_{(j)} \leftarrow \mathbf{X}_j \mathbf{B} \hat{\mathbf{S}}_j^T (\hat{\mathbf{S}}_j \mathbf{B} \hat{\mathbf{S}}_j^T)^{-1}$  (Estimation with AMCA weights)
33:
34:     for  $r = 0, \dots, N_{src}$  do
35:        $\hat{\mathbf{a}}_{(j),r} \leftarrow \frac{\hat{\mathbf{a}}_{(j),r}}{\|\hat{\mathbf{a}}_{(j),r}\|_2}$  (Normalization)
36:
37:     for  $r = 0, \dots, N_{src}$  do
38:       for  $j = 0, \dots, T$  do
39:          $\omega_{j,r}^{FM} \leftarrow \frac{\omega_{j,r}^{FM}}{\sum_l \|\omega_{l,r}^{FM}\|_1}$  (Normalization)
40:
41:      $\mathbf{A}_{FM} \leftarrow \text{FrechetMean}(\hat{\mathbf{A}}_{(0)}, \dots, \hat{\mathbf{A}}_{(j)}, \dots, \hat{\mathbf{A}}_{(T)}, \mathbf{W}^{FM})$  (Fusion)
42:
43:     Restore S and X original column order.
44:      $\mathbf{A} \leftarrow \mathbf{A}_{MF}$ 
45:      $\mathbf{S} \leftarrow \mathbf{A}^\dagger \mathbf{X}$  (Estimation)
46:     return  $\mathbf{A}, \mathbf{S}$ 

```

Algorithm 12 Extended Distributed-GMCA (xDGMCA) segmenting columns and lines

```

1:  $\mathbf{X}^i$  : Line group  $i$  of  $\mathbf{X}$ 
2:  $\mathbf{X}_j^i$  : Column group  $j$  and line group  $i$  of  $\mathbf{X}$ 
3:  $\mathbf{S}_j^{(i)}$  : Column group  $j$  of  $\mathbf{S}$  estimated from line group  $i$ 
4:  $\hat{\mathbf{S}}^{[r]}$  : Entire line corresponding to the source  $r$  of the estimated source matrix.
5:
6: procedure xDGMCA( $\mathbf{X}$ , parameters)
7:   Initialize:  $\mathbf{A}_{init}$ 
8:    $\mathbf{A}_{MF} \leftarrow \mathbf{A}_{init}$ 
9:
10:  for  $k = 0, \dots, N_{epoch}$  do
11:    Shuffle column and line order.
12:    for  $r = 0, \dots, N_{src}$  do
13:       $\lambda_r \leftarrow K_\sigma \sigma_{N,r} + (\|\hat{\mathbf{S}}^r\|_\infty - K_\sigma \sigma_{N,r}) e^{-k \alpha_r^e}$  (Threshold calculation)
14:
15:      for  $j = 0, \dots, T$  do
16:        for  $i = 0, \dots, N_{lines}$  do
17:           $\tilde{\mathbf{S}}_j^{(i)} \leftarrow (\mathbf{A}_{MF}^i)^\dagger \mathbf{X}_j^i$  (Estimation)
18:           $\beta_j^{(i)} \leftarrow \frac{1}{\|\tilde{\mathbf{S}}_j^{(i)}\|_1 + \epsilon}$  (AMCA weight calculation)
19:           $\hat{\mathbf{S}}_j^{(i)} \leftarrow Th_{\lambda_r}(\tilde{\mathbf{S}}_j^{(i)})$  (Thresholding)
20:           $\alpha_i \leftarrow \frac{\|\hat{\mathbf{S}}_j^{(i)}\|_F^2}{\sigma_{X_j^i} \|(\hat{\mathbf{A}}_{MF}^i)^\dagger\|_F^2}$  (Weight calculation)
21:
22:        Normalize  $\alpha_i$  so that  $\sum_i |\alpha_i| = 1$ 
23:         $\beta_j[l] \leftarrow \text{median}_i(\alpha_i[l] \beta_j^{(i)}[l])$ ,  $l = 1, \dots, T_j$  (AMCA weight fusion)
24:         $\hat{\mathbf{S}}_j \leftarrow \sum_i \alpha_i \hat{\mathbf{S}}_j^{(i)}$  (Fusion)
25:        for  $r = 0, \dots, N_{src}$  do
26:           $\omega_{j,r}^{FM} \leftarrow \frac{\|\hat{\mathbf{S}}_j^r\|_F^2}{\sigma_{X_j^r} \|\hat{\mathbf{A}}_{MF}^\dagger\|_F^2}$  (FM Weight calculation)
27:
28:        for  $i = 0, \dots, N_{lines}$  do
29:           $\hat{\mathbf{A}}_{(j)}^i \leftarrow \mathbf{X}_j^i \mathbf{B}_j \hat{\mathbf{S}}_j^T (\hat{\mathbf{S}}_j \mathbf{B}_j \hat{\mathbf{S}}_j^T)^{-1}$  (Estimation with AMCA weights)
30:
31:        for  $r = 0, \dots, N_{src}$  do
32:           $\hat{\mathbf{a}}_{(j),r} \leftarrow \frac{\hat{\mathbf{a}}_{(j),r}}{\|\hat{\mathbf{a}}_{(j),r}\|_2}$  (Normalization)
33:        for  $r = 0, \dots, N_{src}$  do
34:          for  $j = 0, \dots, T$  do
35:             $\omega_{j,r}^{FM} \leftarrow \frac{\omega_{j,r}^{FM}}{\sum_l \|\omega_{l,r}^{FM}\|_1}$  (Normalization)
36:         $\mathbf{A}_{FM} \leftarrow \text{FrechetMean}(\hat{\mathbf{A}}_{(0)}, \dots, \hat{\mathbf{A}}_{(j)}, \dots, \hat{\mathbf{A}}_{(T)}, \mathbf{W}^{FM})$  (Fusion)
37:
38:      Restore  $\mathbf{S}$  and  $\mathbf{X}$  original column order.
39:       $\mathbf{A} \leftarrow \mathbf{A}_{MF}$ 
40:       $\mathbf{S} \leftarrow \mathbf{A}^\dagger \mathbf{X}$  (Estimation)
41:      return  $\mathbf{A}, \mathbf{S}$ 

```

4.9 NUMERICAL EXPERIMENTS

4.9.1 INPUT DATA

In this section, the sources are simulated so that they follow a Generalized Gaussian distribution (figure (4.9.1)) with a fixed β parameter which regulates the level of sparsity of the signal. This model is used as it is a good representation of approximatively sparse signals. In the next chapter, the numerical experiments will be done using real images.

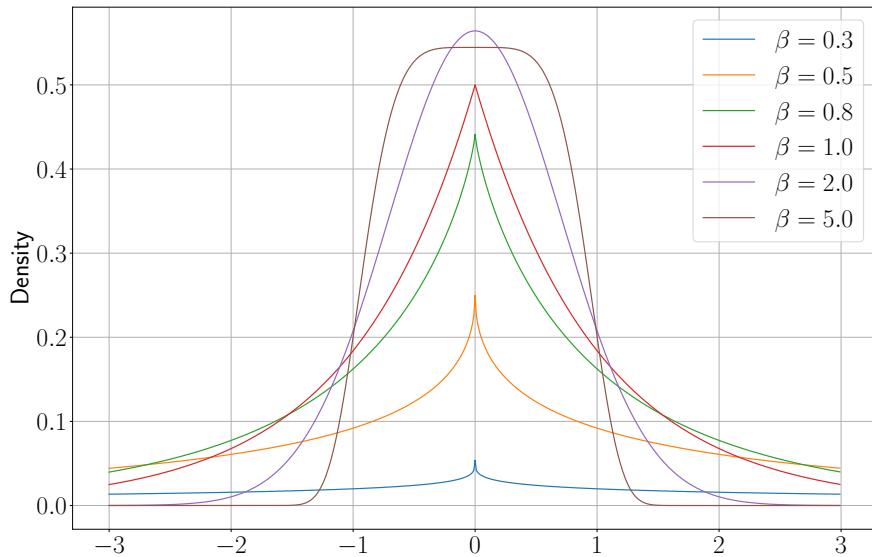


Figure 4.9.1: GG distribution densities for different β parameters.

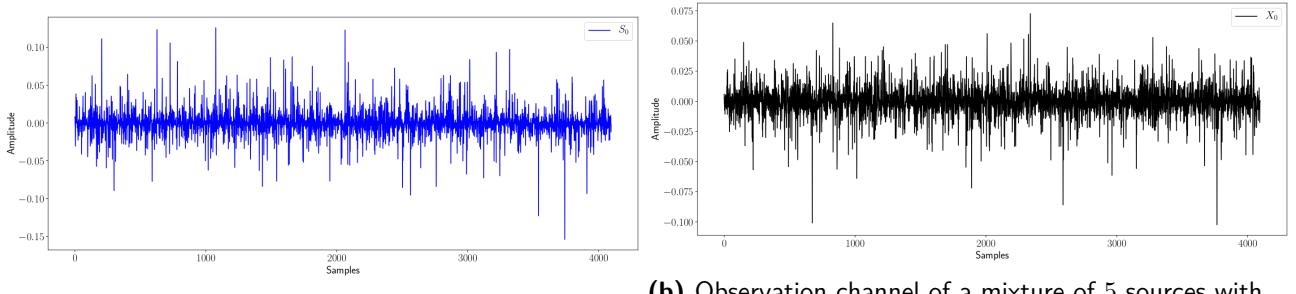
After the sources have been simulated they are mixed with a random mixing matrix that respects the unit ℓ_2 norm. Then, WGN is added to the mixture to get the desired SNR. In order to have a rough idea of the signals treated, figure 4.9.2 shows a realization using standard values for the parameters.

Performance metric - Mixing Matrix Criterion

The metric used to evaluate the performance is purely based on the mixing matrix \mathbf{A} and it is defined as follows:

$$C_A = \|(\mathbf{P}\mathbf{A}^*)^\dagger \hat{\mathbf{A}} - \text{diag}((\mathbf{P}\mathbf{A}^*)^\dagger \hat{\mathbf{A}})\|_{\ell_1} \quad (4.37)$$

where the norm is done column-wise, the $\text{diag}(\cdot)$ operates on matrices and gives a copy of the matrix with zeros in all the elements that do not belong to its diagonal and \mathbf{P} is a permutation matrix



(a) Clean source using a GG model with β of 0.5.

(b) Observation channel of a mixture of 5 sources with observational noise with a SNR of 60dB.

Figure 4.9.2: Simulations under the GG scheme used for testing the algorithms performance.

that allows to correct the difference in permutations following the section 4.4.4.

It is possible to output two results from this metric, one taking the median of the norm applied to each column and the other using the mean, thus leading to $C_{A,med}$ and $C_{A,mean}$ respectively. In this work we will express the different C_A metrics in dB following the next formula:

$$C_A (\text{dB}) = -10 \log_{10} (C_A) \quad (4.38)$$

Normally, $C_{A,med}$ is used as it is more robust to outliers, unless it is specified otherwise.

4.9.2 RESULTS

More tests can be found in the appendix of this document. Nevertheless, the main ones are present in the chapters.

Performance test

The x-axis corresponds to the size t_j of each subproblem (j), which is set for all j to be $t_j = t/J$. The y-axis represents the separation quality, measured by a mixing matrix criterion [4] defined as the average value of the ℓ_1 norm of the columns of $\hat{\mathbf{P}}\hat{\mathbf{A}}^\dagger\mathbf{A}^* - \text{diag}((\mathbf{P}\mathbf{A}^*)^\dagger\hat{\mathbf{A}})$, where \mathbf{A}^* is the ground truth, \mathbf{P} accounts for the correction of the permutations.

To generate the experiments, the source matrix was randomly sampled from a Generalized Gaussian distribution with several profile parameters β between 0.35 and 1.4, having $n = 10$ sources, $t = 10000$ samples and $m = 20$ observations. The noise matrix \mathbf{N} was set to have a SNR of 15dB. The matrix \mathbf{A} is random and with a condition number fixed to 10. The experiment is repeated 3 times and the mean of the results is being plotted.

Four parallelized algorithms are compared: the presented *DGMCA*, the *DGMCA naive* method consisting of solving the J subproblems independently until convergence and performing the aggregation at the end, the *DGMCA Euclidean* method where the RCM is substituted with an Euclidean

mean for the aggregation, and the ODL with its hyperparameters optimized. The results of the different algorithms are benchmarked with the ODL and the GMCA using the entire observation matrix (which is only possible due to the relatively small t that we chose for the sake of the comparison).

The DGMCA outperforms the other parallelized methods maintaining a similar performance compared to the GMCA. It is worth to remark that the performance is limited by the size of the mini-batch and not by the total size t which can be increased thus making the number of mini-batches increase. The separation quality is only reduced for extremely small mini-batches t_j , which was expected due to the lack of statistics for the algorithm to work. The *DGMCA naive* is not plotted for the two smallest t_j as some mini-batches only contained noise, which caused the algorithm not to converge for the threshold level used (as the subproblems are solved independently until convergence).

In addition, the huge gap between DGMCA and its naive version confirms the usefulness of using an aggregation process during the iterations. Furthermore, using the RCM as aggregation and therefore taking into account the geometry of the problem enables better results than with an Euclidian mean.

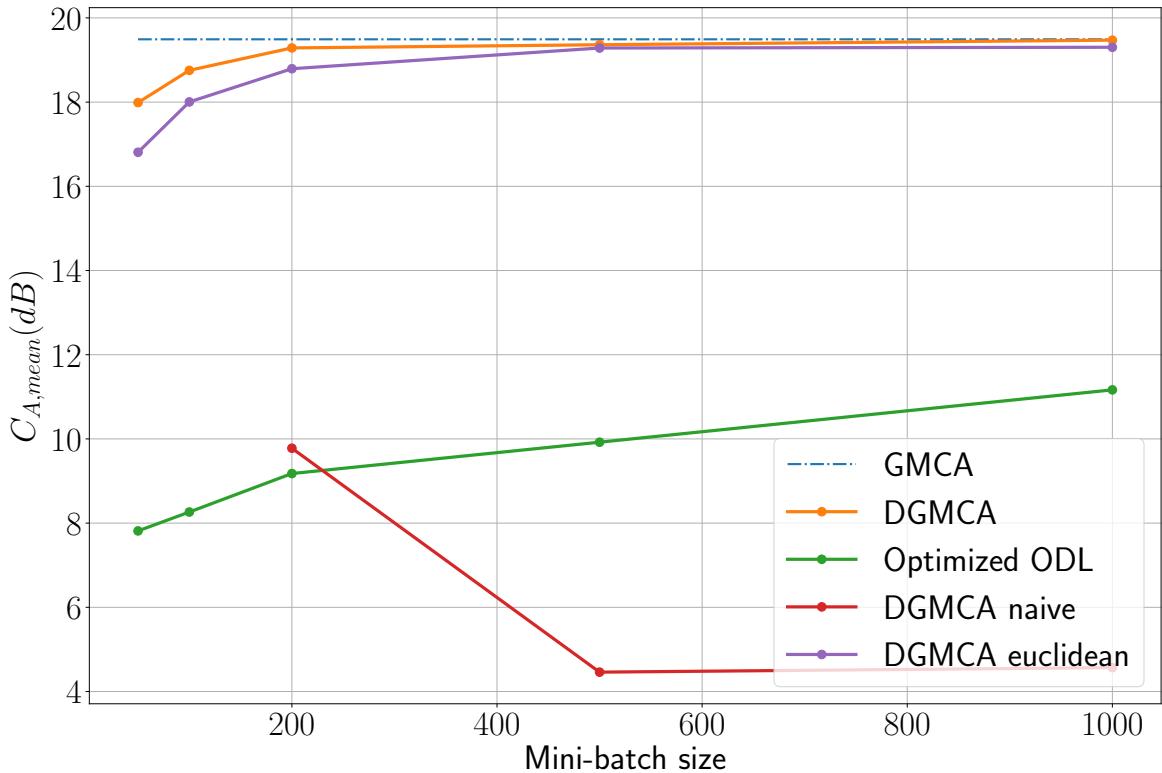


Figure 4.9.3: Performance comparison of the GMCA, the ODL (Online Dictionary Learning [32]), and the DGMCA and its variants. Note that the parameters of the ODL algorithm have been optimized for this experiment by an exhaustive search.

