ANNÉE 2014

UNIVERSITÉ DE RENNES 1

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Européenne de Bretagne

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Traitement du Signal et Télécommunications

Ecole doctorale MATISSE

présentée par

Lu WANG

préparée à l'unité de recherche LTSI - INSERM UMR 1099
Laboratoire de Traitement du Signal et de l'Image
UFR Informatique et Électronique

Nonnegative joint diagonalization by congruence for semi-nonnegative independent component analysis

Thèse soutenue à Rennes le 10 novembre 2014

devant le jury composé de :

Eric Moreau
Professeur des universités
Université de Toulon / rapporteur

Christian Jutten
Professeur des universités
Université Joseph Fourier / rapporteur

Jean-Christophe Pesquet
Professeur des universités
Université Paris-Est Marne-la-Vallée / examinateur

Huazhong Shu
Professeur des universités
Université du Sud-Est à Nankin / examinateur

Laurent Albera
Maître de conférences, HDR
Université de Rennes 1 / directeur de thèse

Lotfi Senhadji
Professeur des universités
Université de Rennes 1 / co-directeur de thèse

Diagonalisation conjointe non-négative par congruence pour l'analyse en composantes indépendantes semi-nonnégative

La Diagonalisation Conjointe par Congruence (DCC) d'un ensemble de matrices apparaît dans nombres de problèmes de Séparation Aveugle de Source (SAS), tels qu'en Analyse en Composantes Indépendantes (ACI). Les développements récents en ACI sous contrainte de nonnégativité de la matrice de mélange, nommée ACI semi-nonnégative, permettent de tirer profit d'une modélisation physique réaliste des phénomènes observés tels qu'en audio, en traitement d'image ou en ingénierie biomédicale. Par conséquent, durant cette thèse, l'objectif principal était non seulement de concevoir et de développer des algorithmes d'ACI semi-nonnégative basés sur de nouvelles méthodes de DCC nonnégative où la matrice de passage recherchée est nonnégative, mais également d'illustrer leur intérêt dans le cadre d'applications pratiques de séparation de sources.

Chapitre 1: Introduction

La SAS consiste à estimer P sources inconnues à partir de N observations. Ces observations sont des mélanges linéaires instantanés bruités des dites sources satisfaisant le modèle suivant [Comon and Jutten, 2010]:

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{\varepsilon} \tag{1}$$

où $\boldsymbol{x} = [x_1, x_2, \ldots, x_N]^\mathsf{T} \in \mathbb{R}^N$ est le vecteur d'observations, où $\boldsymbol{s} = [s_1, s_2, \ldots, s_P]^\mathsf{T} \in \mathbb{R}^P$ est le vecteur de sources, où $\boldsymbol{A} \in \mathbb{R}^{N\times P}$ est la matrice de mélange et où $\boldsymbol{\varepsilon} = [\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N]^\mathsf{T} \in \mathbb{R}^N$ est un vecteur de bruit additif. Le terme "aveuglement" fait référence au fait que les sources et la matrice de mélange sont inconnus. La SAS montre son intérêt dans un grand nombre d'applications, incluant les télécommunications, l'imagerie biomédicale et l'acoustique. Le problème de SAS est mal posé sans hypothèses supplémentaires. Afin d'obtenir des résultats pertinents dans une application réelle, il est nécessaire d'utiliser certains a priori sur les sources ou le modèle de mélange. Exploiter l'indépendance statistique des sources à parti d'un nombre fini de réalisations du vecteur aléatoire $\boldsymbol{x}$ (1) caractérise l'ACI [Jutten and Hérault, 1991,Comon, 1994]. L'hypothèse de

nonnégativité des sources ou de la matrice de mélange, voire les deux, a donné naissance à un large éventail d'approches de SAS, telles que la Factorisation Matricielle Nonnégative (FMN) [Lee and Seung, 1999] et l'ACI nonnégative [Plumbley, 2003]. Dans ce chapitre, nous avons effectué un bref survol bibliographique des méthodes de SAS les plus connues, à savoir l'ACI, l'ACI nonnégative et la FMN. En outre, la méthode de DCC et son lien avec l'ACI sont été décrits. Ensuite, nous avons introduit le modèle proposé d'ACI semi-nonnégative [Wang et al., 2013,Coloigner et al., 2014a,Coloigner et al., 2014b]. Plus précisément, dans le modèle d'ACI semi-nonnégative, la matrice de mélange (contrairement aux sources) est supposée avoir des composantes nonnégatives. En fait, selon ce modèle, les tranches matricielles des tableaux cumulant suivent le modèle de DCC:

$$\boldsymbol{C}^{(k)} = \boldsymbol{A}\boldsymbol{D}^{(k)}\boldsymbol{A}^{\mathsf{T}}, \quad k \in \{1, \ldots, K\} \tag{2}$$

où $\boldsymbol{A} \in \mathbb{R}_{+}^{N\times P}$ est la matrice de passage nonnégative et égale à la matrice de mélange, où $\boldsymbol{D}^{(k)} \in \mathbb{R}^{P\times P}$ sont des matrices diagonales, et où K est le nombre total de tranches matricielles du tableau cumulant. Par conséquent, on peut estimer la matrice $\boldsymbol{A}$ à l'aide de méthode de DCC sous contrainte de nonnégativité sur la matrice de passage, conduisant à la notion de DCC nonnégative. Durant cette thèse, nous avons proposé de nouveaux algorithmes de DCC nonnégative pour résoudre le problème d'ACI semi-nonnégative.

Chapitre 2: Algorithmes de diagonalisation conjointe nonnégative par congruence

Nous avons développé six nouvelles méthodes de DCC nonnégative, qui ont été soigneusement présentées dans ce chapitre. Les six algorithmes proposés peuvent être répartis dans deux groupes selon la stratégie d'optimisation. Le premier groupe d'algorithmes comprend cinq méthodes semi-algébriques, reposant sur une méthode de type Jacobi. Dans ce groupe, la contrainte de nonnégativité a été assurée par un changement de variable carré, permettant ainsi de nous ramener à un problème d'optimisation sans contrainte. L'idée générale de notre approche de type Jacobi est i) de factoriser la racine carrée de la matrice de passage recherchée sous forme d'un produit de matrices élémentaires, chacun n'étant défini que par un seul paramètre, puis ii) d'estimer ces matrices élémentaires l'une après l'autre dans un ordre spécifique. La première méthode utilise la factorisation LU afin de minimiser un critère des moindres carrés directs. Cependant, elle présente une grande complexité numérique. Afin de réduire la complexité numérique, nous avons reformulé le critère des moindres carrés indirects et nous avons considéré les factorisations LU et QR. Une telle reformulation a nécessitée l'utilisation d'une étape de prétraitement sur les matrices à diagonaliser, conduisant à deux algorithmes de calcul efficace. Néanmoins, la validité de ces deux algorithmes est fondée sur certaines hypothèses d'inverisibilité. Par conséquent, les quatrième et cinquième algorithmes proposés ont permis de relâcher ces hypothèses. Ces deux méthodes reposent sur une optimisation des deux critères des moindres carrés indirects basée sur la factorisation LU, sans aucune étape de prétaitement. Le deuxième groupe compte un seul algorithme, qui utilise la méthode des directions alternées. Cet algorithme exploite le fait que le problème de DCC nonnégative peut être reformulé comme un problème des moindres carrés directs sous contraintes de symétrie et de nonnégativité. Un tel algorithme a été dérivé en minimisant séquentiellement le Lagrangien augmenté par rapport aux variables et aux multiplicateurs. En

outre, la complexité numérique pour chaque algorithme a été analysée, ce qui permet une comparaison équitable de leurs performances.

Chapitre 3: Simulations numériques

Les performances des algorithmes proposés de DCC nonnégative ont été évaluées sur des jeux de matrices synthétiques, conjointement diagonalisables. Cinq méthodes de DCC non-orthogonale classiques sans contrainte de nonnégativité, incluant ACDC [Yeredor, 2002], CLU [Maurandi et al., 2013], FFDIAG [Ziehe et al., 2004], LUJ1D [Afsari, 2006], LUJ2D [Afsari, 2006] et une méthode de DCC nonnégative nommée $\text{LM}^{+}_{\text{sym}}$ [Coloigner et al., 2014c], ont été testées comme méthodes de référence. Les performances ont été évaluées en termes de précision de la matrice estimée et de complexité numérique. Tout au long du chapitre, i) les propriétés de convergence, ii) l'influence des erreurs de modèle, iii) l'incidence du nombre de matrices à diagonaliser, iv) l'effet de la cohérence des vecteurs colonnes de la matrice de passage, et v) l'influence du module d'unicité, ont été largement étudiés au moyen de réalisations de Monte Carlo. Les résultats ont montré qu'une meilleure précision d'estimation pouvait être obtenue en particulier dans des contexte difficiles, en exploitant l'information préalable de nonnégativité. Généralement nos algorithmes de type Jacobi fournissent de meilleurs résultats, pour un problème difficile de DCC nonnégative, par exemple pour de faibles valeurs de rapport signal-sur-bruit, pour un petit nombre de matrices à diagonaliser, pour des niveaux élevés de cohérence de la matrice de passage, et pour une grande valeur du module d'unicité. Parmi eux, l'algorithme basé sur le critère des moindres carrés indirects sans étape de prétraitement offre le meilleur compromis entre précision d'estimation et complexité numérique. Lorsque le problème de DCC nonnégative considéré est bien conditionné, c'est-à-dire lorsqu'une valeur élevée du rapport signal-sur-bruit est considérée ou un grand nombre de matrices à diagonaliser est disponible, l'algorithme basé sur l'optimisation des directions alternées est le plus efficace sur le plan de la précision d'estimation pour une complexité numérique équivalente. En la comparant à une méthode existante de DCC nonnégative basée sur l'optimisation de Levenberg Marquardt [Coloigner et al., 2014c], dont la performance est tout aussi compétitive pour certaines expériences, les méthodes proposées requièrent moins de temps de calcul (environ 3 fois moins). De manière générale, les algorithmes proposés offrent le meilleur compromis entre la performance et la complexité.