Computational time gain

The experiment was run using a C++ parallelized version of the DGMCA algorithm and the maximum number of mini-batches used is 40 as it is the number of cores the computer cluster used had. The setup of the experience is similar to the one in Fig (4.9.3), with a β parameter of 0.5, having $n = 5$ sources, $t = 10000$ samples and a *SNR* of 40dB.

The linear trend of the time gain was predicted by the complexity analysis of the algorithms, and now confirmed by the numerical experiment.

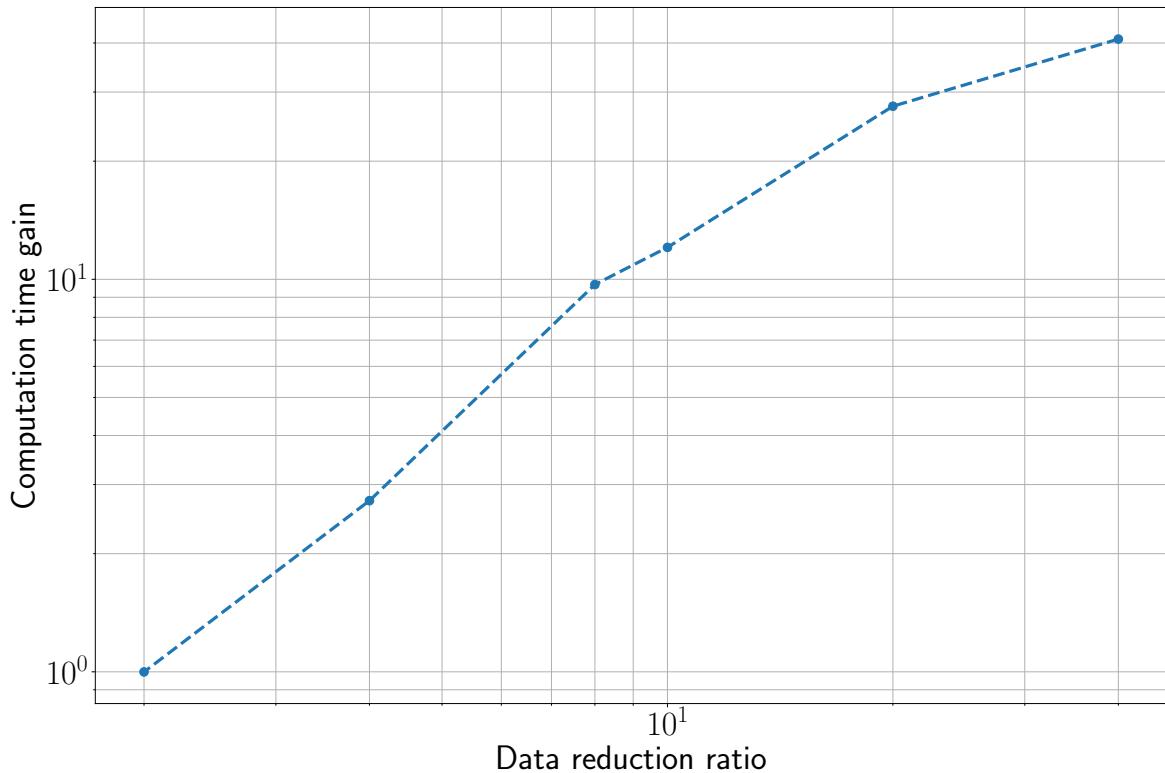


Figure 4.9.4: Computational time gain between the parallelized DGMCA and the GMCA algorithms against the data reduction ratio which is calculated as the problem total size, t , divided by the size of the mini-batch, $t/t_j = J$. Each point on the figure represents the mean over 10 problems.

Performance test with different segmentations

The experiment used a GG model as input for the sources and a random generator for the mixing matrix following the section 4.9.1. The results were obtained using the following initial conditions:

- Data generated with the GG model with $\beta = 0.5$.

- SNR = 60dB
- Number of sources : 5
- Number of observation channels : 40
- Batch sizes (column group sizes) : [200, 250, 400, 500, 1000, 2000, 4000]
- Line group size : 5
- Number of samples per channel : 4500
- Number of epochs : 100
- Number of repetitions : 60

In the figure 4.9.5 the mean of the $C_{A,med}$ obtained in the repetitions is plotted. The repetitions are done as there is a stochastic component in the algorithm when mixing the columns at the start of each epoch, it can be seen in the algorithm (11).

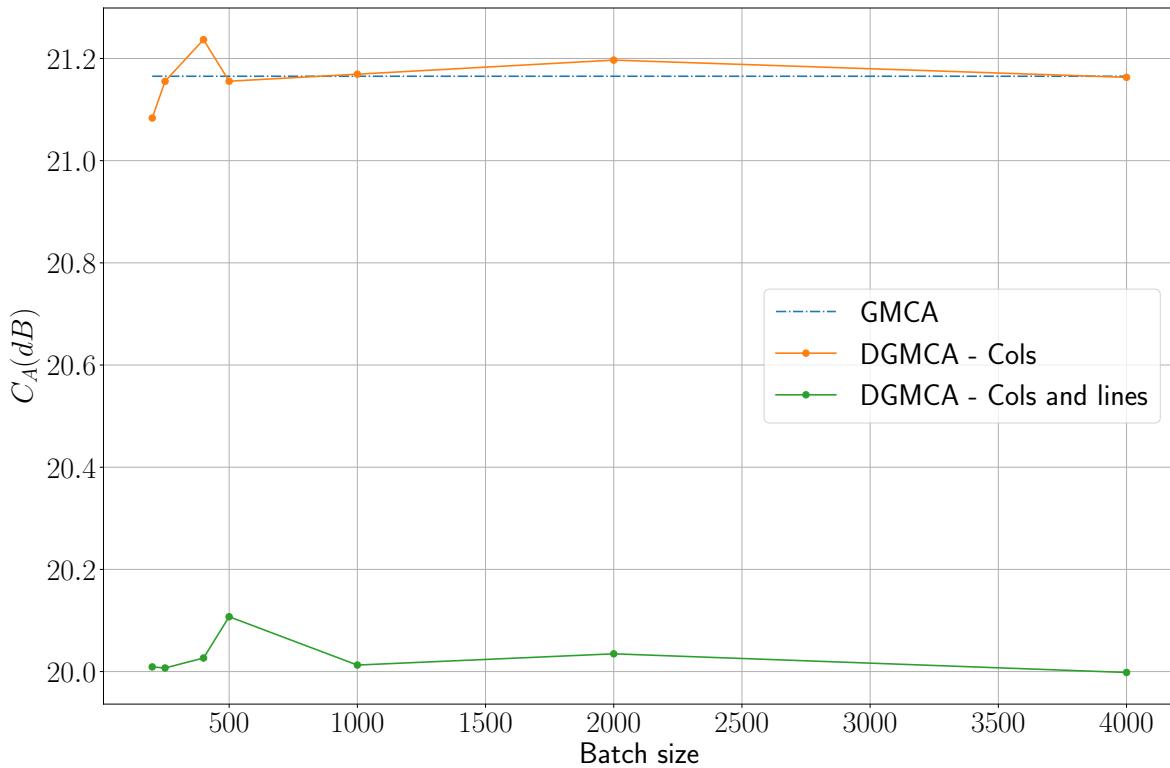


Figure 4.9.5: Performance result comparing the original algorithm GMCA, the DGMCA segmenting only the columns and the DGMCA segmenting the columns and the lines.

The figure 4.9.5 shows how the performance is maintained for different batch sizes. The performance level achieved by the DGMCA algorithm when the segmenting is done only in the columns is practically the same when the batches are big enough. There is a natural performance degradation when the sizes used are too small. The fact that both performance curves of the DGMCA algorithm do not vary when changing the batch size validates the use of the weighted Fréchet mean as the combination method.

The extension of the algorithm, when we segment both lines and columns, exhibits a loss in performance meaning that the combination method proposed in section 4.7 is not as successful as the Fréchet mean. Nevertheless, it has an acceptable performance as there is approximately 1.2dB of loss with respect to the original GMCA algorithm³.

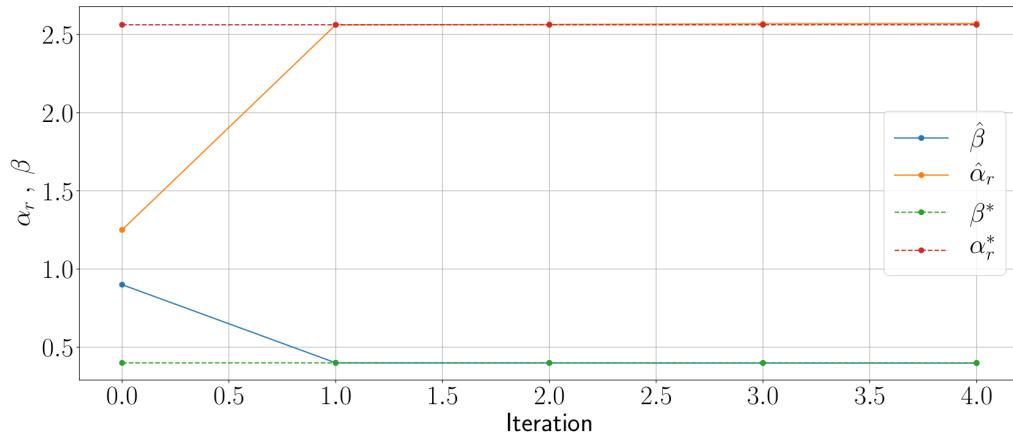
Performance and estimation of the α_r parameter

The simulated data used is obtained with the GG model with a β of 0.4, a SNR of 100dB, 5 sources, 5 observation channels, 4500 number of samples per channel and a batch size of 500. The problem is solved using the DGMCA algorithm, only using 50 epochs as it is done in the developed estimation algorithm, and once it is solved, a pair of $\hat{\alpha}_r$ and $\hat{\beta}$ are estimated. Those parameters are used to solve BSS problem and a C_A is calculated and plotted in the curve of the figure 4.9.6b.

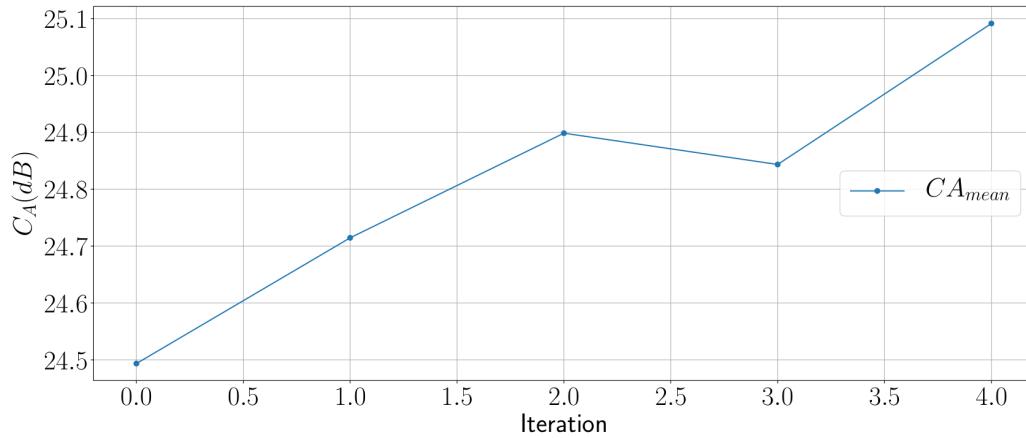
The α_r estimation algorithm is able to arrive to a close level of α_r^* in only one iteration. In this case, as the input data is simulated following a GG model, the gain of the algorithm is not too notorious. To validate the approach, in the next chapter, the estimation algorithm will run with real wavelet coefficients of astrophysics images and it can be seen, in the figure 5.2.4, a performance boost using only one iteration.

The automatic estimation of the α_r gives robustness to the DGMCA algorithm as there is no need to make a choice *a priori* of the parameter and the performance is increased. The main drawback is that it is slower as it needs to run the algorithm over some iteration to have a rough estimation of the sources from where to estimate the β of the GG model.

³Note that the GMCA algorithm used the same thresholding strategy and used the AMCA weight strategy to deal with partially correlated sources. This modifications, that proved to give better results, were added in order to have a honest comparison between the methods.



(a) Comparison between the $\hat{\alpha}_r$ and $\hat{\beta}$ parameter estimations during the iterations and the α_r^* and β^* which are the ones used to simulate the input data.



(b) Performance evaluation for each value of α_r corresponding to the same iteration in the figure 4.9.6a.

Figure 4.9.6: Convergence of the estimation of $\hat{\alpha}_r$ and $\hat{\beta}$, and its corresponding DGMCA performance using the current estimation.

*“El hombre no mate al hombre
 Ni pelee por fantasía
 Tiene en la desgracia mía
 Un espejo en qué mirarse
 Saber el hombre guardarse
 Es la gran sabiduría.”*

El gaucho Martín Fierro - José Hernández

5

Distributed Sparse BSS in a Transformed Domain

MOST OF THE TIME, THE OBSERVATIONS ARE IN A DOMAIN WHERE THE SIGNALS ARE NOT SPARSE. This difficulty has to be overcome by the DGMCA algorithm. Or in a wider way, one should be able to choose the domain in which to impose sparsity that might or might not coincide with the observational domain. That is problem this chapter will address.

5.1 DGMCA IN A TRANSFORMED DOMAIN

Following the cost function in equation (2.5) it is possible to see that the change of domain of sparsity only affects the sparsity inducing term. The data fidelity term and the mixing matrix constraint are left untouched. Going through the optimization algorithm, the minimization with respect to \mathbf{A} remains the same. The optimization of \mathbf{S} changes on the no differentiable term which is evidently the sparsity inducing term. The algorithm could be simply changed in the proximal operator of the sparsity inducing term which means to do the thresholding in the transformed domain. This accounts to change line 17 from algorithm 9 to:

$$\hat{\mathbf{S}}_j^{(k+1)} \leftarrow Th_{\Lambda_j^{(k)}} \left(\tilde{\mathbf{S}}_j^{(k+1)} \Phi^{tr} \right) \Phi \quad (5.1)$$

where Φ^{tr} is the analysis operator that decomposes the source by a series of wavelet coefficients

attached to different atoms in the wavelet dictionary and Φ is the synthesis operator which reconstructs the original signal by assembling the wavelet coefficients and the atoms. The wavelet transform considered will be an orthogonal transform meaning that it verifies $\Phi^{tr}\Phi = \Phi\Phi^{tr} = \mathbf{I}$, the reconstruction property and the tight frame property¹ respectively. In practice we will use an approximation that will be described in section 5.1.1.

Although nowadays there are fast and efficient implementations of different wavelet transforms, the use of analysis and synthesis operators as stated in equation (5.1) makes the DGMCA algorithm less computationally efficient. For each batch in each epoch there is a use of both operators.

A simpler alternative exist in order to avoid the use of the operators in each iteration. It is possible to transform the dataset before the start of the algorithm and to have all the terms in the transformed domain, where the sparsity condition holds. The DGMCA algorithm is ran with the transformed data and when it finishes the data is transformed into its original domain.

Defining $\mathbf{X}_\Phi = \mathbf{X}\Phi^{tr}$ and $\mathbf{S}_\Phi = \mathbf{S}\Phi^{tr}$ we can re-parametrize the cost function of equation (2.5) into:

$$\hat{\mathbf{A}}, \hat{\mathbf{S}}_\Phi = \arg \min_{\mathbf{A}, \mathbf{S}_\Phi} \frac{1}{2} \|\mathbf{X}_\Phi - \mathbf{A}\mathbf{S}_\Phi\|_F^2 + \|\Lambda \odot \mathbf{S}_\Phi\|_p + i_{\mathbf{Y}: \|\mathbf{Y}^k\|_2=1, \forall k}(\mathbf{A}) \quad (5.2)$$

and use it as the input of the DGMCA algorithm. In this way we will maintain the core of the DGMCA algorithm untouched and the main difference will be the addition of two layers, one before the input and one after the output. The first one to transform the observation matrix, and the second one to synthesize the result into the original domain.

In practice, each observational channel is a 2D image that will have to be reshaped into a 1D array that will be a line of the observation matrix \mathbf{X} . In one hand, if the incoming channels are in the sparse domain, the reshaping is trivial, as any reshaping will work fine. In the other hand, if we have to transform the domain, some attention must be taken as we are still considering that the images are large scale. This means that we cannot transform the whole image at once as it is impossible to hold it entirely in memory. This fact oblige us to transform different batches and put them together after.

The strategy taken can be seen in figure 5.1.1 considering a undecimated wavelet transform that will be described in section 5.1.1. The size of the batches has to be big enough (and square as possible) so that the border effects do not hinder the separation process but small enough to be managed by the available computer resources. Each batch is transformed into J levels of decompositions and a *Coarse Resolution*, which is a smoothed version of the original image. The wavelet coefficients are flattened into a 1D array and concatenated one batch after the other as in figure 5.1.1. The *Coarse*

¹We have a tight frame when for the generalized Parseval relation: $a_1\|\alpha\|_2 \leq \|\alpha\Phi^{tr}\|_2 \leq a_2\|\alpha\|_2$ with $0 < a_1 \leq a_2 < +\infty$ it verifies that $a_2 = a_1 = a$ which in that case $\Phi\Phi^{tr} = a\mathbf{I}$. If $a = 1$ we have the Parseval relation. For more information on frames and wavelets the reader is redirected to [19].

Resolution coefficients are not included in the 1D array as they do not show a sparse representation. We can omit them for the separation process as they are not discriminant.

One of the hypothesis taken to validate the approach in this last paragraph is that the wavelet used has a finite support as for example, the *starlets*. If a non local decomposition would have been used, for example, the Fourier transform, the approach taken would be invalid.

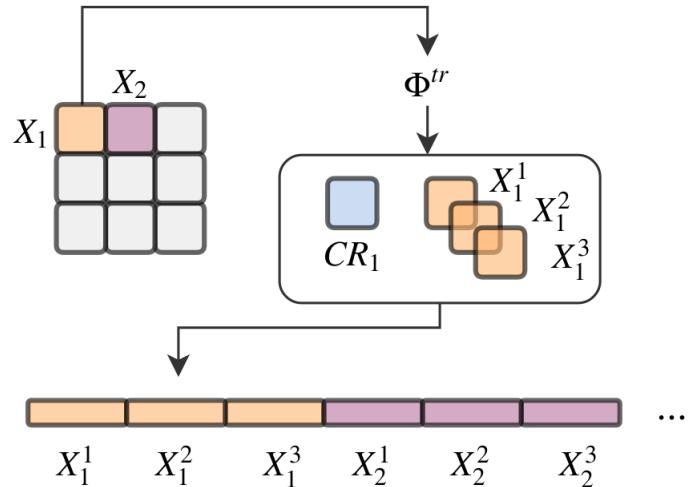


Figure 5.1.1: Analysis scheme for one observational channel using an undecimated wavelet transform with 3 levels of decomposition.

Once all the observation channels have been turned into flat arrays showing a sparse representation they can be stacked into a 2D matrix that will be the \mathbf{X} matrix. This element will be fed to the DGMCA algorithm that will output a mixing matrix \mathbf{A} and a transformed source matrix \mathbf{S}_Φ .

5.1.1 ISOTROPIC UNDECIMATED WAVELETS: THE STARLETS

In practice we will be using the isotropic undecimated wavelets as in most cases the images used for astronomical purposes are more or less isotropic. Another advantage of these wavelets is that they can be implemented in a fast and efficient way, being prepared for large datasets. The undecimated wavelet transform maintains the property of translation-invariance making it very useful for data analysis. The fact of having a finite support, being localized in space, makes them useful for our purposes. For more information about the starlets (or other type of wavelets) the reader is invited to consult the chapter 3 and specifically section 3.5 of the reference [19].

The related pair of symmetrical filters used is the following one:

$$\begin{aligned} h_{1-D}[k] &= [1, 4, 6, 4, 1]/16, \quad k = -2, \dots, 2, \\ h[k, l] &= h_{1-D}[k]h_{1-D}[l], \\ g[k, l] &= \delta[k, l] - h[k, l], \end{aligned} \tag{5.3}$$

where $\delta[0, 0] = 1$ and $\delta[k, l] = 0, \forall (k, l) \neq (0, 0)$. The wavelet coefficients are calculated as the difference from two resolutions:

$$w_{j+1}[k, l] = c_j[k, l] - c_{j+1}[k, l]. \tag{5.4}$$

Each resolution is calculated as follows:

$$c_{j+1}[k, l] = (\bar{h}^{(j)} \bar{h}^{(j)} * c_j)[k, l] \tag{5.5}$$

where $h^{(j)}[l] = h[l]$ if $l/2^j$ is an integer and 0 otherwise. For example having,

$$h^{(1)} = [\dots, h[-2], 0, h[-1], 0, h[0], 0, h[1], 0, h[2], \dots]. \tag{5.6}$$

In one dimension the equation (5.5) can be rewritten as:

$$c_{j+1}[l] = (\bar{h}^{(j)} * c_j)[l] = \sum_k h[k] c_j[l + 2^j k] \tag{5.7}$$

When advancing into a scale j , we obtain a new subband w_j that has the same dimension of the original image. In practice, equation (5.7) is used for the 2D images even though it is a 1D equation. As the filter h is separable it is possible to filter first for each row and then do it for each column giving rise only to 1D operations. The efficient algorithm used for the analysis is the one known as *à trous* ("with holes" in french). The name is inspired by the zeros left between the samples as seen in equation (5.6) and is widely used for undecimated wavelet transforms.

The reconstruction scheme is simply obtained by a co-addition of all wavelet scales and the final smooth subband as in the following equation:

$$c_0[k, l] = c_J[k, l] + \sum_{j=1}^J w_j[k, l] \tag{5.8}$$

An example of a starlet transform can be seen in the figure 5.1.2.

Considering Property 1 from section 3.5 from [19], this wavelet transform respects the perfect re-

construction property, from where $\Phi^{tr}\Phi = \mathbf{I}$, but it is not a the tight frame decomposition, meaning $\Phi\Phi^{tr} \neq \mathbf{I}$. Nevertheless, this redundant type of wavelet transform presents a diagonally dominant Gram matrix ($\Phi\Phi^{tr}$) it can be approximated by a tight frame.

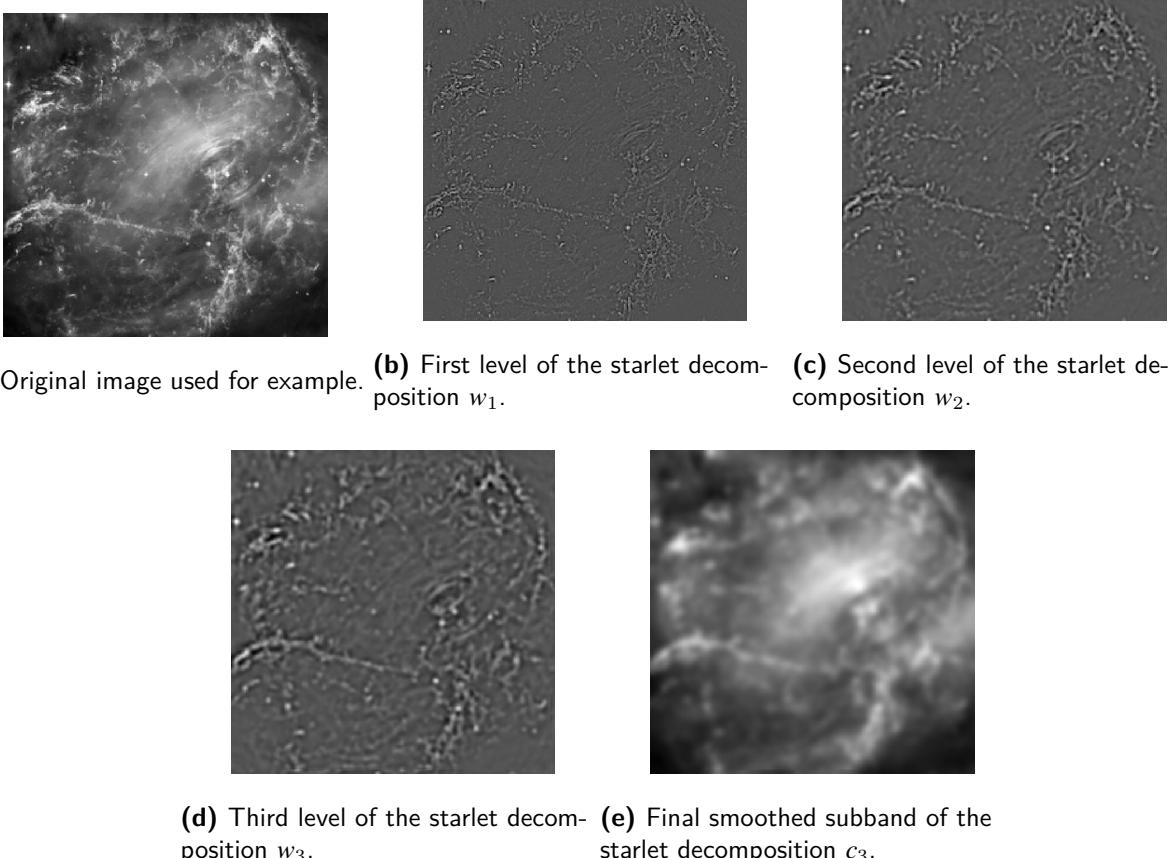


Figure 5.1.2: Starlet transform of an astrophysical image. The addition of the last four figures, from (b) to (e), reproduce exactly the first image (a).

5.2 NUMERICAL EXPERIMENTS

5.2.1 INPUT DATA

The sources used consist of the five images from figure 5.2.1. They are mixed using a known mixing matrix that is issued from power laws as it happens often in the astrophysical domain. After the mixing has been done, WGN is added so that the SNR is the desired one. As the mixing matrix is known, it is possible afterwards to calculate the performance metric C_A . Examples of the real mixtures used

can be seen in figure 5.2.2.

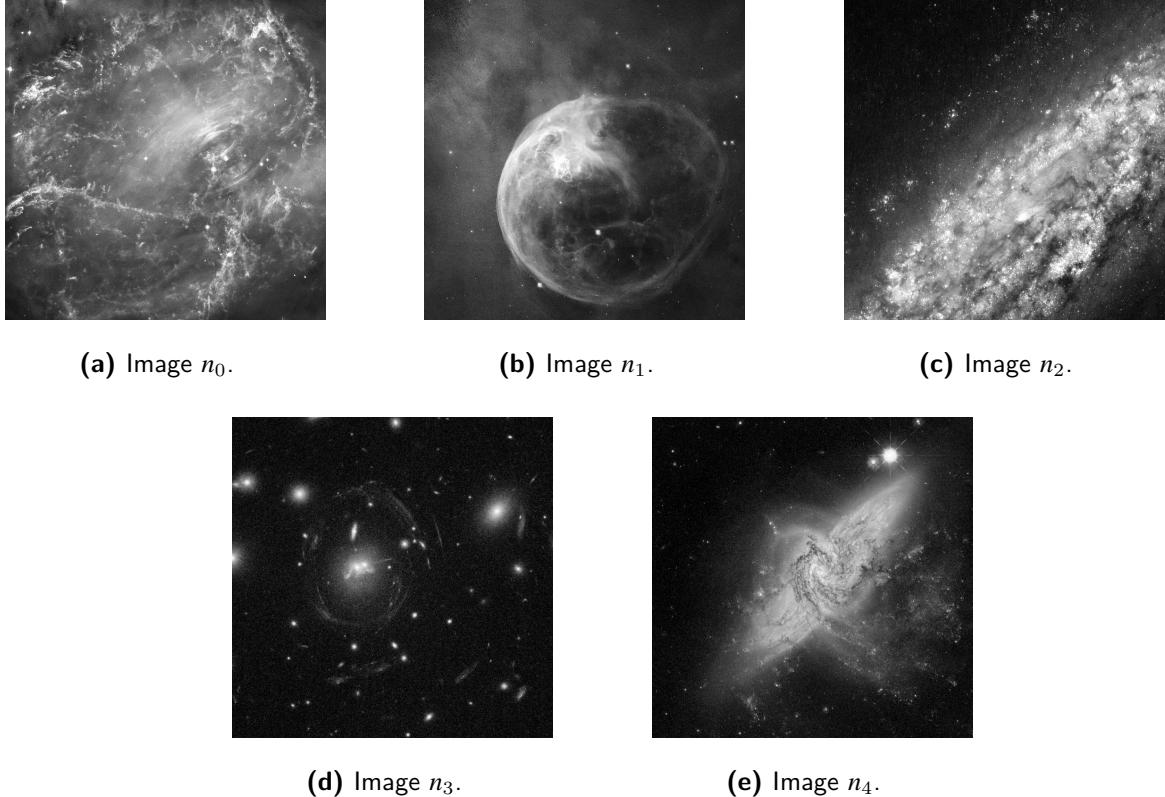
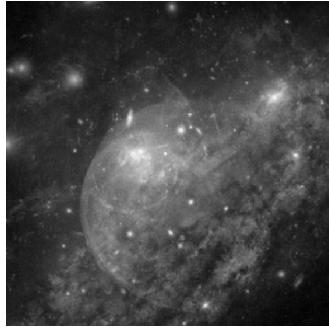


Figure 5.2.1: Images taken from the Hubble Space Telescope used to test the DGMCA algorithm in a transformed domain.

5.2.2 RESULTS

The experiments consist evidently in solving the BSS problem. Firstly when varying the batch size and the line group size. Later, the influence of the automatic tuning of the α_r parameter on the performance of the algorithm is tested. In both experiments there are some common settings that are the next ones:

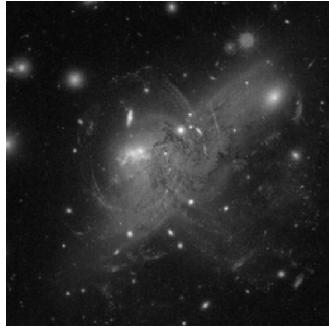
- Sources: Images from figure 5.2.1.
- SNR = 60dB
- Number of sources : 5



(a) Observation channel n_0 .



(b) Observation channel n_1 .



(c) Observation channel n_2 .



(d) Observation channel n_3 .

Figure 5.2.2: Examples of observation channels chosen randomly between the test set.

- Wavelet J : 2 level starlets
- Number of epochs : 100

Performance test

The performance of the DGMCA and the xDGMCA algorithms for different batch sizes and column group sizes is evaluated. In this case the mixing matrix is the same for all the batch sizes and repetitions. The details of the test are the following ones:

- Value of α_r : Automatic tuning
- Number of observation channels : 250
- Number of samples : 32768 (128×128 images with $J = 2$)

- Sizes of batches : [256, 512, 1024, 2048, 4096, 8192]
- Line group sizes : [25, 50]
- Number of repetitions : 2

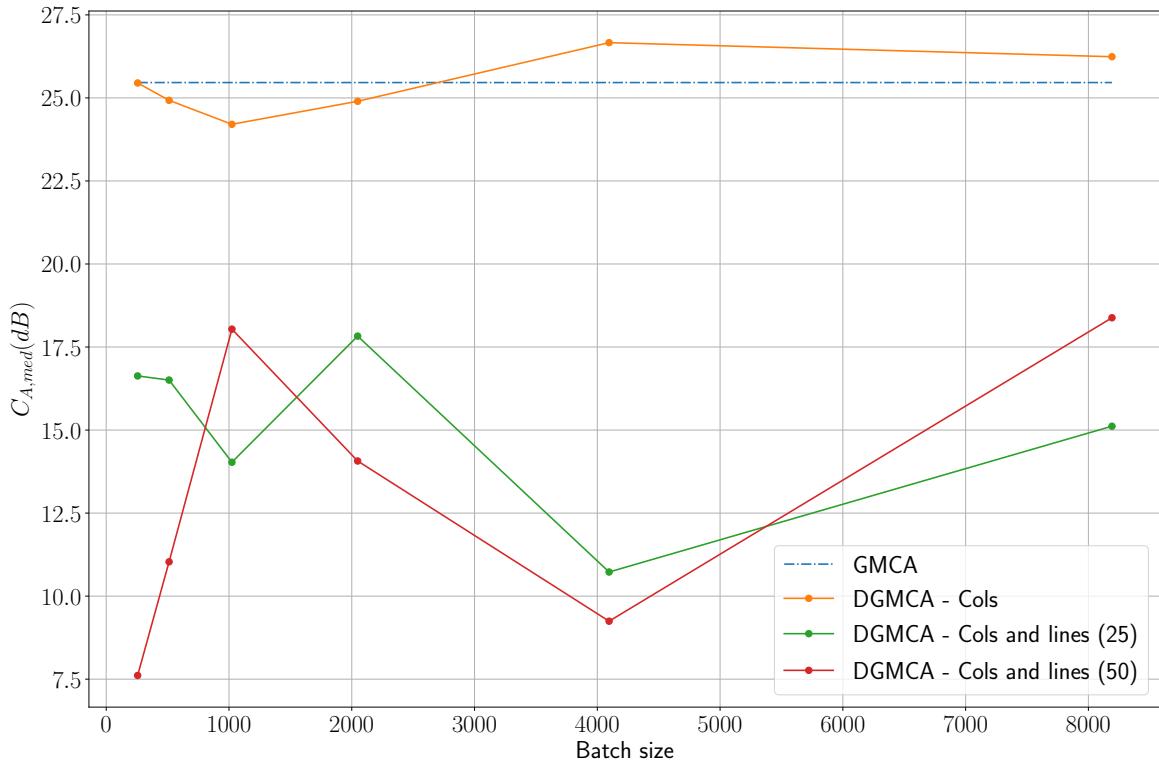


Figure 5.2.3: Performance of the GMCA, DGMCA and xDGMCA algorithms when varying the batch size and the line group size.