Chapitre 4: Applications de séparation aveugle de source

Dans ce chapitre, nous avons appliqué nos algorithmes de DCC nonnégative à un ensemble de tranches matricielles d'un tableau cumulant d'ordre quatre, donnant naissance à une nouvelle classe de méthodes d'ACI semi-nonnégative. L'ACI semi-nonnégative se retrouve dans un certain nombre de problèmes de SAS impliquant une matrice de mélange nonnégative, par exemple dans i) l'analyse de composés chimiques en spectroscopie par résonance magnétique, ii) l'identification des profils spectraux d'harmoniques (par exemple, de notes de piano) d'un morceau de musique mono-canal par décomposition du spectrogramme, iii) l'élimination partielle du texte se trouvant au verso d'une feuille de papier fin. Les méthodes proposées ont été appliquées pour résoudre ces problèmes afin

de vérifier leur capacité de séparation de sources. Nos méthodes ont été comparées à cinq algorithmes de référence en SAS, à savoir deux méthodes d'ACI: CoM2 [Comon, 1994] et SOBI [Belouchrani et al., 1997], trois méthodes exploitant la contrainte de nonnégativité: ACI nonnégative [Plumbley, 2003], FMN [Kim and Park, 2008] et semi-FMN [Ding et al., 2010]. Au vu des résultats obtenus, les méthodes classiques, surtout l'ACI nonnégative, SOBI et semi-FMN, n'étaient parfois pas capables de séparer les sources latentes. CoM2 et FMN ont pour leur part donné de bons résultats d'estimation, malgré quelques imperfections. Les méthodes proposées se sont comportées de la même manière et ont été plus aptes à fournir de meilleurs résultats que les méthodes classiques, en ce qui concerne la précision d'estimation des sources. Nos méthodes ont une complexité numérique modérée lorsque le nombre d'observations n'est pas trop grand. Cependant, leur complexité croît très vite lorsque le nombre d'observations augmente. Heureusement, en incorporant une étape de prétraitement de compression nonnégative, la complexité globale des méthodes proposées d'ACI semi-nonnégative reste acceptable. Ces applications de SAS ont démontré la validité et l'intérêt des algorithmes proposés. Les résultats expérimentaux ont aussi montré qu'en exploitant pleinement l'information préalable des données, telles que la nonnégativité de la matrice de mélange et l'indépendance statistique des sources, les méthodes proposées fournissent de meilleurs résultats d'estimation.

Chapitre 5: Conclusion et perspectives

Ce dernier chapitre conclu le rapport de thèse offrant une discussion sur les différentes contributions de la thèse, mais aussi sur les possibles extensions à venir.

Bibliographie

[Afsari, 2006] Afsari, B. (2006). Simple LU and QR based non-orthogonal matrix joint diagonalization. In *Independent Component Analysis and Blind Signal Separation, Lecture Notes in Computer Science*, volume 3889, pages 1–7.

[Belouchrani et al., 1997] Belouchrani, A., Abed-Meraim, K., Cardoso, J. F., and Moulines, E. (1997). A blind source separation technique using second-order statistics. *IEEE Transactions on Signal Processing*, 45(2):434–444.

[Cologner et al., 2014a] Cologner, J., Albera, L., Kachenoura, A., Noury, F., and Senhadji, L. (2014a). Semi-nonnegative joint diagonalization by congruence and semi-nonnegative ICA. *Signal Processing*, 105:185–197.

[Cologner et al., 2014b] Cologner, J., Fargeas, A., Kachenoura, A., Wang, L., Senhadji, L., De Crevoisier, R., Acosta, O., and Albera, L. (2014b). A novel classification method for prediction of rectal bleeding in prostate cancer radiotherapy based on a semi-nonnegative ICA of 3D planned dose distributions. Accepted for publication in *IEEE Journal of Biomedical and Health Informatics*.

[Cologner et al., 2014c] Cologner, J., Karfoul, A., Albera, L., and Comon, P. (2014c). Line search and trust region strategies for canonical decomposition of semi-nonnegative semi-symmetric 3rd order tensors. *Linear Algebra and its Applications*, 450(1):334–374.

[Comon, 1994] Comon, P. (1994). Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314.

[Comon and Jutten, 2010] Comon, P. and Jutten, C., editors (2010). *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Academic Press.

[Ding et al., 2010] Ding, C., Li, T., and Jordan, M. (2010). Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):45–55.

[Jutten and Hérault, 1991] Jutten, C. and Hérault, J. (1991). Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24(1):1–10.

[Kim and Park, 2008] Kim, H. and Park, H. (2008). Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM Journal on Matrix Analysis and Applications*, 30(2):713–730.

[Lee and Seung, 1999] Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

[Maurandi et al., 2013] Maurandi, V., Luigi, C. D., and Moreau, E. (2013). Fast Jacobi like algorithms for joint diagonalization of complex symmetric matrices. In *EUSIPCO'13, Proceedings of the 21th European Signal Processing Conference*, pages 1–5, Marrakech, Morocco.

[Plumbley, 2003] Plumbley, M. D. (2003). Algorithms for nonnegative independent component analysis. *IEEE Transactions on Neural Networks*, 14(3):534–543.

[Wang et al., 2013] Wang, L., Albera, L., Kachenoura, A., Shu, H. Z., and Senhadji, L. (2013). Nonnegative joint diagonalization by congruence based on LU matrix factorization. *IEEE Signal Processing Letters*, 20(8):807–810.

[Yeredor, 2002] Yeredor, A. (2002). Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation. *IEEE Transactions on Signal Processing*, 50(7):1545–1553.

[Ziehe et al., 2004] Ziehe, A., Laskov, P., Nolte, G., and Muller, K.-R. (2004). A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation. *Journal of Machine Learning Research*, 5:777–800.

Nonnegative joint diagonalization by congruence for semi-nonnegative independent component analysis

A Dissertation
Presented to the Faculty of the Graduate School
of Université de Rennes 1
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

by
Lu WANG
November 2014

to my wife and parents

Contents

Bibliographie **5**

Table of contents **I**

Abstract/Résumé **VI**

Acronyms and Notations **IX**

List of figures **XI**

List of tables **XII**

1 Introduction **1**
1.1 Blind source separation problem 1
1.2 Independent component analysis 2
1.2.1 Identifiability and preprocessing 3
1.2.2 Non-Gaussianity, maximum likelihood and mutual information . . 3
1.2.3 ICA by tensor decomposition of cumulant arrays 6
1.3 From ICA to joint diagonalization by congruence 10
1.3.1 JDC problem 11
1.3.2 Uniqueness and identifiability 11
1.3.3 JDC algorithms 12
1.4 Nonnegative matrix factorization 17
1.4.1 Existence and uniqueness 18
1.4.2 Basic NMF algorithms 19
1.4.3 Semi-NMF algorithms 21
1.5 Nonnegative independent component analysis 22
1.5.1 Preprocessing and identifiablity 22
1.5.2 NICA algorithms 23
1.6 Semi-nonnegative ICA and nonnegative JDC 24
1.6.1 The semi-nonnegative ICA problem 24
1.6.2 Nonnegative JDC algorithms 25
1.6.3 Contribution and outline of thesis 26
1.7 List of publications 29

2 Nonnegative joint diagonalization by congruence **31**
2.1 Preliminaries . . . 31
2.1.1 LU and QR matrix factorizations . . . 31
2.1.2 A Jacobi-like optimization procedure . . . 34
2.1.3 The alternating direction method of multipliers . . . 35
2.2 The $\text{JDC}^{+}_{\text{LU-3}}$ algorithm . . . 36
2.2.1 Algorithm derivation . . . 36
2.2.2 Generalization to the non-square case . . . 39
2.2.3 Practical issues . . . 39
2.3 The $\text{iJDC}^{+}_{\text{LU-1}}$ and $\text{iJDC}^{+}_{\text{QR-1}}$ algorithms . . . 40
2.3.1 Algorithm derivation . . . 40
2.3.2 Practical issues . . . 46
2.3.3 Discussions . . . 47
2.4 The $\text{JDC}^{+}_{\text{LU-1}}$ and $\text{JDC}^{+}_{\text{LU-2}}$ algorithms . . . 48
2.4.1 Algorithm derivation . . . 48
2.4.2 Generalization to the non-square case . . . 53
2.4.3 Practical issues . . . 54
2.5 The $\text{JDC}^{+}_{\text{ADMM-3}}$ algorithm . . . 55
2.5.1 Algorithm derivation . . . 55
2.5.2 Convergence to KKT Points . . . 57
2.6 Numerical complexity analysis . . . 58
2.6.1 The $\text{JDC}^{+}_{\text{LU-3}}$ algorithm . . . 59
2.6.2 The $\text{iJDC}^{+}_{\text{LU-1}}$ and $\text{iJDC}^{+}_{\text{QR-1}}$ algorithms . . . 59
2.6.3 The $\text{JDC}^{+}_{\text{LU-1}}$ and $\text{JDC}^{+}_{\text{LU-2}}$ algorithms . . . 60
2.6.4 The $\text{JDC}^{+}_{\text{ADMM-3}}$ algorithm . . . 60

3 Numerical simulations **63**
3.1 Experimental setup . . . 63
3.2 Convergence property . . . 64
3.2.1 Test on square matrices . . . 64
3.2.2 Test on non-square matrices . . . 65
3.3 Influence of SNR . . . 67
3.3.1 Test on square matrices . . . 69
3.3.2 Test on non-square matrices . . . 69
3.4 Influence of number of input matrices . . . 71
3.4.1 Test on square matrices . . . 71
3.4.2 Test on non-square matrices . . . 71
3.5 Influence of coherence of matrix $\boldsymbol{A}$ . . . 74
3.5.1 Test on square matrices . . . 75
3.5.2 Test on non-square matrices . . . 75
3.6 Influence of modulus of uniqueness . . . 76
3.6.1 Test on square matrices . . . 77
3.6.2 Test on non-square matrices . . . 79
3.7 Concluding remark . . . 79

4 Blind source separation applications **81**
4.1 Separation of simulated MRS signal . . . 82
4.1.1 Influence of the number of observations . . . 86
4.1.2 Influence of SNR . . . 87
4.2 Automatic music transcription . . . 88
4.3 Document restoration . . . 94
4.4 Concluding remark . . . 95

5 Conclusion and perspectives **97**
5.1 Conclusion . . . 97
5.2 Forthcoming work . . . 99

A Proofs of propositions **101**
A.1 Proof of proposition 2.1 . . . 101
A.2 Proof of proposition 2.2 . . . 101
A.3 Proof of proposition 2.3 . . . 102
A.4 Proof of proposition 2.4 . . . 102
A.5 Proofs of propositions 2.6 and 2.7 . . . 102
A.6 Proof of proposition 2.8 . . . 104
A.7 Proof of proposition 2.9 . . . 104

B Nonnegative compression method for SeNICA **107**
B.1 Problem formulation . . . 107
B.2 Algorithm derivation . . . 109
B.2.1 Step 1: estimation of the Givens rotation matrices . . . 109
B.2.2 Step 2: estimation of the elementary upper triangular matrices . . . 110

C Pseudo-codes of algorithms **113**
C.1 Pseudo-code of the cyclic Jacobi-like iteration . . . 113
C.2 Pseudo-code of the $\text{JDC}^{+}_{\text{LU-3}}$ algorithm . . . 114
C.3 Pseudo-code of the $\text{iJDC}^{+}_{\text{LU-1}}$ algorithm . . . 115
C.4 Pseudo-code of the $\text{iJDC}^{+}_{\text{QR-1}}$ algorithm . . . 116
C.5 Pseudo-code of the $\text{JDC}^{+}_{\text{LU-1}}$ algorithm . . . 117
C.6 Pseudo-code of the $\text{JDC}^{+}_{\text{LU-2}}$ algorithm . . . 118
C.7 Pseudo-code of the $\text{JDC}^{+}_{\text{ADMM-3}}$ algorithm . . . 119
C.8 Pseudo-code of the NN-COMP algorithm . . . 120