The performance of the DGMCA is maintained in a similar level as the one in the GMCA. This is another validation of the approach considered in section 4 for the fusion of the mixing matrix estimators and also for the DGMCA algorithm working in the transformed domain with astrophysical images.

The xDGMCA does not show outstanding results having a considerable gap between its performance and the one of the GMCA and DGMCA. In addition, it is showing excessive performance variance for the different group sizes.

Influence of the α_r tuning

The experiment consist of fixing the observation matrix and running the DGMCA algorithm for different values of α_r comparing it with the GMCA and the DGMCA with automatic α_r tuning. The

settings of the test are exhibited next:

- Number of observation channels : 50
- Values of α_r^o : [0.05, 0.5, 5, 50]
- Number of samples : 8192 (64×64 images with $J = 2$)
- Size of batch : 1024
- Number of repetitions : 11

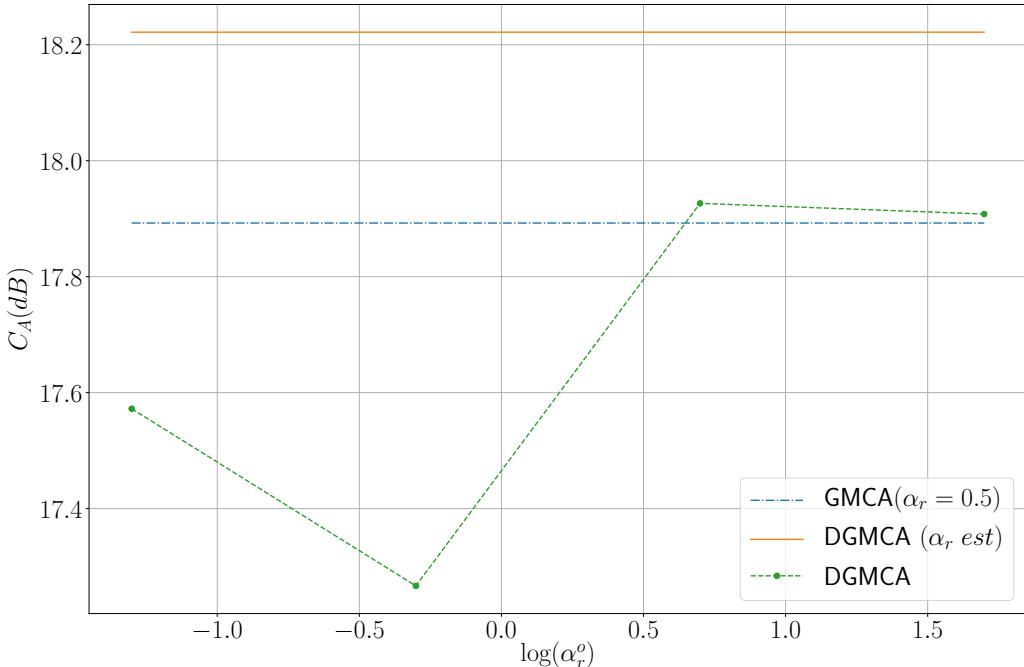


Figure 5.2.4: Performance comparison, while varying α_r^o , of the original GMCA algorithm (with α_r^o fixed to 0.5), the DGMCA segmenting only the columns with and without the estimation of the α_r . The metric used is the $C_{A,mean}$ and the input data is the one seen in figure 5.2.1.

Not only the DGMCA performance is at least boosted around 0.3dB but the algorithm gains stability as there is no need to fix the α_r parameter that can have influence in the algorithm performance.

The gap between the GMCA and the DGMCA with α_r estimated can be explained by the fact that the first algorithm was ran with a fixed α_r which apparently was not very adapted to the sources used.

*“Y les doy estos consejos
Que me han costado alquirloros,
Porque deseo dirijirlos,
Pero no alcanza mi cencia
Hasta darles la prudencia
Que precisan pa seguirlos.”*

El gaucho Martín Fierro - José Hernández

6

Conclusion and Perspectives

6.1 CONCLUSIONS AND CONTRIBUTIONS

We introduced a new method for dealing with large-size Blind Source Separation problems. The new method built concepts from the re-known method GMCA for BSS. The last mentioned algorithm proved to be successful in many application, for example in the estimation of the CMB map giving astonishing results [5], and extensions from it are a current topic of research in the CosmoStat laboratory. The arising of massive international projects like SKA that will issue a massive deluge of data, proves the need for distributed methods of this type.

The developed DGMCA algorithm has proven its performance firstly with simulated data, following the aimed approximatively sparse signals, giving satisfactory results even with a highly distributed scheme. The performance achieved was at the same level of our benchmark, the GMCA algorithm, which does no distribution at all. Secondly, the algorithm was redeveloped to work with natural images as sources, in practice they were astrophysical observations from the Hubble Space Telescope. The results were comforting as the DGMCA achieved again the same level of performance than the GMCA algorithm.

The algorithm is available online on a github¹ repository available for the community in the spirit of reproducible research. A fully distributed version coded in C++ exists and was used for the com-

¹<https://github.com/tobias-liaudat/DGMCA>

putational time gain test. This last version was mostly coded by Jérôme Bobin and was not made public.

The complexity analysis resulted in a gain of computational time by a factor of T that is the number of cores used in the distribution architecture. So, the DGMCA not only allows to work with large scale data, but it speeds up the process with its distribution scheme.

The obtained results validate the proposed method for matrix aggregation mixing the concepts of Riemannian geometry and optimization. One take-home message is the importance of not neglecting the geometry of the data used. The Information Geometry field provides many advances ready to be used as well as the popular Optimal Transport theory.

The final methods have proven to be robust with a new set of heuristics and methods that emphasize this aimed characteristic. Not to mention that they are keeping the constraint of allowing the distribution of the calculations. In the final algorithm, most of the parameters are automatically tuned in function of the input data. A new approach to calculate the alpha thresholding parameter was introduced given the generalized Gaussian behavior of approximatively sparse signals. Results show that the xDGMCA algorithm is not as robust as the DGMCA and the second one should be choose if the opportunity arises.

The obtained results are interesting in an engineering point of view as the data exchange in the fusion of the estimators is practically insignificant to the size of the input data. The sharing is only of estimated mixing matrices (and some other size-negligible parameters) that are orders of magnitude smaller than the observation of source matrix. This is taking out from the start any possible bottleneck in the data transfer of the distribution scheme. The costly calculations and the massive data does not need to be extremely powerful connected allowing an easier delocalisation of the processing units.

After the current ideas are developed, an extension of the DGMCA algorithm coined DecDGMCA will be presented. The method addresses the inverse problem that arises from radio-interferometric data where the BSS is tackled jointly with a deconvolution. Although it has not been thoroughly studied due to time constraints, the first formulations of its implementation can be found in this document in section 6.3.

The main ideas of this work were summed up to make an article that was accepted to the SPARS² 2019 (*Signal Processing with Adaptive Sparse Structured Representations*) conference that can be seen in annex B. It also included a poster session.

After the end of this thesis, Jérôme Bobin continued to work on the Fréchet mean with collaboration from Christophe Kervazo and Tobías Liaudat. The algorithm can be improved to be able to better

²<http://www.spars-workshop.org/en/index.html><http://www.spars-workshop.org/en/index.html>

manage outliers and therefore arrive to superior performances. It is simple to see how the mean and the median manage outliers in a set of real numbers, being the second one robust in comparison with the first one. The same idea can be followed with the ℓ_2 and ℓ_1 norms. We can think of an analogy in a Riemannian manifold, with the Fréchet mean and what is to be done, the Fréchet median. By changing the formulation of the cost function used to optimize in the manifold, going from the square of the geodesic distance to a regularized (to maintain the differentiability at the origin) ℓ_1 norm, it is possible to be more robust to outliers. Preliminary results show that this new version is able to work in parallel and obtain better performances than GMCA. A paper is expected to be written with this ideas, where a first version is expected for August 2019.

Finally, ~~without being pretentious~~, the developed methods are just a step in the stairway that is needed to achieve success in such ambitious projects like SKA. There is still much work to be done but I will hand over the baton at this point, with the hope, that it was a worthwhile effort.

6.2 PERSPECTIVES

There are still some ideas flying over the cuckoo nest that have the potential to disembark to good results.

Following the recent results in [7] with a new method coined bGMCA, the original update strategy used can be assimilated in the distributed scheme of DGMCA allowing to enhance the actual performance results.

In the same spirit of [7], the use of a refinement step can be introduced to the DGMCA algorithm. The non-convex optimization is firstly tackled by the method described in this work, robustly approaching a critical point. In order to be close enough to launch a PALM type algorithm with a proven convergence to a critical point of the non-convex objective function. This refinement step can be done using the results of [37] that developed the asynchronous PALM algorithm that can be parallelized.

There is still work on the implementation part, where the goal is to yield a fully distributed DGMCA toolbox in C language. Not to mention, the tests and validation of the extension part applied to radio-interferometric data with the proposed method DecDGMCA are to be done.

6.3 EXTENSION TO THE JOINT DECONVOLUTION AND BSS

The deconvolution is an unavoidable step when dealing with data issued from astronomical radio-interferometers such as LOFAR, seen in figure 6.3.1, or SKA. The nature of the observations, being masked³ and in a transformed domain, demand powerful reconstruction methods to be able to synthesize the desired images. The task becomes more complex when the goal is to separate sources present in these specific observations.

Although the developed method, DGMCA, is interesting, when it comes to real world applications, such as astrophysics, it is limited due to the nature of the data. It is not an exception to work with convolved data, where the convolution kernel can be the PSF⁴ of an instrument or a decimation mask, or both.

Having its importance stated, the tackling of these two problems, deconvolution and BSS, in a jointly approach can yield better results than treating them separately. Based on the work [8], which effectively does this, we will apply the concepts of DGMCA to it to obtain a new method that is able to cope the deconvolution BSS in a distribution framework as the one in DGMCA.

6.3.1 BACKGROUND

RADIO-INTERFEROMETERS

A radio-interferometer is an array of radio antennas used for astronomical observations where a discretely-sampled single telescope of a very large aperture can be simulated. The aperture is the diameter of the signal collecting region and is one of the decisive characteristic that determines the resolution and the limiting magnitude of the instrument. The resolution is related with the smallest distance between the objects that the instrument is able to resolve or distinguish. The limiting magnitude relates to the brightness of the faintest object that is visible by the instrument. The LOFAR interferometer can be seen in figure 6.3.1.

Bigger apertures are not possible to achieve by single instruments mainly due to construction and material constraints. Nevertheless, in aperture synthesis it is possible to reconstruct images with high resolution by the using of small antennas with smaller resolution.

Astronomical radio interferometers can be used to achieve astonishing angular resolutions when used together as a network. Considering two antennas operating at a frequency ν , the combination or correlation of the two measurements (product operation) defines a measure of the spatial coherence function of the source that we will call from now on complex *visibility* $\bar{\mathcal{V}}_\nu$.

³Decimated, incomplete

⁴Point Spread Function: The response of an imaging system to a point source. It is related to the blurring effect of the imaging instrument which limits its resolution.



(a) Aerial photograph of LOFAR radio telescope.



(b) Image showing the various locations of the International LOFAR Telescope sites in Europe.

Figure 6.3.1: Images of the LOFAR project taken from <https://www.skatelescope.org>

Considering a radio interferometer in a 3 dimensional coordinate system, we can define the following quantities:

- A unitary pointing direction in the sky \vec{s}_0 .
- An orthonormal reference system $\mathcal{R}_{(x,y,z)}(\vec{e}_x, \vec{e}_y, \vec{e}_z)$, the direct space, where the two antennas are placed forming the \vec{b} vector, the interferometric base, that joins both antennas bases.
- An orthonormal reference system $\mathcal{R}_{(u,v,w)}(\vec{u}, \vec{v}, \vec{w})$, the visibility space, where \vec{u} points to the East, \vec{v} points to the North and \vec{w} points in the direction of \vec{s}_0 .
- A unitary vector \vec{s} ($\vec{s} \neq \vec{s}_0$) pointing to any direction (θ, ϕ) . The vector \vec{s} can be expressed in $\mathcal{R}_{(u,v,w)}$ as $\vec{s} = \begin{bmatrix} l \\ m \\ n \end{bmatrix} = \begin{bmatrix} \sin \theta \cos \phi \\ \sin \theta \sin \phi \\ \cos \theta \end{bmatrix}$. The components (l, m, n) allow to uniquely define the vector \vec{s} .

The coordinate systems used are the usual ones, following the textbook [35] and the work [33]. The right-handed coordinate systems (l, m, n) and (u, v, w) can be considered as Fourier pairs. The first one describing the source brightness distribution and the second one the visibilities. These two measures are related as it can be seen in equation (6.1). An illustration of the different systems can be observed in figure 6.3.2.

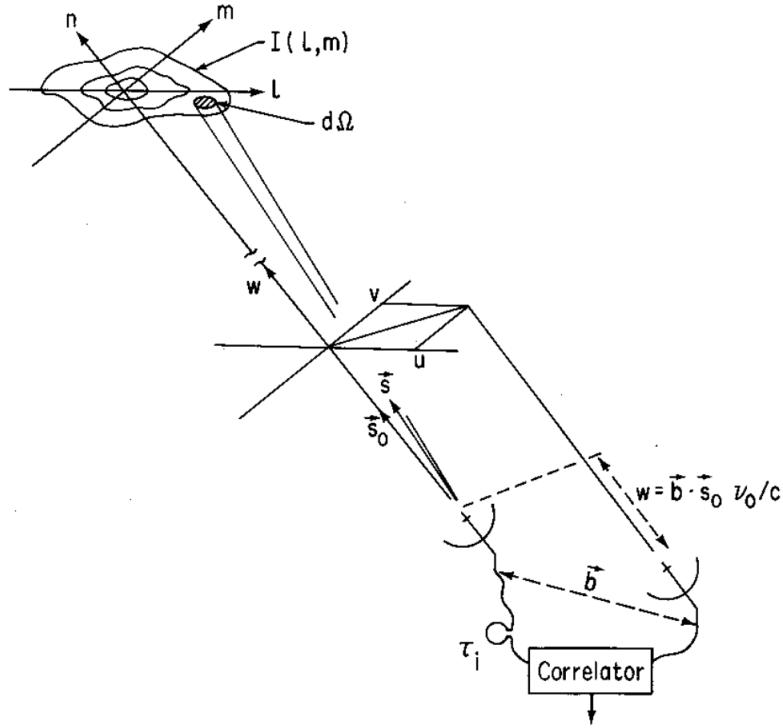


Figure 6.3.2: Illustration of the different coordinate systems given a pair of antennas. Image taken from [35].

The Van Cittern-Zernicke theorem, analyzed in page 595 of [34], allows to relate the surface brightness at a frequency ν of a source \mathcal{B}_ν to its spatial coherence function $\tilde{\mathcal{V}}$ by a Fourier transform \mathcal{F} :

$$\tilde{\mathcal{V}}_\nu = |\mathcal{V}_\nu| e^{i\Phi_\nu} = \mathcal{F} [\mathcal{E}\mathcal{B}_\nu] \quad (6.1)$$

with the following notation:

$$\tilde{\mathcal{V}}_\nu(u, v, w) = \int_{\Omega_{sky}} \mathcal{E}(l, m) \mathcal{B}_\nu(l, m) e^{-j2\pi \vec{b} \cdot (\vec{d} - \vec{d}_0)} d\Omega \quad (6.2)$$

where $|\mathcal{V}|$ and Φ are the amplitude and the phase of the visibility respectively, \mathcal{E} is the response of the antenna and $d\Omega$ is a solid angle differential.

Given the useful Van Cittern-Zernicke theorem, a way to calculate the source source brightness distribution \mathcal{B}_ν , is to use the inverse Fourier transform of the visibility map $\tilde{\mathcal{V}}_\nu$. Each different pair of antennas, with a \vec{b}_i , gives one element of the visibility map at (u_i, v_i, w_i) . In consequence, the number of antennas and its geometric disposition will play a fundamental role in the coverage of the uv plane (visibility plane). The richer the uv plane, the better the source brightness distribution reconstruction will be. This sampling of the uv plane and the subsequent reconstruction to obtain \mathcal{B}_ν is what *aperture synthesis*.

synthesis is about, roughly speaking. An example uv coverage of the LOFAR station can be seen in figure 6.3.3.

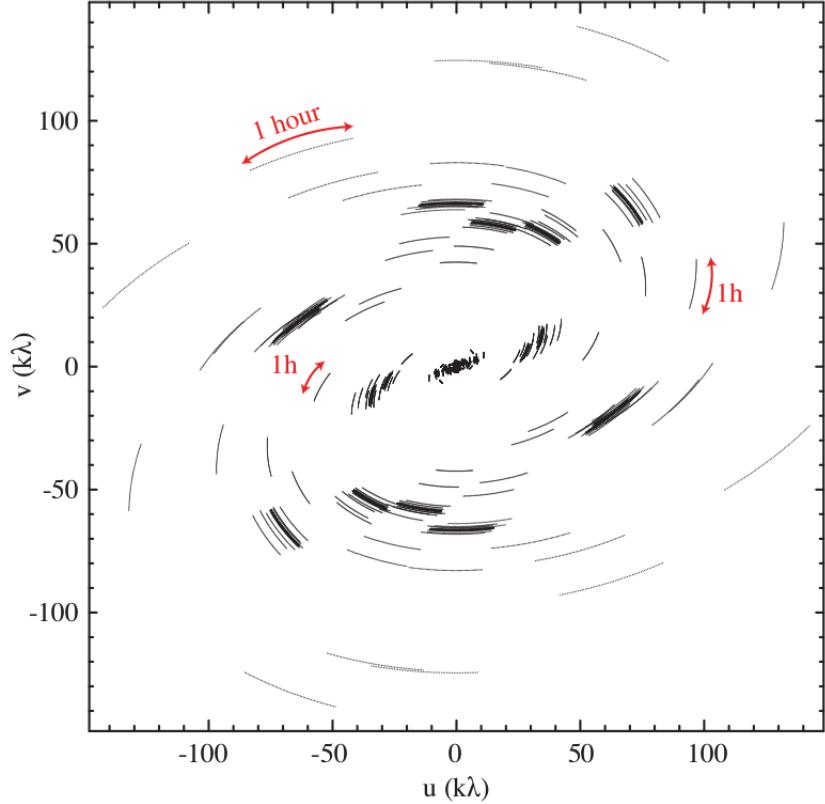


Figure 6.3.3: Distribution of uv spatial frequencies for the LOFAR instrument given an observational time of 1 hour of a source at the zenith in narrow band. Each correlation between a pair of antennas is represented by a single point in an instant. However, due to the movement of the Earth in the hour of observation, the points move to trace segments of ellipses. The axes are graded in units of wavelength with respect to the center phase of the instrument. Image taken from [33].

6.3.2 DBSS - DECONVOLUTION BLIND SOURCE SEPARATION

Solving jointly a BSS problem from incomplete measures in a transformed domain is challenging. The problem of restoring images from incomplete observations has been investigated in the framework of Compressed Sensing (CS). It has proven good results and has been used in several applications like the reconstruction of MRI (Magnetic Resonance Imaging) images. Nevertheless, it does not tackle the problem of BSS. In order to do so, it would be necessary to reconstruct the images given the available samples and then use a BSS algorithm. Following the work of [8], the problems can be solved jointly in what the author coined *Deconvolution Blind Source Separation* (DBSS). If the convolution operator

applied to the mixture is a subsampling operator we fall into the case of the CS and BSS framework but the method developed can work on different convolution matrices. For example, the deblurring and BSS, where the images were convoluted by the PSF of a given instrument.

The proposed imaging model is as follows:

$$\forall \nu \in \{1, 2, \dots, m\}, \mathbf{y}^\nu = \mathbf{h}^\nu * \mathbf{x}^\nu = \mathbf{h}^\nu * \left(\sum_{r=1}^{N_{src}} A_{\nu,r} \mathbf{s}^r \right) + \mathbf{n}^\nu = \mathbf{h}^\nu * (\mathbf{a}^\nu \mathbf{S}) + \mathbf{n}^\nu \quad (6.3)$$

where ν is an observational channel in the LMM or a given frequency, from the m available. It can be seen the degradation of the mixture by the linear operator $\mathbf{H} \in \mathbb{R}^{m \times t}$ which is supposed to be known. In practice, the subsampling masks are known from the instrument and in the case of the blurring effect of the PSF there are several methods for doing accurate estimations like in [36]⁵. In Fourier domain the last equation can be expressed as:

$$\forall \nu \in \{1, 2, \dots, m\}, \tilde{\mathbf{y}}^\nu = \tilde{\mathbf{h}}^\nu \odot \tilde{\mathbf{x}}^\nu = \tilde{\mathbf{h}}^\nu \odot \left(\sum_{r=1}^{N_{src}} A_{\nu,r} \tilde{\mathbf{s}}^r \right) + \tilde{\mathbf{n}}^\nu = \tilde{\mathbf{h}}^\nu \odot (\mathbf{a}^\nu \tilde{\mathbf{S}}) + \tilde{\mathbf{n}}^\nu \quad (6.4)$$

Specifically, given only one measure of visibility in the Fourier domain, it depends on one mask element, a line of the mixing matrix, a column of the source matrix and an element of noise :

$$\tilde{Y}_{\nu,k} = \tilde{H}_{\nu,k} \mathbf{a}^\nu \tilde{\mathbf{s}}_k + \tilde{N}_{\nu,k} \quad (6.5)$$

The author in [8] follows similar guidelines as in GMCA for the BSS problem. The regularization of the inverse problem will be the imposition of the sparsity of the source matrix in a given dictionary Φ . At first, the transform will be considered to be (bi-)orthogonal although in practice it can be relaxed to a transform with a Gram matrix ($\Phi \Phi^{tr}$) that is diagonally dominant.

The estimation problem can be written as the following non-convex optimization:

$$\hat{\mathbf{A}}, \hat{\mathbf{S}} = \arg \min_{\mathbf{A}, \mathbf{S}} \frac{1}{2} \sum_{\nu=1}^m \sum_{k=1}^t \left\| \tilde{Y}_{\nu,k} - \tilde{H}_{\nu,k} \mathbf{a}^\nu \tilde{\mathbf{s}}_k \right\|_2^2 + \sum_{r=1}^{N_{src}} \lambda_r \left\| \mathbf{s}^r \Phi^{tr} \right\|_p + i_{\mathbf{Y}: \|\mathbf{Y}_l\|_2=1, \forall l}(\mathbf{A}) \quad (6.6)$$

The method proposed by [8] is interesting and it gives good results. Nevertheless, it is not adapted to work with large-scale data.

6.3.3 DEC-DGMCA - DISTRIBUTED JOINT DECONVOLUTION AND BSS

The new method will follow the spirit of the GMCA algorithms. The optimization will be done in an alternate way as the Alternating Least Square (ALS) procedure to tackle the multi-convex optimization.

⁵The laboratory CosmoStat is actively working on the PSF estimation for the Euclid mission.

We can define a new function corresponding to the data fidelity term of the objective function in equation (6.6) that is expressed as follows:

$$f(\mathbf{a}^\nu, \tilde{\mathbf{s}}_k) = \frac{1}{2} \sum_{\nu=1}^m \sum_{k=1}^t \left\| \tilde{Y}_{\nu,k} - \tilde{H}_{\nu,k} \mathbf{a}^\nu \tilde{\mathbf{s}}_k \right\|_2^2. \quad (6.7)$$

ESTIMATION OF S

To obtain the first estimation of $\tilde{\mathbf{s}}_k$ we have to let the deviation of the function $f(\mathbf{a}^\nu, \tilde{\mathbf{s}}_k)$ with respect to $\tilde{\mathbf{s}}_k$ vanish, this is $\frac{\partial f(\mathbf{a}^\nu, \tilde{\mathbf{s}}_k)}{\partial \tilde{\mathbf{s}}_k} = 0$. In consequence we arrive to a closed form expression that is the following one:

$$\forall k \in \{1, 2, \dots, t\}; \quad \tilde{\mathbf{s}}_k = \left(\sum_{\nu=1}^m (\tilde{H}_{\nu,k} \mathbf{a}^\nu)^{tr} (\tilde{H}_{\nu,k} \mathbf{a}^\nu) \right)^{-1} \sum_{\nu=1}^m (\mathbf{a}^\nu)^{tr} \tilde{H}_{\nu,k} \tilde{Y}_{\nu,k}, \quad (6.8)$$

that can be rewritten as:

$$\forall k \in \{1, 2, \dots, t\}; \quad \tilde{\mathbf{s}}_k = \left(\mathbf{A}^{tr} \text{diag} \left((\tilde{\mathbf{h}}_k)^2 \right) \mathbf{A} \right)^{-1} \mathbf{A}^{tr} \text{diag} \left(\tilde{\mathbf{h}}_k \right) \tilde{\mathbf{y}}_k. \quad (6.9)$$

We will follow the proposition in [8] of a regularization of the matrix inversion in equation (6.9) to control the condition number of the system. Depending on the convolution operator used in the problem the matrix inversion can be delicate and from there arises the need of a regularization. Finally, adding the Tikhonov regularization to the least-square estimation we obtain:

$$\forall k \in \{1, 2, \dots, t\}; \quad \tilde{\mathbf{s}}_k = \left(\mathbf{A}^{tr} \text{diag} \left((\tilde{\mathbf{h}}_k)^2 \right) \mathbf{A} + \epsilon' \mathbf{I}_{N_{src}} \right)^{-1} \mathbf{A}^{tr} \text{diag} \left(\tilde{\mathbf{h}}_k \right) \tilde{\mathbf{y}}_k, \quad (6.10)$$

where $\mathbf{I}_{N_{src}}$ is a $N_{src} \times N_{src}$ identity matrix, ϵ' is the regularization parameter and the $\text{diag}(\cdot)$ is the diagonal operator that returns a diagonal matrix from a vector.

The minimization of the sparsity term is done with the proximal operator that in practice is, as already seen, the thresholding operator. It is possible to see in the following equation that the thresholding is done in the transformed space. The same strategy from section 5 can be used for going and coming back to the transformed domain. The approach is valid if the transform verifies the same characteristics from the cited section, like to be localized and be a tight frame (or a good approximation of it).

$$\forall r \in \{1, 2, \dots, N_{src}\}, \quad \forall j \in \{1, 2, \dots, T\}; \quad \mathbf{s}_{t_j}^r = \left(Th_{\lambda_r} \left(\mathbf{s}_{t_j}^r \Phi^{tr} \right) \right) \Phi \quad (6.11)$$

Before the thresholding it is necessary to do an inverse Fourier transform as the estimation of the source columns is done entirely in the Fourier space. Once the estimation of $\tilde{\mathbf{s}}_k$ is done for all k it is

possible to obtain \mathbf{S} from $\tilde{\mathbf{S}}$.

At this point, for the moment, we have no more option than using each line of the source matrix entirely to be able to apply the Fourier transform. *This transform is non local, meaning that we need to use each source $\tilde{\mathbf{s}}^r$ entirely to get \mathbf{s}^r* as follows:

$$\forall r \in \{1, 2, \dots, N_{src}\}, \mathbf{s}^r = \mathcal{F}^{-1}(\tilde{\mathbf{s}}^r) \quad (6.12)$$

After the thresholding step and before we continue to the estimation of the mixing matrix we have to come back to the Fourier space as the observations are made in the mentioned space.

$$\forall r \in \{1, 2, \dots, N_{src}\}, \tilde{\mathbf{s}}^r = \mathcal{F}(\mathbf{s}^r) \quad (6.13)$$

ESTIMATION OF A

Once the estimation of the whole source matrix is done, we will move on with the estimation of \mathbf{A} .

The optimization with respect to the data fidelity term is done by vanishing the derivation of $f(\mathbf{a}^\nu, \tilde{\mathbf{s}}_k)$ with \mathbf{a}^ν , this is $\frac{\partial f(\mathbf{a}^\nu, \tilde{\mathbf{s}}_k)}{\partial \mathbf{a}^\nu=0}$. The expression we arrive uses all the $\tilde{\mathbf{S}}$ matrix and is as follows:

$$\forall \nu \in \{1, 2, \dots, m\}; \mathbf{a}^\nu = \sum_{k=1}^t \tilde{H}_{\nu,k} \tilde{Y}_{\nu,k} (\tilde{\mathbf{s}}_k)^H \left(\sum_{k=1}^t (\tilde{H}_{\nu,k} \tilde{\mathbf{s}}_k) (\tilde{H}_{\nu,k} \tilde{\mathbf{s}}_k)^H \right)^{-1}, \quad (6.14)$$

where it can be expressed as well as:

$$\forall \nu \in \{1, 2, \dots, m\}; \mathbf{a}^\nu = \tilde{\mathbf{y}}^\nu \operatorname{diag}(\tilde{\mathbf{h}}^\nu) \tilde{\mathbf{S}}^H \left(\tilde{\mathbf{S}} \operatorname{diag}((\tilde{\mathbf{h}}^\nu)^2) \tilde{\mathbf{S}}^H \right)^{-1}. \quad (6.15)$$

Following the spirit of DGMCA we can segment each matrix, $\tilde{\mathbf{S}}$, $\tilde{\mathbf{Y}}$ and $\tilde{\mathbf{H}}$, in T group of columns, not necessarily contiguous. Thus, changing the estimation of the mixing matrix and being coherent with the notation, the estimation is as follows:

$$\forall \nu \in \{1, 2, \dots, m\}, \forall j \in \{1, 2, \dots, T\}; \mathbf{a}_{(j)}^\nu = \tilde{\mathbf{y}}_{t_j}^\nu \operatorname{diag}(\tilde{\mathbf{h}}_{t_j}^\nu) \tilde{\mathbf{s}}_{t_j}^H \left(\tilde{\mathbf{s}}_{t_j} \operatorname{diag}((\tilde{\mathbf{h}}_{t_j}^\nu)^2) \tilde{\mathbf{s}}_{t_j}^H \right)^{-1}. \quad (6.16)$$

Once all the T mixing matrices $\mathbf{A}_{(j)}$ are estimated the unit norm constraint on their columns has to be applied:

$$\forall r \in \{1, 2, \dots, N_{src}\}, \forall j \in \{1, 2, \dots, T\}; \bar{\mathbf{a}}_{(j),r} = \frac{\mathbf{a}_{(j),r}}{\|\mathbf{a}_{(j),r}\|_2} \quad (6.17)$$

Later, the mixing matrices $\mathbf{A}_{(j)}$ are aggregated using exactly the same formulation of the *Fréchet*

mean from section 4.3 with the weightings $\omega_{r,j}^{FM}$ (described afterwards) issuing \mathbf{A}_{FM} . After the aggregation is done, an epoch ends in the global iteration of the DecDGMCA algorithm.

HEURISTICS

The heuristics needed for the FM weighting and the thresholding strategy have to be slightly modified to account for the new framework.

The FM weight is calculated in the same domain where the mixing matrix is estimated, this is the Fourier domain. The inclusion of the convolution matrix in the weighting follows the arguments given in section 4.4.3, when doing a type of SNR estimation.

$$\omega_{r,j}^{FM} = \frac{\|\tilde{\mathbf{s}}_j\|_2^2 \|\tilde{\mathbf{h}}_j\|_2^2}{\sigma_{Y_j} \|\hat{\mathbf{A}}_{FM}^\dagger\|_F^2}, \quad \forall j \in \{1, \dots, T\}, \forall r \in \{1, \dots, N_{src}\} \quad (6.18)$$

Considering the thresholding, the same strategy as in section 4.4.1 is used. The only difference resides on the calculation of the noise standard deviation $\sigma_{N,r}$.