Bibliography **121**

Abstract

The Joint Diagonalization of a set of matrices by Congruence (JDC) appears in a number of signal processing problems, such as in Independent Component Analysis (ICA). Recent developments in ICA under the nonnegativity constraint of the mixing matrix, known as semi-nonnegative ICA, allow us to obtain a more realistic representation of some real-world phenomena, such as audios, images and biomedical signals. Consequently, during this thesis, the main objective was not only to design and develop semi-nonnegative ICA methods based on novel nonnegative JDC algorithms, but also to illustrate their interest in applications involving Blind Source Separation (BSS). The proposed nonnegative JDC algorithms belong to two fundamental strategies of optimization. The first family containing five algorithms is based on the Jacobi-like optimization. The nonnegativity constraint is imposed by means of a square change of variable, leading to an unconstrained problem. The general idea of the Jacobi-like optimization is to factorize the matrix variable as a product of a sequence of elementary matrices which is defined by only one parameter, then to estimate these elementary matrices one by one in a specific order. The second family containing one algorithm is based on the alternating direction method of multipliers. Such an algorithm is derived by successively minimizing the augmented Lagrangian function of the cost function with respect to the variables and the multipliers. Experimental results on simulated matrices show a better performance of the proposed algorithms in comparison with several classical JDC methods, which do not use the nonnegativity as constraint prior. It appears that our methods can achieve a better estimating accuracy particularly in difficult contexts, for example, for a low signal-to-noise ratio, a small number of input matrices and a high coherence level of matrix. Then we show the interest of our approaches in solving real-life problems. To name a few, we are interested in i) the analysis of the chemical compounds in the magnetic resonance spectroscopy, ii) the identification of the harmonically fixed spectral profiles (such as piano notes) of a piece of signal-channel music record by decomposing its spectrogram, iii) the partial removal of the show-through effect of digital images, where the show-through effect were caused by scanning a semi-transparent paper. These applications demonstrate the validity and improvement of our algorithms, comparing with several state-of-the-art BSS methods.

Rèsumé

La Diagonalisation Conjointe par Congruence (DCC) d'un ensemble de matrices apparaît dans nombres de problèmes de traitement du signal, tels qu'en Analyse en Composantes Indépendantes (ACI). Les développements récents en ACI sous contrainte de nonnégativité de la matrice de mélange, nommée ACI semi-nonnégative, permettent de tirer profit d'une modélisation physique réaliste des phénomènes observés tels qu'en audio, en traitement d'image ou en ingénierie biomédicale. Par conséquent, durant cette thèse, l'objectif principal était non seulement de concevoir et développer des algorithmes d'ACI semi-nonnégative basés sur de nouvelles méthodes de DCC nonnégative où la matrice de passage recherchée est nonnégative, mais également d'illustrer leur intérêt dans le cadre d'applications pratiques de séparation de sources. Les algorithmes de DCC nonnégative proposés exploitent respectivement deux stratégies fondamentales d'optimisation. La première famille d'algorithmes comprend cinq méthodes semi-algébriques, reposant sur la méthode de Jacobi. Cette famille prend en compte la nonnégativité par un changement de variable carré, permettant ainsi de se ramener à un problème d'optimisation sans contrainte. L'idée générale de la méthode de Jacobi est de i) factoriser la matrice recherchée comme un produit de matrices élémentaires, chacune n'étant définie que par un seul paramètre, puis ii) d'estimer ces matrices élémentaires l'une après l'autre dans un ordre spécifique. La deuxième famille compte un seul algorithme, qui utilise la méthode des directions alternées. Un tel algorithme est obtenu en minimisant successivement le Lagrangien augmenté par rapport aux variables et aux multiplicateurs. Les résultats expérimentaux sur les matrices simulées montrent un gain en performance des algorithmes proposés par comparaison aux méthodes DCC classiques, qui n'exploitent pas la contrainte de nonnégativité. Il semble que nos méthodes peuvent fournir une meilleure précision d'estimation en particulier dans des contextes difficiles, par exemple, pour de faibles valeurs de rapport signal sur bruit, pour un petit nombre de matrices à diagonaliser et pour des niveaux élevés de cohérence de la matrice de passage. Nous avons ensuite montré l'intérêt de nos approches pour la résolution de problèmes pratiques de séparation aveugle de sources. Pour n'en citer que quelques-uns, nous sommes intéressés à i) l'analyse de composés chimiques en spectroscopie par résonance magnétique, ii) l'identification des profils spectraux des harmoniques (par exemple, de notes de piano) d'un morceau de musique mono-canal par décomposition du spectrogramme, iii) l'élimination partielle du texte se trouvant au verso d'une feuille de papier fin. Ces applications démontrent la validité et l'intérêt de nos algorithmes en comparaison des méthodes classique de séparation aveugle de source.

Acronyms and Notations

Acronyms

Signal processing area

- **CPD**: Canonical Polyadic Decomposition
- **BSS**: Blind Source Separation
- **ICA**: Independent Component Analysis
- **INDSCAL**: INdividual Differences in SCALing
- **JDC**: Joint Diagonalization by Congruence
- **NJDC**: Nonnegative Joint Diagonalization by Congruence
- **MI**: Mutual Information
- **ML**: Maximum Likelihood
- **NICA**: Nonnegative Independent Component Analysis
- **NMF**: Nonnegative Matrix Factorization
- **PCA**: Principal Component Analysis
- **SeNICA**: Semi-Nonnegative Independent Component Analysis

Statistics

- **SO**: Second Order
- **TO**: Third Order
- **FO**: Fourth Order
- **HO**: Higher Order
- **HOS**: Higher Order Statistics

- **iid**: independent and identically distributed
- **pdf**: probability density function

Methods

- **AC**: Alternating Columns
- **ACDC**: Alternating Columns and Diagonal Center
- **ADMM**: Alternating Direction Method of Multipliers
- **ALM**: Augmented Lagrangian Multiplier
- **ALS**: Alternating Least Squares
- **ANLS**: Alternating Nonnegative Least Squares
- **CLU**: Coupled LU matrix factorization
- **DC**: Diagonal Center
- **DIEM**: DIagonalization using Equivalent Matrices
- **EJD**: Exact Joint Diagonalization
- **ELS**: Enhanced Line Search
- **EVD**: EigenValue Decomposition
- **FFDIAG**: Fast Frobenius DIAGonalization
- **GEVD**: Generalized EigenValue Decomposition
- **HALS**: Hierarchical Alternating Least Squares
- **InforMax**: Information Maximization
- **JAD**: Joint Approximate Diagonalization
- **JADE**: Joint Approximate Diagonalization of Eigen-matrices
- **LM**: Levenberg Marquardt
- **MU**: Multiplicative Updates
- **NLS**: Nonnegativity constrained Least-Square
- **NN-COMP**: NonNegative COMPression
- **QDIAG**: Quadratic DIAGonalization
- **SOBI**: Second Order Blind Identification
- **STOTD**: Simultaneous Third-Order Tensor Diagonalization

- **SVD**: Singular Value Decomposition
- **UWEDGE**: Uniformly Weighted Exhaustive Diagonalization with Gauss itErations

Others

- **CPM**: Completely Positive Matrix
- **ECG**: ElectroCardioGraphy
- **EEG**: ElectroEncephalography
- **flop**: floating point operation
- **GB**: GigaByte
- **GKLD**: Generalized Kullback-Leibler Divergence
- **KKT**: Karush–Kuhn–Tucker
- **MEG**: MagnetoEncephaloGraphy
- **MRS**: Magnetic Resonance Spectroscopy
- **NP**: Non-deterministic Polynomial time
- **SED**: Square of Euclidean Distance
- **SNR**: Signal to Noise Ratio

Notations

Typography

- $\mathbb{R}^{N_1 \times N_2 \times \cdots \times N_i}$ denotes the set of real-valued $(N_1 \times N_2 \times \cdots \times N_i)$ arrays
- $\mathbb{R}_+^{N_1 \times N_2 \times \cdots \times N_i}$ denotes the set of nonnegative real-valued $(N_1 \times N_2 \times \cdots \times N_i)$ arrays
- vectors are denoted by bold lowercase letters ($\boldsymbol{a}$, $\boldsymbol{b}$, ...)
- matrices are denoted by bold uppercase letters ($\boldsymbol{A}$, $\boldsymbol{B}$, ...)
- HO arrays are denoted by bold calligraphic letters ($\boldsymbol{\mathcal{A}}$, $\boldsymbol{\mathcal{B}}$, ...)
- the (i, j)-th entry of a matrix $\boldsymbol{A}$ is symbolized by $A_{i,j}$
- the (i, j, k)-th component of a third order array $\boldsymbol{\mathcal{B}}$ is symbolized by $\mathcal{B}_{i,j,k}$
- $\boldsymbol{A}_{:,i}$ (also $\boldsymbol{a}_i$ for simplicity) denotes the i-th column vector of matrix $\boldsymbol{A}$; $\boldsymbol{A}_{i,:}$ denotes the i-th row vector of matrix
- $\widehat{\boldsymbol{A}}$ denotes the estimate of $\boldsymbol{A}$

Operators

- $\circ$ denotes the outer product
- $\bullet$ denotes the scalar product
- $\odot$ stands for the Khatri-Rao product
- $\boxdot$ stands for the Hadamard product (element-wise product)
- $\oslash$ stands for the matrix element-wise division
- $^{\mathsf{T}}$ is the transpose operator
- $^{-1}$ is the inverse operator for square matrices
- $^{-\mathsf{T}}$ denotes the inverse after the transpose operator
- $^{\sharp}$ is the Moore-Penrose pseudo-inverse operator
- vec($\boldsymbol{A}$) reshapes a matrix into a column vector by stacking its columns vertically
- det($\boldsymbol{A}$) is the determinate of a square matrix $\boldsymbol{A}$
- diag($\boldsymbol{A}$) returns a matrix comprising only the diagonal elements of matrix $\boldsymbol{A}$
- Diag($\boldsymbol{b}$) gives a diagonal matrix whose diagonal elements are given by the vector $\boldsymbol{b}$
- off($\boldsymbol{A}$) vanishes the diagonal elements of the input matrix $\boldsymbol{A}$
- E{.} denotes the mathematical expectation
- cum{.} denotes the cumulant function

Usual matrices

- $\mathbf{0}_N$: N-dimensional column vector of zeros
- $\mathbf{0}_{N,P}$: $(N \times P)$-dimensional matrix of zeros
- $\boldsymbol{I}_N$: $(N \times N)$-dimensional identity matrix

Variables

- N: number of observations
- P: number of sources
- M: number of sample points
- K: number of matrices to be jointly diagonalized