$$\lambda_r^{(i)} = K_\sigma \sigma_{N,r} + (\|\mathbf{s}^r\|_\infty - K_\sigma \sigma_{N,r}) e^{-i \alpha_r^c}, \quad \forall i \in \{1, \dots, N_{epoch}\} \quad (6.19)$$

Note that the norm is done on the sources when they are in its sparse domain as the thresholding is taking place there.

*“La ley es tela de araña,
y en mi ignorancia lo explico,
no la temo el hombre rico,
no la temo el que mande,
pues la rompe el bicho grande
y sólo enrieda a los chicos.”*

El gaucho Martín Fierro - José Hernández

A

Additional numerical experiments

Several tests were carried on in order to validate one or other approach and to be able to discriminate between different strategies concerning the algorithm heuristics.

A.1 NUMERICAL EXPERIMENTS

A.1.1 TEST: THRESHOLDING STRATEGY

This test performs a comparison between different thresholding strategies. The thresholding strategy 1 (Th_1) consists of a high value that goes decreasing at each epoch until it reaches the value of $K_{\sigma}\sigma$. The strategy uses the percentile of the sources estimation that are over $K_{\sigma}\sigma$ and goes decreasing linearly as the epochs go by. The thresholding strategy 2 (Th_2) is the one adopted and has already been explained in section 4.4.1. The weighting strategy for the Fréchet mean is the one adopted for the DGMCA.

Settings:

- Number of observation channels : 20
- Number of sources: 5
- SNR: 60dB

- Values of β : 0.5
- Number of samples : 2000
- Size of batches : [200, 250, 400, 500, 1000, 2000]
- Number of repetitions : 25
- Number of epochs: 100
- Use of AMCA weights: Yes

It is clear seen on figure A.1.1 that the second strategy outperforms the first mostly when batch sizes are small.

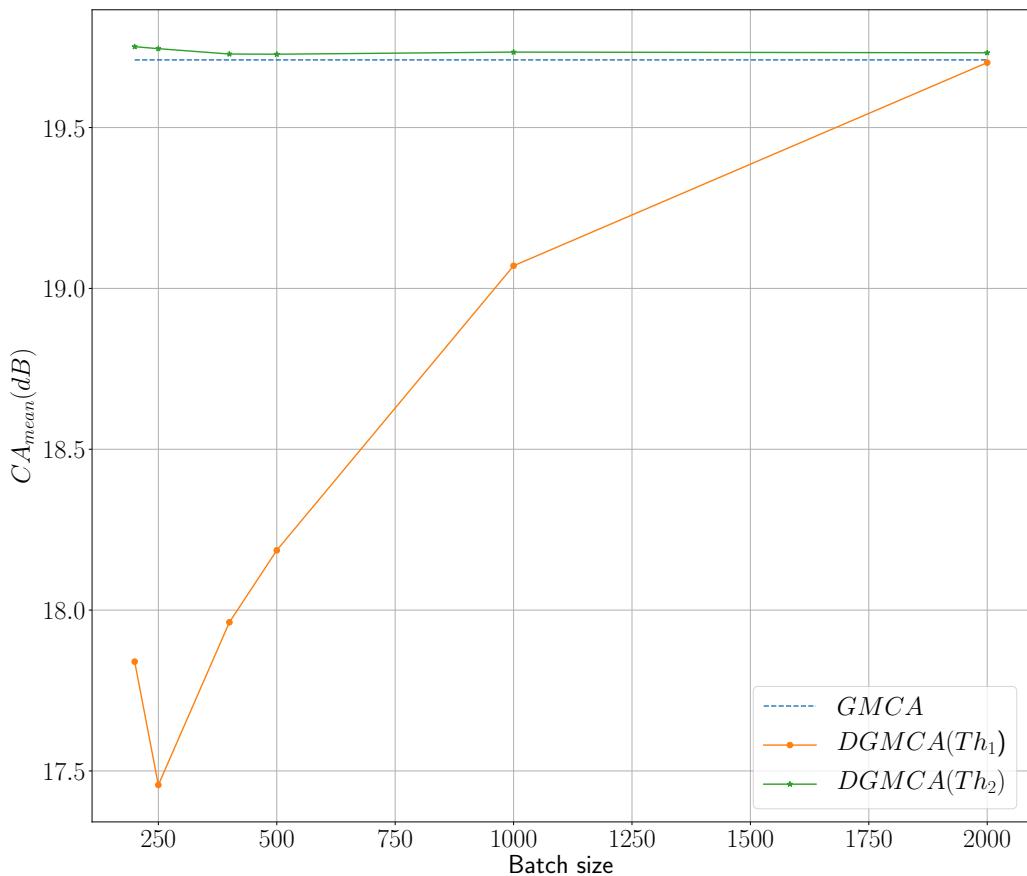


Figure A.1.1: Performance comparison of two thresholding strategies for different batch sizes.

A.1.2 TEST: LEVEL OF SPARSITY

This test will evaluate the performance of the DGMCA algorithm when changing the level of sparsity of the sources used. This level is controlled by the β parameter of the Generalized Gaussian model used to simulate the sources. The weighting strategy for the Fréchet mean is the one adopted for the DGMCA.

Settings:

- Number of observation channels : 20
- Number of sources: 5
- SNR: 60dB
- Values of β : [0.1, 0.15, 0.25, 0.35, 0.5, 0.7, 1]
- Number of samples : 2000
- Size of batch : [200]
- Number of repetitions : 11
- Number of epochs: 100
- Use of AMCA weights: Yes

The DGMCA suffers when the sparsity level is too high in comparison to the performance of the GMCA. Nevertheless, as it can be seen in figure it still shows a satisfactory performance over a vast range of sparsity levels.

A.1.3 TEST: NUMBER OF EPOCHS

The analysis of the performance of the algorithm in function of the number of epochs used is the objective of this test. The algorithm presents a fast convergence towards the stable level of performance.

Settings:

- Number of observation channels : 20
- Number of sources: 5

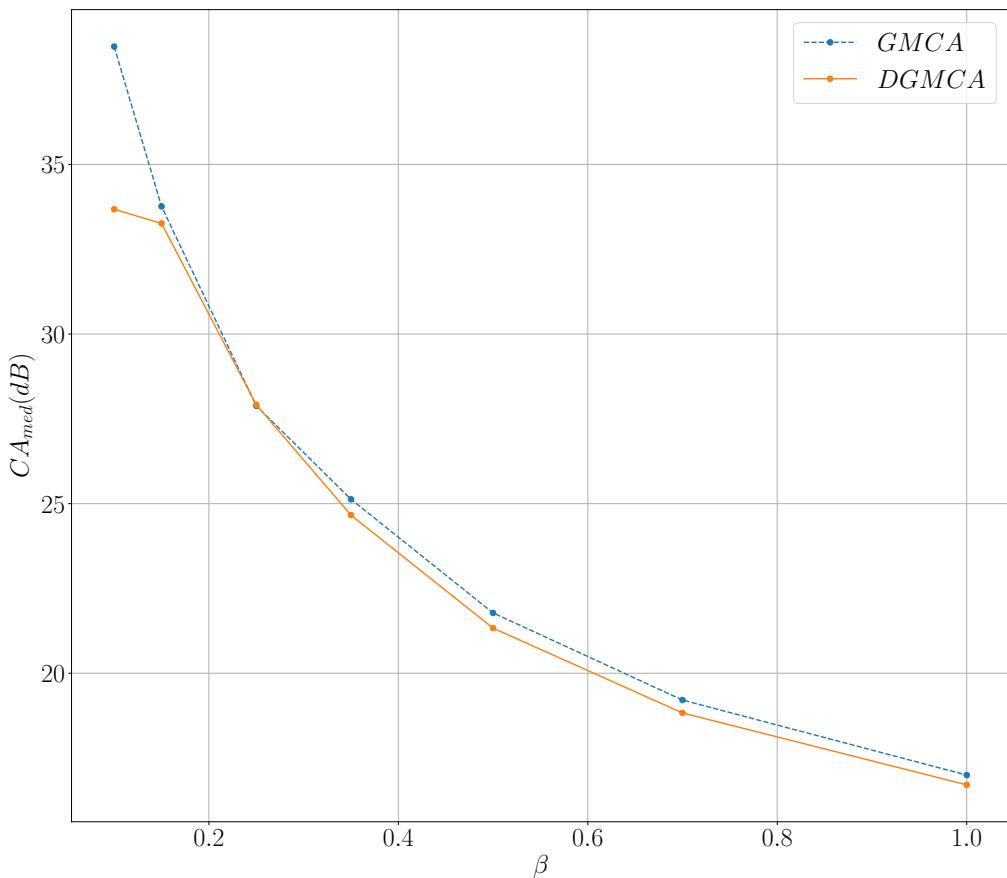


Figure A.1.2: Performance test for different levels of sparsity.

- SNR: 60dB
- Values of β : 0.5
- Number of samples : 2000
- Size of batch : [200]
- Number of repetitions : 25
- Number of epochs: [10, 25, 50, 100, 200, 400, 800]
- Use of AMCA weights: Yes

The algorithm seems to have converged to its stable performance at 25 epochs as seen in figure A.1.3. Nevertheless, a conservative approach is taken and the default number of iterations is set to

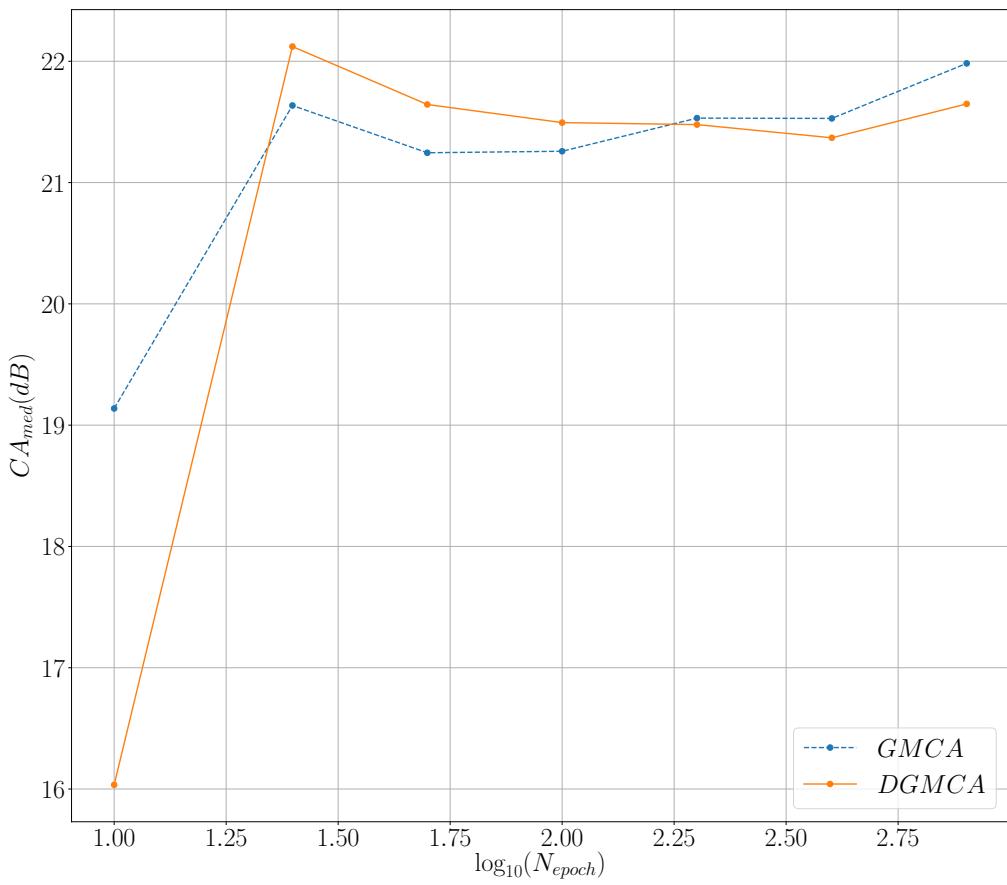


Figure A.1.3: Performance level in function of the number of epochs used on the algorithms.

100.

A.1.4 TEST: NUMBER OF SOURCES

The purpose of this test is to see the performance of the algorithm for different number of sources.

Settings:

- Number of observation channels : 50
- Number of sources: [2, 5, 10, 15]
- SNR: 60dB
- Values of β : 0.5
- Number of samples : 2000

- Size of batch : [200, 250, 400, 500, 1000, 2000]
- Number of repetitions : 11
- Number of epochs: 100
- Use of AMCA weights: Yes

We can see in figure A.1.4 that the performance of the DGMCA is damaged for a bigger number of sources when we are using small batch sizes. Given the number of sources and the lack of statistics in the smaller batches the separation performance drastically decreases.

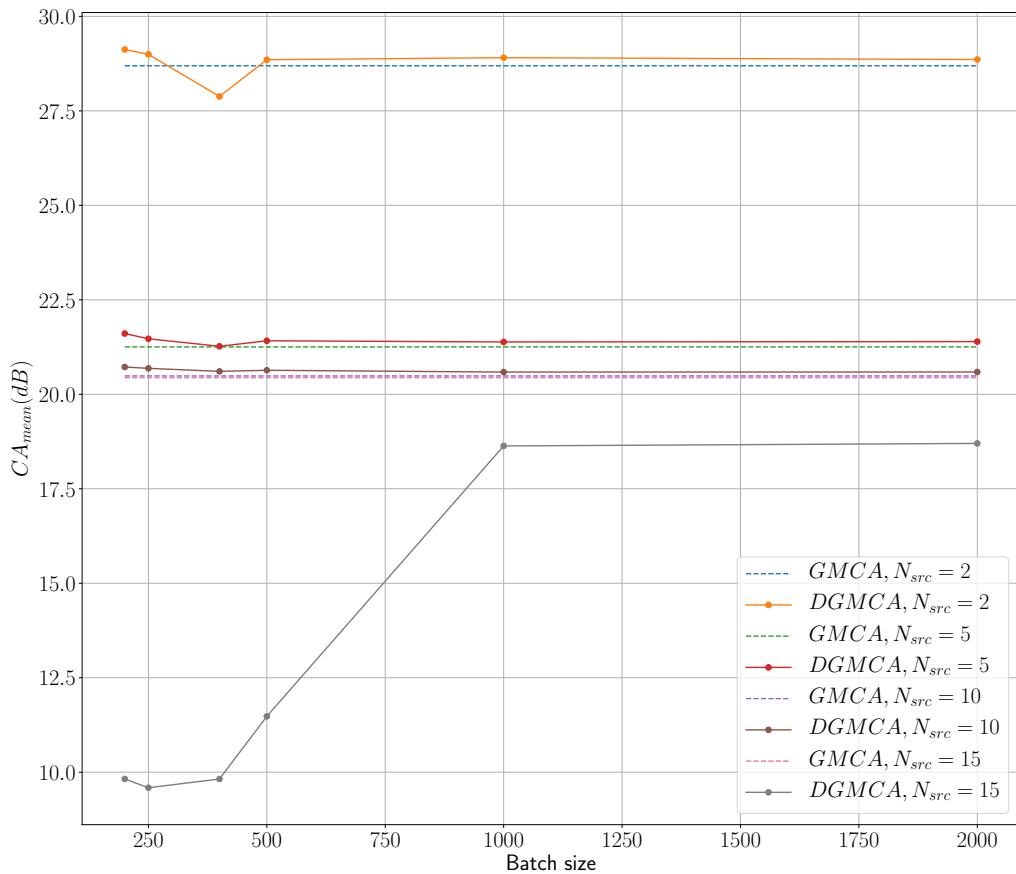


Figure A.1.4: Performance level for different number of sources and different sizes of batches.

*“Atención pido al silencio
y silencio a la atención,
que voy en esta ocasión,
si me ayuda la memoria,
a mostrarles que a mi historia
le faltaba lo mejor.”*

El gaucho Martín Fierro - José Hernández

B

SPARS Conference Article

In this annex we will present the article accepted to the SPARS¹ 2019 (*Signal Processing with Adaptive Sparse Structured Representations*) conference, which also included a poster presentation.

¹<http://www.spars-workshop.org/en/index.html>

Distributed sparse BSS for large-scale datasets

Tobias I. Liaudat, Jérôme Bobin, and Christophe Kervazo

CEA, IRFU, SEDI/Service d'Astrophysique

91191 Gif-sur-Yvette Cedex, France

Email: tobiasliaudat@gmail.com

Abstract—Blind Source Separation (BSS) [1] is widely used to analyze multichannel data stemming from origins as wide as astrophysics to medicine. Forthcoming projects like the SKA telescope will substantially increase the amount of data to process and existent BSS methods do not efficiently handle large datasets. In this work, we propose a new method coined DGMCA (Distributed Generalized Morphological Component Analysis) in which the original BSS problem is decomposed into subproblems that can be tackled in parallel, alleviating the large-scale issue. We propose to use the RCM (Riemannian Center of Mass – [6][7]), during the iterative process, to aggregate the estimations yielded by the different subproblems. The approach is made robust both by the chosen the weights of the RCM and the adaptation of the heuristic parameter choice proposed in [4] to the parallel framework. The results obtained show that the proposed approach is able to handle large-scale problems with a linear acceleration performing at the same level as GMCA and maintaining an automatic choice of parameters.

I. LARGE-SCALE BLIND SOURCE SEPARATION

Given m row observations of size t stacked in a matrix \mathbf{Y} assumed to follow a linear model $\mathbf{Y} = \mathbf{AS} + \mathbf{N}$, the objective of BSS [1] is to estimate the matrices \mathbf{A} (size $m \times n$) and \mathbf{S} (size $n \times t$) up to a mere permutation and scaling indeterminacy. In this model, \mathbf{A} mixes the n row sources in \mathbf{S} , the observations being disrupted by some unknown noise \mathbf{N} (size $m \times t$). We will assume that $n \leq m$. While ill-posed, this problem can be regularized assuming the sparsity of \mathbf{S} [2] resulting in a good separation quality [5]. The estimation will then turn into the minimization of:

$$\hat{\mathbf{A}}, \hat{\mathbf{S}} \in \arg \min_{\mathbf{A}, \mathbf{S}} \frac{1}{2} \|\mathbf{Y} - \mathbf{AS}\|_F^2 + \|\Lambda \odot \mathbf{S}\|_1 + i_{\mathcal{C}}(\cdot) \mathbf{x}_{k:} \|\mathbf{x}_k\|_2 = 1, \forall k(\mathbf{A}), \quad (1)$$

with $\|\cdot\|_F$ the Frobenius norm, Λ the regularization parameters and $i_{\mathcal{C}}(\cdot)$ the indicator function of the set \mathcal{C} . The first term is a data fidelity one, the second enforces the sparsity and the last avoids degenerated solutions with $\|\mathbf{A}\|_F^2 \rightarrow 0$ by enforcing unit norm columns. To tackle Eq. (1), the GMCA [4] algorithm has been a success, partly due to an automatic decreasing parameter strategy making it robust to local minima. However, we will assume that the data \mathbf{Y} are *large-scale* in the sense that t can attain huge values (e.g. up to 10^9 samples), which makes the treatment of \mathbf{Y} as a whole intractable. In this context, using GMCA is prohibitive.

II. PROPOSED METHOD

This difficulty motivates the construction of J subproblems (j) of the type $\mathbf{Y}_j = \mathbf{AS}_j + \mathbf{N}_j$ where j denotes a subset of t_j columns of the corresponding matrices. We use disjoint sets with $\sum_j |t_j| = t$. A natural idea is then the extension of GMCA to work in parallel on tractable smaller subproblems to minimize Eq. (1). While this approach is reminiscent to mini-batch approaches in machine learning [9], it raises two issues in the context of solving BSS through GMCA: i) each subproblem (j) yields a full estimate $\hat{\mathbf{A}}_{(j)}$ of \mathbf{A} . Is it possible to aggregate them to get a better final estimate?; ii) is it possible to extend the automatic parameter choice of GMCA (that made its success) to a parallel implementation?

A naive approach would be to independently solve each subproblem

(j) and aggregate the different final results. However, since GMCA is an iterative algorithm, aggregating the estimations $\hat{\mathbf{A}}_{(j)}$ of the different subproblems (j) *during the iterations* should reduce the error propagation, as the estimations are shared throughout the subproblems. More specifically, our DGMCA algorithm performs the aggregation through the weighted RCM [6] of the different columns $\hat{\mathbf{a}}_{i,(j)}^{(k+1)}$ of the estimations $\hat{\mathbf{A}}_{(j)}^{(k+1)}$ yielded by the different (j) subproblems at iteration $k+1$, which enables to take into account the geometry of the problem, the fact that each column belongs to the unit hypersphere. Its calculation is done following a gradient descent whose convergence is assured by [7]. Roughly speaking, the RCM can be understood as a weighted angular mean on the hypersphere. To robustify this process, we further propose to compute the weights based on an estimation of the Signal-to-Noise Ratio (SNR) of the corresponding estimated sources $\tilde{\mathbf{s}}_j^{i,(k+1)}$ to penalize noisy estimations (cf. Fig. 3).

Concerning question ii), the parameter choice of GMCA needs to access the whole distribution of the sources at each iteration, which is intractable in the large-scale regime. We propose a new strategy that can be used in a parallel implementation. It is based on a parametrized exponential decay which adapts to the signal statistics by using the maximum value of the estimated sources. The $\|\cdot\|_\infty$ can be taken of each mini-batch and then the maximum of these values will give the global $\|\cdot\|_\infty$. The expression is the following one:

$$\lambda_i^{(k+1)} = K\sigma_{\mathbf{S}^{i,(k)}} + (\|\mathbf{S}^{i,(k)}\|_\infty - K\sigma_{\mathbf{S}^{i,(k)}}) e^{-k\alpha_i}, \quad (2)$$

where $\mathbf{S}^{i,(k)}$ is the i row of the source matrix, $\sigma_{\mathbf{S}^{i,(k)}}$ is the noise standard deviation of the i source (it can be estimated using the Median Absolute Deviation) and K is a constant depending on the desired noise thresholding, generally set to 3. The threshold decay rapidity is regulated by the parameter α_i and can be adjusted in the first iterations by fitting a generalized Gaussian to the sources.

As hypothesis, we assume that the different sources are not correlated and that each source signal is stationary throughout the columns of \mathbf{S} . In practice, to partially overcome the first assumption, the penalization idea in [5] can be introduced in DGMCA. Concerning the second one, each subset j of columns can be randomly chosen in each iteration aiming to have identically distributed subproblems.

III. EMPIRICAL RESULTS AND CONCLUSIONS

Numerical experiments can be found in Fig. 1 and 2, while the algorithm overview in Fig. 3. In brief, our method paves the way for distributed approaches of BSS problems with automatic parameter tuning. It not only handles large-scale datasets but it enables a linear acceleration. Furthermore, it does not lower the separation quality compared to GMCA and outperforms methods like the optimized ODL [9].

ACKNOWLEDGEMENTS

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

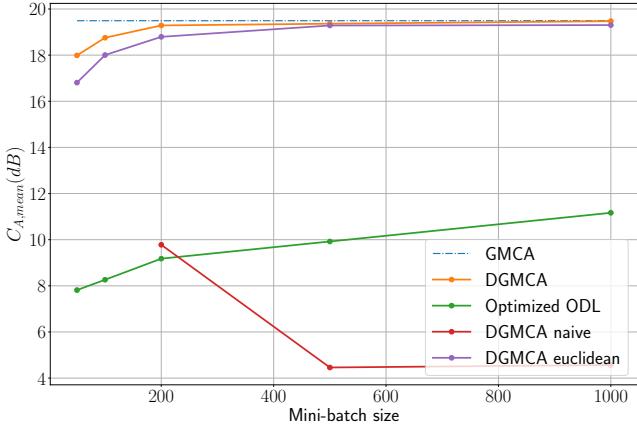


Fig. 1. Performance comparison of the GMCA, the ODL (Online Dictionary Learning [9]), and the DGMCA and its variants. Note that the parameters of the ODL algorithm have been optimized for this experiment by an exhaustive search. The x-axis corresponds to the size t_j of each subproblem (j), which is set for all j to be $t_j = t/J$. The y-axis represents the separation quality, measured by a mixing matrix criterion [5] defined as the average value of the ℓ_1 norm of the columns of $\mathbf{P}\hat{\mathbf{A}}^\dagger\mathbf{A}^* - \mathbf{I}$, where \mathbf{A}^* is the ground truth, \mathbf{P} accounts for the correction of the permutations and \mathbf{I} is the identity matrix. To generate the experiments, the source matrix was randomly sampled from a Generalized Gaussian distribution with several profile parameters β between 0.35 and 1.4, having $n = 10$ sources, $t = 10000$ samples and $m = 20$ observations. The noise matrix \mathbf{N} was set to have a SNR of 15dB. The matrix \mathbf{A} is random and with a condition number fixed to 10. The experiment is repeated 3 times and the mean of the results is being plotted.

Four parallelized algorithms are compared: the presented *DGMCA*, the *DGMCA naive* method consisting of solving the J subproblems independently until convergence and performing the aggregation at the end, the *DGMCA Euclidean* method where the RCM is substituted with an Euclidean mean for the aggregation, and the ODL with its hyperparameters optimized. The results of the different algorithms are benchmarked with the ODL and the GMCA using the entire observation matrix (which is only possible due to the relatively small t that we chose for the sake of the comparison).

The DGMCA outperforms the other parallelized methods maintaining a similar performance compared to the GMCA. It is worth noting that the performance is limited by the size of the mini-batch and not by the total size t which can be increased thus making the number of mini-batches increase. The separation quality is only reduced for extremely small mini-batches t_j , which was expected due to the lack of statistics for the algorithm to work. The *DGMCA naive* is not plotted for the two smallest t_j as some mini-batches only contained noise, which caused the algorithm not to converge for the threshold level used (as the subproblems are solved independently until convergence). In addition, the huge gap between DGMCA and its naive version confirms the usefulness of using an aggregation process during the iterations. Furthermore, using the RCM as aggregation and therefore taking into account the geometry of the problem enables better results than with the Euclidian mean. In the context of reproducible research, the code is available online at:

<https://github.com/tobias-liaudat/DGMCA>.

REFERENCES

- [1] P. Comon and C. Jutten, “Handbook of Blind Source Separation: Independent component analysis and applications,” Academic press, 2010.
- [2] M. Zibulevsky and B. A. Pearlmutter, “Blind source separation by sparse decomposition in a signal dictionary,” *Neural computation*, vol. 13, no. 4, pp. 863-882, 2001.
- [3] A. Cichocki and R. Zdunek, “Regularized alternating least squares algorithms for non-negative matrix/tensor factorization,” in *International Symposium on Neural Networks*. Springer, pp. 793-802, 2007.
- [4] J. Bobin, J.-L. Starck, J. Fadili, and Y. Moudden, “Sparsity and morphological diversity in blind source separation,” *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2662-2674, 2007.
- [5] J. Bobin, J. Rapin, A. Larue, and J.-L. Starck, “Sparsity and adaptivity for the blind separation of partially correlated sources,” *IEEE Transactions on Signal Processing*, vol. 63, no. 5, pp. 1199-1213, 2015.

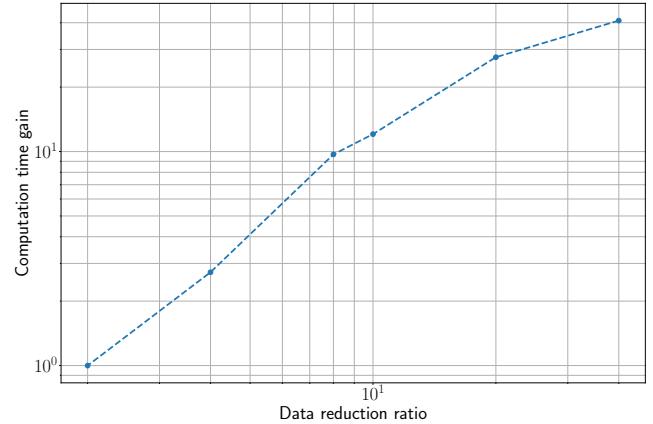


Fig. 2. Computational time gain between the parallelized DGMCA and the GMCA algorithms against the data reduction ratio, which is calculated as the problem total size, t , divided by the size of the mini-batch, $t/t_j = J$. Each point on the figure represents the mean over 10 problems. The experiment was run using a C++ parallelized version of the DGMCA algorithm and the maximum number of mini-batches used is 40 as it is the number of cores the computer cluster used had. The setup of the experience is similar to the one in Fig 1, with a β parameter of 0.5, having $n = 5$ sources, $t = 10000$ samples and a *SNR* of 40dB. The linear trend of the time gain was predicted by the complexity analysis of the algorithms, and now confirmed by the numerical experiment. In brief, the time spent by the DGMCA is approximately the one spent in the GMCA algorithm divided by the number of subproblems used.

```

1: procedure DGMCA( $\mathbf{Y}$ , parameters)
2:   while do not converge do
3:     Calculate  $\lambda_i^{(k)}$ 
4:     for  $j = 1, \dots, J$  do
5:        $\tilde{\mathbf{S}}_j^{(k+1)} \leftarrow (\hat{\mathbf{A}}_{RCM}^{(k)})^\dagger \mathbf{Y}_j$            (LS estimation)
6:        $\hat{\mathbf{S}}_j^{(k+1)} \leftarrow \mathcal{S}_{\Lambda^{(k)}}(\tilde{\mathbf{S}}_j^{(k+1)})$       (Prox. op. of  $\|\Lambda^{(k)} \odot \cdot\|_1$ )
7:        $\hat{\mathbf{A}}_{(j)}^{(k+1)} \leftarrow \mathbf{Y}_j (\hat{\mathbf{S}}_j^{(k+1)})^\dagger$            (LS estimation)
8:        $\hat{\mathbf{a}}_{i, (j)}^{(k+1)} \leftarrow \frac{\hat{\mathbf{a}}_{i, (j)}^{(k+1)}}{\|\hat{\mathbf{a}}_{i, (j)}^{(k+1)}\|_2}, \forall i \in \{1, \dots, n\}$  (Prox. op. of  $i_C(\cdot)$ )
9:        $[\mathbf{W}_{RCM}^{(k+1)}]_{i,j} = \|\hat{\mathbf{s}}_j^{(k+1)}\|_2^2 / \sigma_{Y_j} \|\hat{\mathbf{A}}^{(k)}\|_F^2, \forall i, j$ .
10:      Correct permutations in  $\mathbf{A}_{(j)}^{(k+1)}, \forall j$ .
11:       $\hat{\mathbf{A}}_{RCM}^{(k+1)} \leftarrow RCM(\hat{\mathbf{A}}_{(1)}^{(k+1)}, \dots, \hat{\mathbf{A}}_{(J)}^{(k+1)}, \mathbf{W}_{RCM}^{(k+1)})$  (Aggregation)
12:       $k \leftarrow k + 1$ 
13:   return  $\hat{\mathbf{A}}_{RCM}^{(k)}, \hat{\mathbf{S}}^{(k)}$ 

```

Fig. 3. DGMCA. the operator $(\cdot)^\dagger$ is the pseudo-inverse, $\mathcal{S}_\lambda(\cdot)$ is the soft-thresholding operator with the threshold λ , \mathbf{a}_i denotes the i th column of \mathbf{A} and the subscript (j) denotes the estimation of the j subproblem.

- [6] B. Afsari, “Riemannian Lp center of mass: Existence, uniqueness, and convexity” *Proc. Amer. Math. Soc.*, vol. 139, no. 2, pp. 655-673, 2011.
- [7] B. Afsari, R. Tron and R. Vidal, “On The Convergence of Gradient Descent for Finding the Riemannian Center of Mass,” arXiv:1201.0925 (Dec 2011).
- [8] J. Munkres, “Algorithms for the Assignment and Transportation Problems,” *Journal of the Society of Industrial and Applied Mathematics*, 5(1): pp. 32-38, 1957.
- [9] J. Mairal, F. Bach, J. Ponce and G. Sapiro, “Online Learning for Matrix Factorization and Sparse Coding,” *Journal of Machine Learning Research*, 11(1): pp. 19-60, 2010.