Norms

- $|.|$: absolute value
- $\|.\|$: Euclidean norm
- $\|.\|_F$: Frobenius norm

List of Figures

3.1 The convergence in terms of minimizing the cost functions for estimating a (3×3) square nonnegative joint transformation matrix $\boldsymbol{A}$ from 5 input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (3×3) with three different SNR values. . 66

3.2 The convergence in terms of minimizing the cost functions for estimating a (5×3) non-square nonnegative joint transformation matrix $\boldsymbol{A}$ from 5 input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (5×5) with three different SNR values. 68

3.3 Influence of SNR for estimating a (3×3) square nonnegative joint transformation matrix $\boldsymbol{A}$ from 5 input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (3×3). 70

3.4 Influence of SNR for estimating a (5×3) non-square nonnegative joint transformation matrix $\boldsymbol{A}$ from 5 input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (5×5). 72

3.5 Influence of number K of the input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (3×3) for estimating a (3×3) square nonnegative joint transformation matrix $\boldsymbol{A}$ with SNR = 10 dB. 73

3.6 Influence of number K of the input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (5×5) for estimating a (5×3) square nonnegative joint transformation matrix $\boldsymbol{A}$ with SNR = 10 dB. 74

3.7 Influence of coherence of the column vectors of $\boldsymbol{A}$ for estimating a (3×3) square nonnegative joint transformation matrix $\boldsymbol{A}$ from 5 input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (3×3) with SNR = 10 dB. 76

3.8 Influence of coherence of the column vectors of $\boldsymbol{A}$ for estimating a (5×3) non-square nonnegative joint transformation matrix $\boldsymbol{A}$ from 5 input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (5×5) with SNR = 10 dB. 77

3.9 Influence of modulus of uniqueness for estimating a (3×3) square nonnegative joint transformation matrix $\boldsymbol{A}$ from 5 input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (3×3) with SNR = 10 dB. 78

3.10 Influence of modulus of uniqueness for estimating a (5×3) non-square nonnegative joint transformation matrix $\boldsymbol{A}$ from 5 input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (5×5) with SNR = 10 dB. 80

4.1 An example of the BSS results of 2 simulated MRS metabolites by the proposed four methods without using the nonnegative compression step. The number of observations is set to $N = 20$ and the SNR value is fixed to 10 dB. 83
4.2 Continuation of the first example of the BSS results of 2 simulated MRS metabolites by the proposed six methods with the nonnegative compression step. The number of observations is set to $N = 20$ and the SNR value is fixed to 10 dB. 84
4.3 Continuation of the first example of the BSS results of 2 simulated MRS metabolites by five classical BSS methods. The number of observations is set to $N = 20$ and the SNR value is fixed to 10 dB. The red dot circles indicate the significant estimation errors in figures (b) to (f). 85
4.4 Influence of the number of observations for estimating 2 MRS source metabolites with SNR = 10 dB. 87
4.5 Influence of SNR for estimating 2 MRS source metabolites from 20 observations. 89
4.6 Three representations of the music data. The music waveform is contaminated by pink noises with a SNR value of 5 dB. 90
4.7 Separated spectra obtained from six BSS methods from decomposing the music magnitude spectrogram. The red dot circles indicate the significant estimation errors in figures (d) to (f). 92
4.8 Temporal weights obtained by six BSS methods by decomposing the music magnitude spectrogram. Each row in each subfigure represents the amplitude of the corresponding column vector of matrix $\boldsymbol{A}$. 93
4.9 Restoration of the grayscale front and back sides of a real document by six BSS methods. The input images are obtained from [Tonazzini et al., 2007]. 95

B.1 An illustration of the proposed NN-COMP algorithm. (a) the input prewhitening matrix $\boldsymbol{W}$; (b) and (c) the proposed algorithm. (b) $\boldsymbol{V}$ is obtained by multiplying $\boldsymbol{W}$ by a Givens rotation matrix. (c) $\overline{\boldsymbol{W}}$ is obtained by multiplying $\boldsymbol{V}$ by an elementary upper triangular matrix. The blue circles denote the nonnegative values and the red circles denote the negative values. 108

List of Tables

1.1 Some different assumptions of NMF, semi-NMF, ICA, NICA and SeNICA for solving the BSS problem $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{\varepsilon}$. 25

1.2 The differences between the proposed six nonnegative JDC methods according to the cost functions, the optimization schemes and the assumption on $\boldsymbol{A}$. When we use the square change variable $\boldsymbol{A} = \boldsymbol{B}^{\boxdot 2}$, the LU and QR matrix factorizations are used to decompose $\boldsymbol{B}$, leading to the first five Jacobi-like optimization methods. The sixth method based on ADMM optimization estimates $\boldsymbol{A}$ without change of variable. [1] Regarding $\text{iJDC}^{+}_{\text{LU-1}}$ and $\text{iJDC}^{+}_{\text{QR-1}}$, the joint transformation matrix $\boldsymbol{A}$ is constrained to be square. And a preprocessing step that inverses each input matrix $\boldsymbol{C}^{(k)}$ is required. 28

2.1 Numerical complexities of the proposed methods and of ACDC [Yeredor, 2002], CLU [Maurandi et al., 2013], FFDIAG [Ziehe et al., 2004], LUJ1D, LUJ2D [Afsari, 2006] and $\text{LM}^{+}_{\text{sym}}$ [Coloigner et al., 2014c]. 62

3.1 Average numerical complexities of twelve methods in the convergence test for estimating a (3×3) square nonnegative joint transformation matrix $\boldsymbol{A}$ from 5 input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (3×3) with various SNR values. 67

3.2 Average numerical complexities of six methods in the convergence test for estimating a (5×3) non-square nonnegative joint transformation matrix $\boldsymbol{A}$ from 5 input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (5×5) with various SNR values. 67

4.1 The theoretical fundamental frequencies of four musical notes, namely C_5, A^{b}_{4}, F_4 and D^{b}_{4}, on an ideal piano [Jorgensen, 1991]. 90

4.2 Numerical complexities of six BSS methods in the experiment of automatic music transcription of a piano record. 91

4.3 Numerical complexities of six BSS methods in the experiment of document restoration. 94

Chapter 1

Introduction

1.1 Blind source separation problem

The Blind Source Separation (BSS) problem consists of retrieving unobserved sources from the observations. In this thesis report we focus on the following linear instantaneous mixture model in the field $\mathbb{R}$ of real numbers:

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{\varepsilon} \tag{1.1}$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_N]^\mathsf{T} \in \mathbb{R}^N$ is the observation vector, where $\boldsymbol{s} = [s_1, s_2, \ldots, s_P]^\mathsf{T} \in \mathbb{R}^P$ is the source vector, where $\boldsymbol{A} \in \mathbb{R}^{N \times P}$ is a mixing matrix, and where $\boldsymbol{\varepsilon} = [\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N]^\mathsf{T} \in \mathbb{R}^N$ is an additive noise vector. The blindness refers to the fact that neither the sources nor the mixing matrix is known.

The BSS problem appears in a wide range of application areas, where in general a multi-sensor system is involved, including remote sensing [Ma et al., 2014], telecommunication [Chevalier and Chevreuil, 2010], multiple-input multiple-output radar systems [Li and Liu, 1998], medical imaging [Wang et al., 2010], brain-computer interface [Kachenoura et al., 2008, Albera et al., 2008], image analysis [Be'ery and Yeredor, 2008], and audio separation [Fuentes et al., 2013]. For example, in the context of biomedical signal processing, the cutaneous ElectroCardioGraphy (ECG) measured on the mother's skin is a non-invasive technique that enables possibility of visualizing the electrical activity of a fetal heart, namely the fetal ECG. These cutaneous ECG recordings can be considered as an approximate noisy instantaneous linear mixture of potential signals generated by underlying bioelectric phenomena, such as the maternal and fetal heart activity, potential distributions generated by respiration and stomach activity [Zarzoso et al., 2000]. Therefore, in order to evaluate the health condition of the fetus and to reveal information for early diagnosis, BSS methods were proposed to estimate the fetal ECG from the recordings on the mother's skin [Zarzoso et al., 2000, De Lathauwer et al., 2000, Niknazar, 2013].

In the BSS model (1.1), it is easy to see that the scale and permutation indeterminacies will necessarily hold. More precisely, a scaling factor can always be exchanged between one source s_i and the corresponding column vector of the mixing matrix $\boldsymbol{A}$, namely $\boldsymbol{a}_i$, without changing the product. Similarly, permuting the order of the sources and the corresponding column vectors of the mixing matrix will not change the observations. Hence the amplitude, the sign and the order of the sources are not uniquely defined.

Therefore, BSS methods estimate the source vector up to a permutation and a scale diagonal matrix. Besides the scale and permutation indeterminacies, the BSS problem (1.1) is still ill-posed without additional assumptions on the model. In order to achieve relevant separating results with an actual application, it is necessary to utilize prior knowledge about the sources' characteristics or about the mixing model. For example, the sources could satisfy some basic assumptions, such as statistical independence, nonnegativity or sparsity. Similarly, the mixing matrix could also preserve some properties, such as nonnegativity or sparsity. The fundamental statistical independence assumption of the sources leads to the well-established theory of Independent Component Analysis (ICA) [Jutten and Hérault, 1991, Comon, 1994]. The nonnegativity assumption on the sources or on the mixing matrix or on both, gives birth to a wide range of successful BSS approaches, such as Nonnegative Matrix Factorization (NMF) [Lee and Seung, 1999] and Nonnegative ICA (NICA) [Plumbley, 2003]. During this thesis, the main objective was to develop BSS methods that retrieve statistical independence sources assuming a nonnegative mixing matrix. That is to say the mixing matrix in model (1.1) is constrained to have nonnegative components, while the values of the sources are unconstrained, giving rise to the concept of Semi-Nonnegative ICA (SeNICA). By now only a few methods have been proposed to address this problem [Cologner et al., 2010, Cologner, 2012, Cologner et al., 2014a, Cologner et al., 2014b]. In this report, we propose to solve the SeNICA problem by means of a class of new Nonnegative Joint Diagonalization by Congruence (NJDC) methods.

The rest of this section is organized into two parts. In the first part we provide a brief overview of some well-known BSS methods. Section 1.2 summarizes the existing results of ICA. Section 1.3 is devoted to get a closer look on Joint Diagonalization by Congruence (JDC) and its connection to ICA. Afterwards we introduce some nonnegativity constrained BSS approaches, including NMF in section 1.4 and NICA in section 1.5. In the second part the main principle of the proposed methods is presented. We address the proposed SeNICA model as well as the problem of NJDC in section 1.6. Furthermore, a list of my publications is given in section 1.7.