Distributed sparse BSS for large-scale datasets

DE LA RECHERCHE À L'INDUSTRIE



Tobias Liaudat, Jérôme Bobin, Christophe Kervazo

CEA Saclay, Irfu/SEDI, France | christophe.kervazo@cea.fr

Abstract

Blind Source Separation (BSS) is widely used to analyze multichannel data stemming from origins as wide as astrophysics to medicine. However, existent methods do not efficiently handle large datasets. In this work, we propose a new method coined dGMCA (distributed Generalized Morphological Component Analysis) in which the original BSS problem is decomposed into subproblems which can be tackled in parallel, alleviating the large-scale issue. We use the RCM (Riemannian Center of Mass) to aggregate during the iterative process the estimations yielded by the different subproblems. We further robustify our large-scale sparse BSS method through the introduction of distributed automatic hyper-parameter choices.

Context

Sparse BSS

- Some multichannel data \mathbf{X} composed of m row observations are assumed to be the linear combination, entangled of noise \mathbf{N} , of n unknown elementary sources \mathbf{S} of t samples (Comon 2010):

$$\mathbf{X} = \mathbf{AS} + \mathbf{N} \quad (1)$$

- Goal: estimate *physical* \mathbf{A} and \mathbf{S} from the sole \mathbf{X} . Ill-posed unsupervised matrix separation problem mandating additional information: here, focus on sparse sources (Bobin 2007).

Large-scale BSS and data deluge

- Ever growing datasets in many fields. In astronomy: SKA, Euclid, LSST → up to $t = 10^9$ samples!
- Two challenges: i) Reduce computation time; ii) Alleviate the memory burden.

We aim at performing sparse BSS on *large-scale* datasets \mathbf{X} in a scalable and reliable way.

Problem formulation and state-of-art

Sparse BSS as an optimization problem

- A way to perform sparse BSS is to look for a minimizer of (Zibulevski 2001):

$$\underset{\mathbf{A}, \mathbf{S}}{\text{minimize}} \frac{1}{2} \|\mathbf{X} - \mathbf{AS}\|_F^2 + \left\| \mathbf{R}_S \odot \mathbf{S} \Phi^T \right\|_1 + \sum_{j \in [1, n]} i_{\|\mathbf{A}^j\|_2=1}(\mathbf{A}) \quad (2)$$

- 3 terms: 1) data fidelity stemming from Gaussian noise assumption; 2) a ℓ_1 sparsity constraint on \mathbf{S} ; \mathbf{R}_S controls the trade-off with 2), Φ is a sparsifying transform; 3) oblique constraint on \mathbf{A} , used to avoid degenerated solutions.

Related works

No fully satisfying works for large-scale sparse BSS:

- Small-scale sparse BSS algorithms*: (Bobin 2007) based on projected Alternating Least Square (pALS), enabling an automatic choice of \mathbf{R}_S . Works well on small-scale data only.
 - General large-scale sparse matrix factorization algorithms*: (Mairal 2010, Davis 2016) Highly scalable as based on mini-batches. But do not use pALS, implying low-reliability for sparse BSS (Kervazo 2018).
- Proposed method: merge the best of the above two-worlds by **introducing mini-batches in the pALS scheme** of (Bobin 2007).

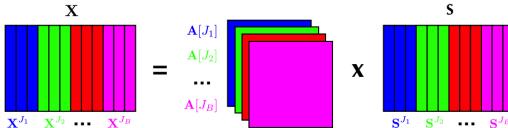
Distributing the GMCA algorithm: overview

Introducing mini-batch pALS: difficulty

- Main idea: re-use the GMCA scheme of (Bobin 2007) but split initial dataset \mathbf{X} into B mini-batches \mathbf{X}^{J_b} , $b \in [1, B]$. The indices of the columns are denoted as J_b (here, $\#J_b = t/b$).

But:

- Issue i) Several estimations $\mathbf{A}[J_b] \in \mathbb{R}^{m \times n}$ of the same $\mathbf{A} \in \mathbb{R}^{m \times n}$



→ Use an aggregation method to get a final estimate $\hat{\mathbf{A}}$

- Issue ii) The automatic regularization parameter \mathbf{R}_S choice which made the strength of GMCA uses the *whole* source distribution.

→ We introduce a highly parallelizable adaptative parameter \mathbf{R}_S choice.

dGMCA: algorithm structure

```

1: procedure dGMCA(X,A(0))
2:   for k = 1, ..., K do
3:     Choose  $J_1, J_2, \dots, J_B$  as a partition of [1, t]
4:     for b = 1, ..., B do
5:        $\hat{\mathbf{S}}^{J_b(k)} = \hat{\mathbf{A}}^{(k-1)\dagger} \mathbf{X}^{J_b(k)}$ 
6:        $\mathbf{R}_S^{(k)} = \text{DISTRIBUTED\_R}_S\text{\_CHOICE}$ 
7:        $w[J_b(k)] = \text{COMPUTE\_WEIGHT}(\hat{\mathbf{S}}^{J_b(k)})$ 
8:        $\hat{\mathbf{S}}^{J_b(k)} = \mathcal{S}_{\mathbf{R}_S^{(k)}}(\hat{\mathbf{S}}^{J_b(k)})$ 
9:        $\mathbf{A}[J_b(k)] = \Pi_{\|\cdot\|_2=1}(\mathbf{X}^{J_b(k)} \hat{\mathbf{S}}^{J_b(k)\dagger})$ 
10:    end for
11:     $\hat{\mathbf{A}}^{(k)} = \text{AGGREGATE}(\mathbf{A}[J_1(k)], \mathbf{A}[J_2(k)], \dots, \mathbf{A}[J_B(k)], w[J_1(k)], \dots, w[J_B(k)])$ 
12:  end for
13:  return  $\hat{\mathbf{A}}^{(K)}, \hat{\mathbf{S}}^{(K)}$ 
14: end procedure

```

Detailing the main steps

Aggregation step

- Simplistic solution to aggregate the different $\mathbf{A}[J_b], b \in [1, B]$: Euclidean mean.
- $\hat{\mathbf{A}}$ must respect the oblique constraint → **aggregate using a Riemannian Center of Mass – RCM** (Afsari 2011).
- Compute using (Afsari 2011) the RCM by finding the point on the manifold minimizing a sum of the weighted square geodesic lengths d :

$$\forall j \in [1, n], \quad \hat{\mathbf{A}}^j = \underset{\mathbf{a}}{\operatorname{argmin}} \sum_{b \in [1, B]} w[J_b]^j d^2(\mathbf{a}, \mathbf{A}[J_b]^j). \quad (3)$$

- Robustification: weights $w[J_b]^j$ are used to penalize noisy sources estimated from a given mini-batch b :

$$\forall j \in [1, n], \quad \forall b \in [1, B]; \quad w[J_b]^j = \frac{([A[J_b]^+]_j \Sigma_N [A[J_b]^+]_j^T)^{-1}}{\sum_{b=1}^n ([A[J_b]^+]_j \Sigma_N [A[J_b]^+]_j^T)^{-1}} \quad (4)$$

Distributing the \mathbf{R}_S parameter choice

- In GMCA, \mathbf{R}_S is chosen as an increasing percentile of the estimated sources → requires the whole source distribution.

- Here, we use parametrized exponentially decaying \mathbf{R}_S , which is *easily parallelizable*:

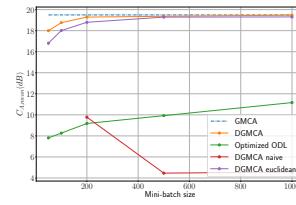
$$R_{is}^i = K \sigma_i + \left(\left\| \hat{S}_i^{(k)} \right\|_\infty - K \sigma_i \right) \exp(-k \alpha_i), \quad (5)$$

where σ_i is an estimation of the back-projected noise on source S_i std (estimated through the Median Absolute Deviation operator), $K = 3$ is a parameter (set according to a fixed point argument, see (Bobin 2007)), $\left\| \hat{S}_i^{(k)} \right\|_\infty$ is the maximum absolute value of $\hat{S}_i^{(k)}$ and α_i is a parameter controlling the exponential decay speed.

- Parameter α_i can be chosen fitting a generalized Gaussian to the sources during the first iterations. However, in practice the results are quite insensitive to α_i values, enabling to set them beforehand.

Numerical experiments: simulations

- Sources follow a Generalized Gaussian distribution with $\beta \in [0.35, 1.4]$. The parameters are $n = 10$, $m = 20$, $t = 10000$, $SNR = 15dB$. The matrix \mathbf{A} is random with a condition number of 10.



Left: separation quality in terms of $C_A = -10 \log \left(\left\| P \hat{\mathbf{A}} - \mathbf{I} \right\|_F \right)$, with P correcting permutations and scale; Right: Computational time gain between GMCA and dGMCA.

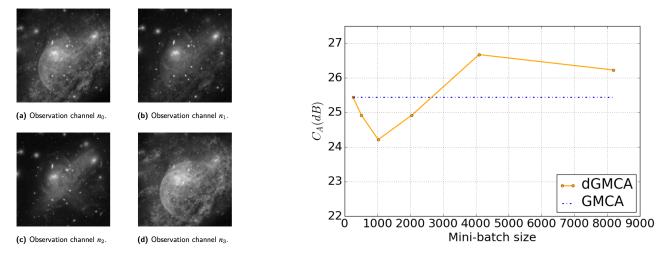
- dGMCA enables to maintain a high separation quality while providing an almost linear gain in computation time.

Numerical experiments: realistic sources

- $n = 5$ realistic astrophysical sources, mixed through a \mathbf{A} matrix resembling power laws. There are $m = 250$ observations, $t = 32768$.

- The sources are sparse in the starlet domain (Starck 2010)

- The transform is applied on each mini-batch. With relatively large mini-batches, the support of the transform is small enough to limit border effects.



Conclusion and perspectives

We introduced in this work the dGMCA algorithm, which enables to perform large-scale sparse BSS through a parallelization of the GMCA algorithm. The method is based on a mini-batch pALS optimization scheme. The originality lies in the use of the Riemannian Center of Mass to aggregate the different estimates during the estimation process, enabling to take into account the geometry of the problem. The approach is further robustified both by a clever choice of the weights of the RCM based on the estimated SNR and a parallelized heuristic regularization parameter choice. Numerical experiments on both simulated and realistic sources show that the proposed approach is able to handle large-scale problems, as it enables a linear acceleration and alleviate the memory burden, while getting results almost as good as GMCA.

Bibliography

- [1] J. Bobin, J.-L. Starck, J. M. Fadili, Y. Moudden. Sparsity and Morphological Diversity in Blind Source Separation. *IEEE Transactions on Image Processing*, Institute of Electrical and Electronics Engineers, 2007, 16 (11), pp.2662-2674.
- [2] J. Bobin, J.-L. Starck, J. M. Fadili, Y. Moudden. Blind Source Separation: The Sparsity Revolution, *Advances in Imaging and Electron Physics* , Vol 152, pp 221 – 306, 2008.
- [3] J. Bobin, J.-L. Starck, J. Fadili, Y. Moudden, and D. Donoho, Morphological Component Analysis: An adaptive thresholding strategy, *IEEE Trans. On Image Processing*, 16 (2007), pp. 2675 – 2681.
- [4] J. Bobin, J. Rapin, J.-L. Starck and A. Larue, Sparsity and adaptivity for the blind separation of partially correlated sources, *IEEE TSP*, 63(5), 2015.
- [5] J. Bobin, F. Sureau, J.-L. Starck, A. Rassat, & P. Paykari, 2014, *A&A*, 563, A105.
- [6] C. Kervazo, J. Bobin, C. Chenot. Heuristics for Efficient Sparse Blind Source Separation. EU-SIPCO 2018, Sep 2018, Rome, Italy. <hal-01853246>
- [7] C. Kervazo, J. Bobin, C. Chenot. Blind separation of a large number of sparse sources. *Signal Processing*, Volume 150, 2018, Pages 157-165, ISSN 0165-1684, <https://doi.org/10.1016/j.sigpro.2018.04.006>.
- [8] M. Jiang, J. Bobin, J.L. Starck. Joint Multichannel Deconvolution and Blind Source Separation. *SIAM J. Imaging Sci.*, 10(4), 1997-2021. (25 pages) (2017)
- [9] C. Chenot. Parcimonie, diversité morphologique et séparation robuste de sources. *Traitemen*t du signal et de l'image. Université Paris-Saclay, 2017. Français.
- [10] J. Rapin. Décompositions parcimonieuses pour l'analyse avancée de données en spectrométrie pour la Santé. *Mathématiques générales [math.GM]*. Université Paris Sud - Paris XI, 2014. Français. <NNT : 2014PA112378>. <tel-01128517>

- [11] Y. Xu, W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion", SIAM J. Imag. Sci., vol. 6, no. 3, pp. 1758-1789, 2013.
- [12] P. Paatero, U. Taaper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. Environmetrics, 5 (1994), pp. 111-126.
- [13] Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverseproblems. SIAM Journal on Imaging Sciences 2(1), 183–202 (2009).
- [14] J. Bolte, S. Sabach, M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Mathematical Programming, 146 (2014), pp.459-494.
- [15] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. Journal of optimization theory and applications, 109 (2001), pp. 475-494.
- [16] B. Asfari, R. Tron, R. Vidal. On The Convergence of Gradient Descent for Finding the Riemannian Center of Mass. arXiv:1201.0925 (Dec 2011)
- [17] B. Afsari. Riemannian L_p center of mass: Existence, uniqueness, and convexity. Proc. Amer. Math. Soc., 139:655–673, 2011.
- [18] P.-A. Absil, R. Mahony, and R. Sepulchre. Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton, NJ, 2008.
- [19] J.L. Starck, F. Murtagh, and J. Fadili, Sparse Image and Signal Processing: Wavelets and Related Geometric Multiscale Analysis, Cambridge University Press, Cambridge (GB), 2016.
- [20] Varanasi, M. K. and Aazhang B. "Parametric generalized Gaussian density estimation" The Journal of the Acoustical Society of America. 86(4) (1989) 1404-1415.
- [21] M. N. Do and M. Vetterli, "Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance," in IEEE Transactions on Image Processing, vol. 11, no. 2, pp. 146-158, Feb. 2002.
- [22] H. W. Kuhn. The Hungarian Method for the assignment problem. Naval Research Logistics Quarterly, 2:83-97, 1955.
- [23] J. Munkres. Algorithms for the Assignment and Transportation Problems. Journal of the Society of Industrial and Applied Mathematics, 5(1):32-38, March, 1957.
- [24] E.J. Candès, M.B. Wakin, S.P. Boyd. Enhancing Sparsity by Reweighted ℓ_1 Minimization. arXiv:0711.1612. (2007)

- [25] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, et al. Hyperspectral Unmixing Overview: Geometrical, Statistical, and Sparse Regression Based Approaches. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, IEEE, 2012, 5 (2), pp.354-379. <10.1109/JSTARS.2012.2194696>. <hal-00760787>
- [26] Y. Moudden and J. Bobin, Hyperspectral BSS using GMCA with spatio-spectral sparsity constraints, *IEEE Transactions on Image processing*, 20 (2011), pp. 872–879.
- [27] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent component analysis and applications*, Academic press, 2010.
- [28] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari. 2009. Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation. Wiley Publishing.
- [29] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge university press, 2004.
- [30] N. Parikh, S. P. Boyd, et al., *Proximal algorithms.*, Foundations and Trends in optimization, 1 (2014).
- [31] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein (2011), "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers", *Foundations and Trends in Machine Learning*: Vol. 3: No. 1, pp 1-122.
- [32] J. Mairal, F. Bach, J. Ponce, G. Sapiro. Online Learning for Matrix Factorization and Sparse Coding. *Journal of Machine Learning Research*, Journal of Machine Learning Research, 2010, 11 (1), pp.19–60.
- [33] J. N. Girard. Développement de la Super Station LOFAR & observations planétaires avec LOFAR. Instrumentation et méthodes pour l'astrophysique [astro-ph.IM]. Observatoire de Paris, 2013. Français.
- [34] A.R. Thompson, J.M. Moran et J. George W. Swenson. *Interferometry and synthesis in radio astronomy*. John Wiley & Sons, 2008.
- [35] G. B. Taylor, C. L. Carilli, and R. A. Perley. Synthesis imaging in radio astronomy ii. In *Synthesis Imaging in Radio Astronomy II*, volume 180, 1999.
- [36] F. M. Ngolè Mboula. Advanced methods and algorithm for high precision astronomical imaging. *Image Processing*. Université Paris-Saclay, 2016. English

- [37] D. Davis. The asynchronous palm algorithm for nonsmooth nonconvex problems. arXiv preprint arXiv:1604.00526, 2016