1.2 Independent component analysis

Generally, the methods based on the assumption that the sources are non-Gaussian (or at most one Gaussian source), and statistically independent, belong to the category of ICA approaches. ICA has emerged into a rapidly growing field due to the pioneering work of Jutten and Hérault [Hérault and Jutten, 1986, Jutten, 1987, Jutten and Hérault, 1991] and later to the work of Comon [Comon, 1994] who gave a mathematical framework of ICA. The ICA problem is defined as follows:

Problem 1.1. *Given M realizations of an observation random vector $\boldsymbol{x} \in \mathbb{R}^N$, find a mixing matrix $\boldsymbol{A} \in \mathbb{R}^{N\times P}$ and the M corresponding realizations of a source random vector $\boldsymbol{s} \in \mathbb{R}^P$, such that:*

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{\varepsilon} \tag{1.2}$$

where $\boldsymbol{s}$ has statistically independent components and $\boldsymbol{\varepsilon} \in \mathbb{R}^N$ is an additive noise random vector independent of $\boldsymbol{s}$.

Statistical independence is a reasonable assumption in many realistic problems, and ICA finds its applications in various areas including, but not limited to, radiocommunications [Chevalier and Chevreuil, 2010], biomedical engineering [Kachenoura et al., 2008, Albera et al., 2010, Albera et al., 2012, Hajipour Sardouie et al., 2014], remote sensing [Moussaoui et al., 2008], and face recognition [Bartlett et al., 2002, Kim et al., 2005]. Furthermore, many convenient measures of independence were proposed, such as kurtosis, negentropy, mutual information and cumulants, leading to different ICA algorithms.

In the rest of the section, after a brief discussion about the identifiability and preprocessing step of ICA, we review the following four popular ICA methodologies: ICA by maximization of non-Gaussianity, ICA by maximum likelihood estimation, ICA by minimization of mutual information, and ICA by tensor decomposition of cumulant arrays. For a broader survey we refer to [Comon and Jutten, 2010]. Overviews of some recent advances of ICA can be found in [Hyvärinen, 2013, Adalı et al., 2014].

1.2.1 Identifiability and preprocessing

In fact, if the ICA model (1.2) is noiseless, it is essentially identifiable under the following three conditions [Comon, 1994, Eriksson and Koivunen, 2004]:

1. The joint probability density function (pdf) of $\boldsymbol{s}$ factorizes as a product of the marginal pdfs of the sources s_i.

2. The pdf's of the source components s_i are non-Gaussian.

3. The mixing matrix $\boldsymbol{A}$ must be of full column rank.

The identifiability signifies that the estimated mixing matrix and sources are equal to the original mixing matrix and sources, respectively, up to the permutation and scale indeterminacies. The second assumption could be possibly relaxed for at most one Gaussian component. The identifiability issues of noisy ICA were addressed in [Davies, 2004].

Generally it is assumed that the observation vector has zero mean, which can be done by centering it. Moreover, optionally the use of Principal Component Analysis (PCA) as a spatial whitening has been adopted as a mean to reduce the search space of the mixing matrix to the group of unitary matrices, and also to reduce the effect of additive noise. PCA transforms the observation vector $\boldsymbol{x} \in \mathbb{R}^N$ into a vector $\boldsymbol{z} \in \mathbb{R}^P$ by multiplying it on the left a whitening matrix $\boldsymbol{W} \in \mathbb{R}^{P \times N}$, such that $\boldsymbol{W}(\boldsymbol{x} - \boldsymbol{\varepsilon})$ has an identity covariance matrix. The resulting mixing matrix $\boldsymbol{W}\boldsymbol{A} \in \mathbb{R}^{P \times P}$ is an orthogonal matrix. The row vectors of $\boldsymbol{W}$ are the P eigenvectors of the covariance matrix of $\boldsymbol{x}$ associated with the P largest magnitude eigenvalues, scaled by the inverse of the square root of the corresponding eigenvalues. It is noteworthy that the whitening step introduces additional errors that cannot be fully compensated for in the following ICA step.

1.2.2 Non-Gaussianity, maximum likelihood and mutual information

In this section, we review three fundamental ICA ideologies, including the ICA algorithms based on non-Gaussianity maximization, the ICA methods based on maximum likelihood estimation, and the ICA approaches based on mutual information minimization.

1.2.2.1 ICA by maximization of non-Gaussianity

The central limit theorem says that the distribution of a sum of statistically independent independent and identically distributed (iid) random variables tends toward a Gaussian distribution. The fundamental idea is that, if none of the sources has a Gaussian pdf, the estimated source $\widehat{s}_j = \boldsymbol{V}_{j,:}\boldsymbol{x}$ usually has a distribution that is closer to Gaussian than any of the original sources s_i and that becomes the least Gaussian when $\widehat{s}_j$ is equal to one of the original source s_i, i.e. when $\boldsymbol{V}_{j,:}$ is the j-th row vector of the demixing matrix $\boldsymbol{V}$. Therefore maximizing the non-Gaussianity of $\widehat{s}_j = \boldsymbol{V}_{j,:}\boldsymbol{x}$ with respect to $\boldsymbol{V}_{j,:}$ results in finding one of the sources. In practice, the kurtosis and the negentropy are two commonly used measures of non-Gaussianity.

- **Kurtosis**. The kurtosis of a random variable s, denoted by $\mathrm{kurt}(s)$, is defined as follows:

$$\mathrm{kurt}(s) = \mathrm{E}\{s^4\} - 3(\mathrm{E}\{s^2\})^2 \tag{1.3}$$

  assuming s having zero mean, where $\mathrm{E}\{\cdot\}$ denotes the mathematical expectation. The absolute value of kurtosis is zero for a Gaussian variable, and greater than zero for most non-Gaussian random variables [Hyvärinen and Oja, 2000]. Many ICA algorithms were proposed by maximizing the kurtosis. Delfosse and Loubaton proposed a deflation approach [Delfosse and Loubaton, 1995]. Hyvärinen and Oja gave a fast fixed-point algorithm in [Hyvärinen and Oja, 1997]. Papadias proposed a stochastic gradient method based on a multiuser kurtosis criterion [Papadias, 2000]. Zarzoso and Comon introduced a line search optimization for the kurtosis contrast function [Zarzoso and Comon, 2010].

- **Negentropy**. The differential entropy function $\mathrm{H}(\cdot)$ of a random vector $\boldsymbol{s}$ with a joint pdf denoted by $p_{\boldsymbol{s}}$ is defined as follows [Cover and Thomas, 1991]:

$$\mathrm{H}(\boldsymbol{s}) = -\int p_{\boldsymbol{s}}(\boldsymbol{\eta}) \log(p_{\boldsymbol{s}}(\boldsymbol{\eta}))\mathrm{d}\boldsymbol{\eta} \tag{1.4}$$

  Based on equation (1.4), the negentropy function $\mathrm{J}(\cdot)$ is defined as follows:

$$\mathrm{J}(\boldsymbol{s}) = \mathrm{H}(\boldsymbol{s}_{\mathrm{Gaussian}}) - \mathrm{H}(\boldsymbol{s}) \tag{1.5}$$

  where $\boldsymbol{s}_{\mathrm{Gaussian}}$ is a Gaussian random vector of the same covariance matrix as $\boldsymbol{s}$. Negentropy is always nonnegative and it is zero if and only if $\boldsymbol{s}$ has a Gaussian distribution. The computation of negentropy is difficult since it needs to estimate the pdf. Therefore, approximations of negentropy were proposed in order to simplify the computation, for example [Hyvärinen, 1997]:

$$\mathrm{J}(\boldsymbol{s}) \propto (\mathrm{E}\{\mathrm{G}(\boldsymbol{s})\} - \mathrm{E}\{\mathrm{G}(\boldsymbol{\nu})\})^2 \tag{1.6}$$

  where $\mathrm{G}(\cdot)$ is a non-quadratic functions, and where $\boldsymbol{\nu}$ is a vector of zero-mean unit-variance Gaussian variables. Based on the above approximation, Hyvärinen proposed a fast fixed-point ICA algorithm, known as FastICA [Hyvärinen, 1999]. Koldovský et al. provided an improved version of FastICA whose accuracy attained the Cramár-Rao lower bound [Koldovský et al., 2006].

1.2.2.2 ICA by maximum likelihood estimation

For the noiseless mixture $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}$, if the pdf is specified for each source, the distribution for the whole data set $\boldsymbol{x}$ is determined for any value of $\boldsymbol{A}$. It is denoted by $p_{\boldsymbol{x}|\boldsymbol{A}}$ and the Maximum Likelihood (ML) estimate of $\boldsymbol{A}$ is $\widehat{\boldsymbol{A}}_{\mathrm{ML}} = \mathrm{argmax}_{\boldsymbol{A}}\, p_{\boldsymbol{x}|\boldsymbol{A}}$ [Cardoso, 2010]. The ML is computationally complicated, since it requires to estimate the pdf's of the sources. There are two ways of avoiding the estimation of pdf's. The first way is to use the prior densities in the likelihood. The second way to is to approximate the densities of the variables using a family of simple densities that are specified by a limited number of parameters as described in [Girolami, 1998, Lee et al., 1999, Hyvärinen et al., 2001]. The ICA approach by ML estimation was originated by Pham et al. [Pham et al., 1992], and gained popularity after the introduction of the Information Maximization (InfoMax) principle by Bell and Sejnowski [Bell and Sejnowski, 1995]. The equivalence between the InfoMax principle and ML was proven in [Cardoso, 1997, Obradovic and Deco, 1998]. Many algorithms were proposed thereafter. Pham and Garat provided two methods by using prior distributions of the sources [Pham and Garat, 1997]. Boscolo et al. proposed an ICA method based on a nonparametric kernel density estimation technique [Boscolo et al., 2004]. Dégerine and Zaïdi derived an ML approach for Gaussian autoregressive sources [Dégerine and Zaïdi, 2004]. Kokkinakis and Nandi proposed to use generalized gamma densities in the ML estimation for ICA [Kokkinakis and Nandi, 2007]. Xue et al. solved the ML-based ICA by means of gradient equation [Xue et al., 2009]. Ge and Ma studied the spurious solutions of the ML approach [Ge and Ma, 2010]. Vía et al. provided an ICA method for quaternion signals [Vía et al., 2011].

1.2.2.3 ICA by minimization of mutual information

Mutual Information (MI) is a well-established information-theoretic measure of statistical dependence, based on which ICA approaches can be developed. MI between P random variables, s_i, $i \in \{1, \ldots, P\}$, is defined as follows:

$$\mathrm{I}(s_1, s_2, \ldots, s_P) = \sum_{i=1}^{P} \mathrm{H}(s_i) - \mathrm{H}(\boldsymbol{s}) \tag{1.7}$$

where the negentropy function $\mathrm{H}(\cdot)$ is defined in equation (1.4). MI is nonnegative and equal to zero if and only if the random variables are statistically independent. MI-based ICA methods intend to minimize the MI of the separated sources $\widehat{\boldsymbol{s}} = \boldsymbol{V}\boldsymbol{x}$ with respect to the demixing matrix $\boldsymbol{V}$, yielding the maximum independence.

MI can serve as a unifying framework for maximization of non-Gaussianity and ML estimation [Yang, 1997, Obradovic and Deco, 1998]. A large number of ICA methods are based on the minimization of MI. He et al. proposed a nonparametric algorithm by means of clustering-based multivariate density estimation [He et al., 2000]. Pham proposed a method based on higher order statistics [Pham, 2000], and a method for colored sources via the Gaussian MI criterion [Pham, 2001a]. Babaie-Zadeh et al. designed an algorithm based on differential MI [Babaie-Zadeh et al., 2004]. Kraskov et al. gave several MI estimators for ICA [Kraskov et al., 2004]. Babaie-Zadeh and Jutten introduced a nonparametric gradient method [Babaie-Zadeh and Jutten, 2005]. A fast fixed-point algorithm was provided by Hulle [Hulle, 2008]. Other methods include [Almeida, 2003, Pham, 2004, Li and Adalı, 2010, Rashid and Yu, 2012], to cite a few.

1.2.3 ICA by tensor decomposition of cumulant arrays

Higher-Order (HO) tensors are essentially HO arrays whose components have more than two indices, and they can be seen as HO extensions of vectors (first order arrays) and matrices (second order arrays). HO cumulant arrays can be considered as HO generalizations of the covariance matrix, and they are the measure of statistical dependence at higher order. An important class of ICA approaches, as well as the proposed methods in this thesis report, consists of using HO cumulant arrays.

1.2.3.1 Preliminaries

Now we introduce some basic definitions in multilinear algebra which are necessary for the cumulant-based ICA methods.

Definition 1.1. *The outer product* $\boldsymbol{\mathcal{T}} = \boldsymbol{u}^{(1)} \circ \boldsymbol{u}^{(2)} \circ \cdots \circ \boldsymbol{u}^{(q)}$ *of* q *vectors* $\boldsymbol{u}^{(i)} \in \mathbb{R}^{N_i}$, $i \in \{1, \ldots, q\}$, *is a* q*-th order array in* $\mathbb{R}^{N_1 \times N_2 \times \cdots \times N_q}$ *whose entries are defined by* $\mathcal{T}_{i_1, i_2, \ldots, i_q} = u^{(1)}_{i_1} u^{(2)}_{i_2} \cdots u^{(q)}_{i_q}$.

Definition 1.2. *Each* q*-th order array* $\boldsymbol{\mathcal{T}}$ *expressed as the outer product of* q *vectors is a rank-*1 q*-th order array.*

More generally, the rank of a q-th order array is defined as follows:

Definition 1.3. *The rank of an array* $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{N_1 \times N_2 \times \cdots \times N_q}$, *denoted by* $\mathrm{rank}(\boldsymbol{\mathcal{T}})$, *is the minimal number of rank-*1 *arrays belonging to* $\mathbb{R}^{N_1 \times N_2 \times \cdots \times N_q}$ *that yield* $\boldsymbol{\mathcal{T}}$ *in a linear combination.*

Despite the similarity between the definition of the tensor rank and its matrix counterpart, the rank of a HO array may exceed its dimensions [Comon et al., 2009]. An important concept of matrix rank named Kruskal rank is defined as follows [Harshman and Lundy, 1984]:

Definition 1.4. *The Kruskal rank of a matrix* $\boldsymbol{U}$, *denoted by* $\mathrm{rank}_{\mathrm{K}}(\boldsymbol{U})$, *is defined as the largest number* ℓ *such that every subset of* ℓ *columns of* $\boldsymbol{U}$ *is linearly independent.*

We can extract a matrix slice from a HO array by means of the following definition:

Definition 1.5. *A* q*-th order array matrix slice is a* 2*-dimensional section (fragment) of a* q*-th order array, obtained by fixing* $q - 2$ *of the* q *indices [Cichocki et al., 2009].*

Now we define the multiplication of a HO array with a matrix:

Definition 1.6. *The mode-*i *product of an array* $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{N_1 \times N_2 \times \cdots \times N_q}$ *by a matrix* $\boldsymbol{U} \in \mathbb{R}^{J_i \times N_i}$, *denoted by* $\boldsymbol{\mathcal{T}} \times_i \boldsymbol{U}$ *is an array in* $\mathbb{R}^{N_1 \times N_2 \times \cdots \times N_{i-1} \times J_i \times N_{i+1} \times \cdots \times N_q}$ *defined by:*

$$(\boldsymbol{\mathcal{T}} \times_i \boldsymbol{U})_{n_1, n_2, \ldots, j_i, \ldots, n_q} = \sum_{n_i=1}^{N_i} \mathcal{T}_{n_1, n_2, \ldots, n_i, \ldots, n_q} U_{j_i, n_i} \tag{1.8}$$

for all index values of j_i.

According to definitions 1.1, 1.2 and 1.3, the low-rank Canonical Polyadic Decomposition (CPD) model of a given HO array is defined by:

Definition 1.7. *Given an integer number P of components, the CPD model of a q-th order array $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{N_1 \times N_2 \times \cdots \times N_q}$ can be expressed as follows:*

$$\boldsymbol{\mathcal{T}} = \sum_{p=1}^{P} \boldsymbol{u}_p^{(1)} \circ \boldsymbol{u}_p^{(2)} \circ \cdots \circ \boldsymbol{u}_p^{(q)} + \boldsymbol{\mathcal{R}} \tag{1.9}$$

where the q-th order array $\boldsymbol{\mathcal{R}}$ represents the model residual.

The notation $\boldsymbol{\mathcal{T}} = [\![\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \ldots, \boldsymbol{U}^{(q)}]\!] + \boldsymbol{\mathcal{R}}$ refers to the CPD of $\boldsymbol{\mathcal{T}}$ by equation (1.9) with the associated loading matrices $\boldsymbol{U}^{(i)} = [\boldsymbol{u}_1^{(i)}, \cdots, \boldsymbol{u}_P^{(i)}] \in \mathbb{R}^{N_i \times P}$. When $\boldsymbol{\mathcal{R}}$ is a null array, we have an exact CPD. An exact CPD is considered to be essentially unique when it is only subject to scale and permutation indeterminacies. A sufficient condition of uniqueness was defined by means of Kruskal rank as presented in the following theorem [Sidiropoulos and Bro, 2000]:

Theorem 1.1. *The CPD of a rank-P q-th order array $\boldsymbol{\mathcal{T}} = [\![\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \ldots, \boldsymbol{U}^{(q)}]\!]$ is considered to be essentially unique when the following inequality holds:*

$$\sum_{i=1}^{q} \mathrm{rank_K}(\boldsymbol{U}^{(i)}) \geq 2P + (q-1) \tag{1.10}$$

More refined uniqueness conditions specifically for third-order arrays were addressed in [Kruskal, 1977, Ten Berge and Sidiropoulos, 2002, Jiang and Sidiropoulos, 2004, De Lathauwer, 2006].

The CPD problem and its algorithms have been studied extensively for the past four decades, see [Harshman, 1970, Harshman and Lundy, 1994, Comon, 2002, De Lathauwer et al., 2004, De Lathauwer, 2006, Rajih et al., 2008, Comon et al., 2009, Kolda and Bader, 2009, Karfoul et al., 2011, Röemer and Haardt, 2013, Hajipour Sardouie et al., 2013, Luciani and Albera, 2014, Comon, 2014] and references therein.

As a special form of the CPD of a third-order array where two loading matrices are equal, the low-rank INdividuals Differences in SCALing (INDSCAL) analysis [Carroll and Chang, 1970] is defined as follows:

Definition 1.8. *For a given P, corresponding to the number of rank-1 terms, the INDSCAL decomposition of a third-order array $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{N_1 \times N_1 \times N_3}$ can be expressed as:*

$$\boldsymbol{\mathcal{T}} = \sum_{p=1}^{P} \boldsymbol{u}_p^{(1)} \circ \boldsymbol{u}_p^{(1)} \circ \boldsymbol{u}_p^{(3)} + \boldsymbol{\mathcal{R}} \tag{1.11}$$

with the loading matrices $\boldsymbol{U}^{(1)} = [\boldsymbol{u}_1^{(1)}, \cdots, \boldsymbol{u}_P^{(1)}] \in \mathbb{R}^{N_1 \times P}$ and $\boldsymbol{U}^{(3)} = [\boldsymbol{u}_1^{(3)}, \cdots, \boldsymbol{u}_P^{(3)}] \in \mathbb{R}^{N_3 \times P}$, and $\boldsymbol{\mathcal{R}}$ being the residual term.

The INDSCAL decomposition can also be described by using the frontal slices of $\boldsymbol{\mathcal{T}}$:

$$\forall\, k \in \{1, 2, \cdots, N_3\},\ \boldsymbol{T}^{(k)} = \boldsymbol{\mathcal{T}}_{:,:,k} = \boldsymbol{U}^{(1)} \boldsymbol{D}_{U3}^{(k)} (\boldsymbol{U}^{(1)})^{\mathsf{T}} + \boldsymbol{R}^{(k)} \tag{1.12}$$

where $\boldsymbol{T}^{(k)} = \boldsymbol{\mathcal{T}}_{:,:,k}$ is the k-th frontal slice of $\boldsymbol{\mathcal{T}}$, where $\boldsymbol{D}_{U3}^{(k)} \in \mathbb{R}^{P \times P}$ is a diagonal matrix whose diagonal contains the elements of the k-th row vector of $\boldsymbol{U}^{(3)}$, and where $\boldsymbol{R}^{(k)} = \boldsymbol{\mathcal{R}}_{:,:,k}$ is the k-th frontal slice of $\boldsymbol{\mathcal{R}}$. In fact, equation (1.12) shows that the

INDSCAL decomposition problem is essentially a joint diagonalization by congruence problem which we will address in the next section. The uniqueness condition for the INDSCAL decomposition was established by Afsari [Afsari, 2008].

The q-th order cumulant array of a real random vector is essentially a q-th order array, and it can be related to moments of order lower than and equal to q using Leonov-Shiryaev formula [McCullagh, 1987]:

Property 1.1. *The q-th order cumulant array $\boldsymbol{\mathcal{C}}_{\boldsymbol{x}}^{(q)} \in \mathbb{R}^{N\times N\times\cdots\times N}$ of a real random vector $\boldsymbol{x} \in \mathbb{R}^N$ satisfies the following element-wise equation:*

$$\begin{aligned}\mathcal{C}_{\boldsymbol{x},n_1,n_2,\ldots,n_q}^{(q)} &= \operatorname{cum}\{x_{n_1}, x_{n_1}, \ldots, x_{n_q}\} \\ &= \sum_{\ell}(-1)^{\ell-1}(\ell-1)!\,\mathrm{E}\{\prod_{n\in I_1} x_n\}\,\mathrm{E}\{\prod_{n\in I_2} x_n\}\cdots\mathrm{E}\{\prod_{n\in I_\ell} x_n\}\end{aligned} \tag{1.13}$$

where the summation includes all possible partitions $\{I_1, I_2, \ldots, I_\ell\}$ $(1 \le \ell \le q)$ of the integers $\{n_1, n_2, \ldots, n_q\}$ (see [Karfoul et al., 2010] for more details).

For a real zero-mean random vector $\boldsymbol{x}$, the cumulants up to order four are expressed in details as follows:

$$\mathcal{C}_{\boldsymbol{x},n_1}^{(1)} = \mathrm{E}\{x_{n_1}\} \tag{1.14}$$

$$\mathcal{C}_{\boldsymbol{x},n_1,n_2}^{(2)} = \mathrm{E}\{x_{n_1}x_{n_2}\} \tag{1.15}$$

$$\mathcal{C}_{\boldsymbol{x},n_1,n_2,n_3}^{(3)} = \mathrm{E}\{x_{n_1}x_{n_2}x_{n_3}\} \tag{1.16}$$

$$\begin{aligned}\mathcal{C}_{\boldsymbol{x},n_1,n_2,n_3,n_4}^{(4)} &= \mathrm{E}\{x_{n_1}x_{n_2}x_{n_3}x_{n_4}\} - \mathrm{E}\{x_{n_1}x_{n_2}\}\,\mathrm{E}\{x_{n_3}x_{n_4}\} \\ &\quad - \mathrm{E}\{x_{n_1}x_{n_3}\}\,\mathrm{E}\{x_{n_2}x_{n_4}\} - \mathrm{E}\{x_{n_1}x_{n_4}\}\,\mathrm{E}\{x_{n_2}x_{n_3}\}\end{aligned} \tag{1.17}$$

HO cumulant arrays enjoy a number of important properties, which are listed as follows [Nikias and Mendel, 1993, De Lathauwer, 2010]:

1. *Symmetry*: real cumulant arrays are fully symmetric under an arbitrary permutation of the indices.

2. *Multi-linearity*: if a real random vector $\boldsymbol{x}$ admits the linear transformation $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}$, then the following equality holds:

$$\boldsymbol{\mathcal{C}}_{\boldsymbol{x}}^{(q)} = \boldsymbol{\mathcal{C}}_{\boldsymbol{s}}^{(q)} \times_1 \boldsymbol{A} \times_2 \boldsymbol{A} \cdots \times_q \boldsymbol{A} \tag{1.18}$$

3. *Partitioning of independent variables*: if a subset of p random variables x_{n_1}, x_{n_2}, …, x_{n_p} is statistically independent of the other variables, then their HO cumulants are equal to zero. Therefore, a HO cumulant array of a random vector that has mutually independent components is a diagonal array. In this case only the entries of which all the indices are equal can be different from zero.

4. *Sum of independent variables*: the cumulant array of a sum of independent random vectors is the sum of the individual cumulant arrays.

5. *Non-Gaussianity*: HO cumulant arrays of a Gaussian variable are zero arrays. In combination with the fourth property, for a random vector $\boldsymbol{x}$ corrupted by an additive Gaussian noise vector $\boldsymbol{\varepsilon}$, we have, for $q > 2$:

$$\boldsymbol{\mathcal{C}}_{\boldsymbol{x}+\boldsymbol{\varepsilon}}^{(q)} = \boldsymbol{\mathcal{C}}_{\boldsymbol{x}}^{(q)} + \boldsymbol{\mathcal{C}}_{\boldsymbol{\varepsilon}}^{(q)} = \boldsymbol{\mathcal{C}}_{\boldsymbol{x}}^{(q)} \tag{1.19}$$

Consequently, we can state that HO cumulants of a set of random variables give an indication of their mutual statistical dependence, and that the HO cumulant of a single random variable is some measure of its non-Gaussianity [Pesquet and Moreau, 2001, De Lathauwer, 2010]. Moreover, the above properties of a HO cumulant array determine the algebraic structure of that array, which establishes the algorithmic foundation of a family of well-known ICA methods and also of the proposed algorithms in this thesis report.

1.2.3.2 Cumulant-based ICA algorithms

Considering the ICA model (1.2) and using the properties of cumulants discussed in the previous section, the q-th order cumulant array of the observation vector $\boldsymbol{x}$ can be expressed as follows:

$$\boldsymbol{\mathcal{C}}_{\boldsymbol{x}}^{(q)} = \boldsymbol{\mathcal{C}}_{\boldsymbol{s}}^{(q)} \times_1 \boldsymbol{A} \times_2 \boldsymbol{A} \cdots \times_q \boldsymbol{A} + \boldsymbol{\mathcal{C}}_{\boldsymbol{\varepsilon}}^{(q)} = \sum_{p=1}^{P} \mathcal{C}_{\boldsymbol{s},p,\ldots,p}^{(q)} \, \boldsymbol{a}_p \circ \boldsymbol{a}_p \circ \cdots \circ \boldsymbol{a}_p + \boldsymbol{\mathcal{C}}_{\boldsymbol{\varepsilon}}^{(q)} \tag{1.20}$$

where $\boldsymbol{a}_p$ is the p-th column vector of the mixing matrix $\boldsymbol{A}$, where $\boldsymbol{\mathcal{C}}_{\boldsymbol{\varepsilon}}^{(q)}$ is the q-th order cumulant array of the noise vector, and where $\mathcal{C}_{\boldsymbol{s},p,\ldots,p}^{(q)}$ denotes the q-th order cumulant of the p-th source. The second equality holds when the sources are statistically independent, in which case the cumulant array $\boldsymbol{\mathcal{C}}_{\boldsymbol{s}}^{(q)}$ of the source vector is diagonal. Moreover, $\boldsymbol{\mathcal{C}}_{\boldsymbol{\varepsilon}}^{(q)}$ $(q > 2)$ vanishes if the noise has a Gaussian distribution. Equation (1.20) is essentially the CPD of a cumulant array. Under the uniqueness condition of the CPD on the matrix $\boldsymbol{A}$, and assuming that all the sources have non-zero q-th order cumulants $\mathcal{C}_{\boldsymbol{s},p,\ldots,p}^{(q)}$ and $\boldsymbol{\varepsilon}$ is a Gaussian noise vector, the mixing matrix $\boldsymbol{A}$ can be estimated uniquely by CPD of $\boldsymbol{\mathcal{C}}_{\boldsymbol{x}}^{(q)}$ $(q > 2)$ up to scale and permutation indeterminacies. Cardoso proposed a super-symmetric decomposition method for the Fourth-Order (FO) cumulant array [Cardoso, 1991]. Comon provided a CPD method for cumulant arrays but restricted the mixing matrix $\boldsymbol{A}$ to be of size (2×3) [Comon, 2004]. Albera et al. proposed an algorithm to compute the CPD of a Hermitian FO array and applied it the FO cumulant array of the observation array in order to identify $\boldsymbol{A}$ [Albera et al., 2005]. The method was extended to $2q$-th order Hermitian arrays by the same authors in [Albera et al., 2004], allowing for the identification of underdetermined mixing matrices (i.e. with more columns than rows). Ferréol et al. showed how to combine several time-delayed FO cumulant arrays for the identification of underdetermined mixing matrices, provided that one of the FO cumulant arrays was positive (or negative) semi-definite [Ferréol et al., 2005]. De Lathauwer and Castaing proposed to decompose jointly several Second-Order (SO) time-delayed cumulant arrays [De Lathauwer and Castaing, 2008]. Karfoul et al. provided a joint CPD method of several HO cumulant arrays relaxing the positive (or negative) semi-definite assumption [Karfoul et al., 2010]. A generalization of the Enhanced Line Search Alternating Least Squares (ELS-ALS) method [Rajih et al., 2008] to the case of complex-valued arrays of order strictly greater than 3 and a new algorithm for the CPD of Hermitian HO arrays were proposed in [Karfoul et al., 2011].

On the other side, considering the cumulant array of the estimated source vector $\widehat{\boldsymbol{s}} = \boldsymbol{V}\boldsymbol{x}$:

$$\boldsymbol{\mathcal{C}}_{\widehat{\boldsymbol{s}}}^{(q)} = \boldsymbol{\mathcal{C}}_{\boldsymbol{x}}^{(q)} \times_1 \boldsymbol{V} \times_2 \boldsymbol{V} \cdots \times_q \boldsymbol{V} \tag{1.21}$$

the demixing matrix $\boldsymbol{V}$ can be estimated by transforming $\boldsymbol{\mathcal{C}}_{\widehat{\boldsymbol{s}}}^{(q)}$ into a diagonal array, yielding $\widehat{\boldsymbol{s}}$ having statistically independent components. Suppose that the cumulant array is computed on the whitened observation vector. The search space of $\boldsymbol{V}$ is then reduced to the group of $(P \times P)$ unitary matrices. A $(P \times P)$ unitary matrix can be parameterized as a product of $P(P-1)/2$ Givens rotation matrices. Comon proposed to use the FO cumulant array and to estimate these Givens rotation matrices sequentially with Jacobi iterations. In each Jacobi iteration, the parameter of one rotation matrix can be estimated either by maximizing the total sum of squares of the diagonal components of the cumulant array $\boldsymbol{\mathcal{C}}_{\widehat{\boldsymbol{s}}}^{(q)}$ [Comon, 1994], or by maximizing the trace of that cumulant array if all the source cumulants are known to be positive [Comon and Moreau, 1997]. The former algorithm is known as CoM2, and the latter is called CoM1, where the numbers "2" and "1" indicate that the algorithms maximize the 2-norm and 1-norm of the diagonal components, respectively. Cardoso and Souloumiac proposed to maximize another cost function (called contrast [Comon, 1994, Moreau, 2001]) based on FO cumulants. This maximization problem was reformulated as the Joint Approximate Diagonalization of Eigen-matrices, leading to the JADE algorithm [Cardoso and Souloumiac, 1993]. On the other hand, De Lathauwer et al. considered the FO cumulant array as a stack of N Third Order (TO) arrays of dimension $(N \times N \times N)$, and proposed an ICA algorithm based on Simultaneous Third-Order Tensor Diagonalization (STOTD) [De Lathauwer et al., 2001]. Moreau extended the JADE and STOTD methods to any HO cumulant by providing a new family of contrast functions that served as a unifying framework for JADE and STOTD [Moreau, 2001]. Moreover, he revealed a link between these new contrast functions and a joint diagonalization criterion, yielding a novel method called eJADE. Tobias and Wiskott extended the CoM2 algorithm by incorporating both the TO and FO cumulant arrays [Blaschke and Wiskott, 2004]. However, it was shown in [Moreau, 2006] that the contrast function used by Tobias and Wiskott was a particular case of the general contrast introduced by Moreau in [Moreau, 2001]. Another ICA method called Second Order Blind Identification (SOBI) for separating stationary sources with different spectral contents was proposed by Belouchrani et al., which aimed at jointly diagonalizing a set of whitened SO time-delayed cumulant arrays [Belouchrani et al., 1997].

As informed in the previous paragraphs, Joint Diagonalization by Congruence (JDC) is the core step in quite a number of cumulant-based ICA methods, such as JADE [Cardoso and Souloumiac, 1993], eJADE [Moreau, 2001] and SOBI [Belouchrani et al., 1997]. The connection between ICA and JDC, as well as the JDC algorithms, will be addressed in the next section.

1.3 From ICA to joint diagonalization by congruence

Joint diagonalization of cumulant matrix slices by congruence is a principal procedure for many ICA algorithms [Moreau and Adalı, 2013]. In this section, firstly we study the matrix-wise structure of cumulant arrays and show that the matrix slices of cumulant arrays preserve a joint congruence transformation structure. The mixing matrix of ICA can be estimated by means of a JDC technique. Secondly, the uniqueness and identifiability issues are briefly addressed. Thirdly, we provide an overview of existing JDC algorithms.

1.3.1 JDC problem

A matrix slice of a q-th order cumulant array $\boldsymbol{\mathcal{C}}_{\boldsymbol{x}}^{(q)}$ ($q > 2$) can be obtained by varying its first two indices and picking one particular value for the other indices. If $\boldsymbol{x}$ obeys the ICA model (1.2) and is contaminated by a Gaussian noise, from equation (1.20), the matrix slices have the following structure:

$$\boldsymbol{C}_{\boldsymbol{x};:,:,n_3,\ldots,n_q}^{(q)} = \sum_{n_1=1}^{P} (\mathcal{C}_{\boldsymbol{s},n_1,\ldots,n_1}^{(q)} A_{n_3,n_1} \cdots A_{n_q,n_1})\, \boldsymbol{a}_{n_1} \circ \boldsymbol{a}_{n_1} = \boldsymbol{A}\, \boldsymbol{D}^{(n_3,\ldots,n_q)} \boldsymbol{A}^{\mathsf{T}} \tag{1.22}$$

where A_{n_i,n_j} is the (n_i, n_j)-th component of $\boldsymbol{A}$, where $\boldsymbol{a}_{n_1}$ denotes the n_1-th column vector of $\boldsymbol{A}$, where $\boldsymbol{D}^{(n_3,\ldots,n_q)} \in \mathbb{R}^{P\times P}$ are diagonal matrices, and where the equality $\boldsymbol{a}_{n_1} \circ \boldsymbol{a}_{n_1} = \boldsymbol{a}_{n_1}\boldsymbol{a}_{n_1}^{\mathsf{T}}$ is used implicitly. Equation (1.22) shows that the matrix slices of cumulant arrays share the following joint congruence transformation:

$$\boldsymbol{C}^{(k)} = \boldsymbol{A}\boldsymbol{D}^{(k)}\boldsymbol{A}^{\mathsf{T}}, \quad k \in \{1,\ldots,K\} \tag{1.23}$$

where $\boldsymbol{D}^{(k)} \in \mathbb{R}^{P\times P}$ are diagonal matrices, and where $K = N^{q-2}$ is the total number of matrix slices from a q-th order cumulant array. In this case, K matrices $\boldsymbol{C}^{(k)} \in \mathbb{R}^{N\times N}$ are jointly diagonalizable and $\boldsymbol{A} \in \mathbb{R}^{N\times P}$ is called the joint transformation matrix. It is noteworthy that in order to use HO cumulant arrays and their matrix slices in ICA, the sources are assumed to be mutually independent and non-Gaussian.

Consequently, in order to identify the mixing matrix $\boldsymbol{A}$ from the cumulant matrix slices, we aim at solving the following JDC problem:

Problem 1.2. *Given K symmetric matrices $\boldsymbol{C}^{(k)} \in \mathbb{R}^{N\times N}$ and an integer P, find a joint transformation matrix $\boldsymbol{A} \in \mathbb{R}^{N\times P}$ and K diagonal matrices $\boldsymbol{D}^{(k)} \in \mathbb{R}^{P\times P}$ such that these matrices satisfy equation* (1.23).

In most papers, the terms "joint diagonalization" and "simultaneous diagonalization" were utilized to state the JDC problem, such as in [Yeredor, 2002]. We propose to emphasis the "Congruence" transformation in the Joint Diagonalization problem, leading to the acronym "JDC", in oder to avoid confusion with the joint diagonalization by equivalence problem $\boldsymbol{C}^{(k)} = \boldsymbol{A}\boldsymbol{D}^{(k)}\boldsymbol{A}^{-1}$, which is an essentially different problem when $\boldsymbol{A}$ is not unitary [Luciani and Albera, 2010,Luciani and Albera, 2011,Luciani and Albera, 2014].

1.3.2 Uniqueness and identifiability

If the ICA model (1.2) is noiseless and the mixing matrix $\boldsymbol{A} \in \mathbb{R}^{N\times P}$ is of full column rank with $N > P$, one can always find P of the N observations which are linearly independent and discard the other observed mixtures without loss of information, leading to a square mixing matrix. More generally, an appropriate $(P \times N)$ matrix transformation can be applied to the N observations in order to merge them and to reduce the dimension of the observation vector. Therefore, without loss of generality, we can assume that for a noiseless JDC problem the joint transformation matrix $\boldsymbol{A}$ is a nonsingular square matrix of dimension $(N \times N)$.

Afsari proposed a uniqueness condition for the noiseless JDC problem by measuring the coherence of the K diagonal matrices $\boldsymbol{D}^{(k)}$ [Afsari, 2008]. Let $\boldsymbol{D}$ be a $(K\times N)$ matrix

whose (k, n)-th component is defined as follows:

$$D_{k,n} = D_{n,n}^{(k)}, \quad k \in \{1, \ldots, K\},\ n \in \{1, \ldots, N\} \tag{1.24}$$

where $D_{n,n}^{(k)}$ is the (n, n)-th component of the k-th diagonal matrix $\boldsymbol{D}^{(k)}$. Let $\boldsymbol{d}_{n_1}$ and $\boldsymbol{d}_{n_2}$ denote the n_1-th and n_2-th column vectors of $\boldsymbol{D}$, respectively. The cosine of angle ψ_{n_1,n_2} between two vectors $\boldsymbol{d}_{n_1}$ and $\boldsymbol{d}_{n_2}$ can be derived by using the following Euclidean dot product formula:

$$\boldsymbol{d}_{n_1}^{\mathsf{T}} \boldsymbol{d}_{n_2} = \|\boldsymbol{d}_{n_1}\| \|\boldsymbol{d}_{n_2}\| \cos(\psi_{n_1,n_2}) \tag{1.25}$$

Definition 1.9. *The coherence ρ of K diagonal matrices $\boldsymbol{D}^{(k)}$ is defined as the maximum absolute cosine of angle ψ_{n_1,n_2} between the column vectors of $\boldsymbol{D}$ as follows:*

$$\rho = \max_{\substack{n_1,n_2 \\ n_1 \neq n_2}} |\cos(\psi_{n_1,n_2})| \ \textit{with}\ \cos(\psi_{n_1,n_2}) = \frac{\boldsymbol{d}_{n_1}^{\mathsf{T}} \boldsymbol{d}_{n_2}}{\|\boldsymbol{d}_{n_1}\| \|\boldsymbol{d}_{n_2}\|} \tag{1.26}$$

If one of the two vectors is zero, ρ is defined to be one by convention.

The coherence ρ, also known as the modulus of uniqueness, captures the uniqueness property of the JDC problem as described in the following theorem [Afsari, 2008].

Theorem 1.2. *Let each $\boldsymbol{C}^{(k)}$ satisfy equation* (1.23)*. The necessary and sufficient condition to identify a unique nonorthogonal joint transformation matrix $\boldsymbol{A}$ up to scale and permutation indeterminacies is $\rho < 1$.*

The uniqueness issue of the complex-valued JDC problem was addressed in [Kleinsteuber and Shen, 2013]. Based on theorem 1.2 and the properties of cumulants, Afsari provided an identifiability condition for the JDC-based ICA method [Afsari, 2008].

Theorem 1.3. *(Identifiability of ICA - JDC formulation) Regarding the noiseless ICA model, let us assume that i) the covariance matrix $\boldsymbol{R}_{\boldsymbol{s}}$ of the source vector $\boldsymbol{s}$ is nonsingular, ii) the q-th order marginal cumulants (for some $q > 2$) of all the source components exist and iii) at most one of them is zero. Then an exact JDC of the N^{q-2} cumulant matrix slices $\boldsymbol{C}^{(q)}_{\boldsymbol{x},:,:,n_3,\ldots,n_q}$ as well as the covariance matrix $\boldsymbol{R}_{\boldsymbol{x}}$, results in finding $\boldsymbol{A}$ up to column scaling and permutation. For a source vector with finite cumulants of all orders, this process fails to identify $\boldsymbol{A}$ if and only if the source vector has more than one Gaussian component.*

Note that the use of the covariance matrix is optional when all the source components have non-zero q-th order cumulants.

1.3.3 JDC algorithms

The JDC problem has been widely studied during the past two decades and has been mostly handled as an optimization problem. The algorithms mainly depend on the criterion chosen to perform the optimization. These criteria include the indirect-fit criterion, the direct-fit criterion, the combination of the direct and indirect-fit criteria and the approximate log-likelihood criterion. Following the recent survey of Chabriel et al. [Chabriel et al., 2014], we provide a comprehensive overview of these four groups of criteria and the associated JDC algorithms. In the rest of this section, when a JDC problem is mentioned without further specification, it refers to a "square" problem, in which the matrices $\boldsymbol{A}$, $\boldsymbol{D}^{(k)}$ and $\boldsymbol{C}^{(k)}$ are of size $(N \times N)$.

1.3.3.1 Minimizing the indirect-fit criterion

Based on the JDC model (1.23), the matrices $\boldsymbol{D}^{(k)}$ can be expressed as a function of $\boldsymbol{A}$ by $\boldsymbol{D}^{(k)} = \boldsymbol{A}^{-1}\boldsymbol{C}^{(k)}(\boldsymbol{A}^{-1})^{\mathsf{T}}$ and should be diagonal by definition. A natural way to compute $\boldsymbol{A}$ is to minimize the squares of the off-diagonal components of $\boldsymbol{D}^{(k)}$ with respect to $\boldsymbol{A}$, leading to the following indirect-fit criterion:

$$\Psi_1(\boldsymbol{A}) = \sum_{k=1}^{K} \left\| \operatorname{off}(\boldsymbol{A}^{-1}\boldsymbol{C}^{(k)}\boldsymbol{A}^{-\mathsf{T}}) \right\|_F^2 \tag{1.27}$$

where $\operatorname{off}(\cdot)$ sets the diagonal entries of the matrix argument to zero, and where $\boldsymbol{A}^{-\mathsf{T}}$ denotes $(\boldsymbol{A}^{-1})^{\mathsf{T}}$. Frequently the minimization of criterion (1.27) is performed with respect to a matrix $\boldsymbol{V}$ such that $\boldsymbol{V} = \boldsymbol{A}^{-1}$, instead of $\boldsymbol{A}$ for simplicity, and $\boldsymbol{V}$ is called the joint diagonalizer. In the context of ICA, $\boldsymbol{V}$ can be regarded as a demixing matrix. To use this criterion the matrix $\boldsymbol{V}$ (or $\boldsymbol{A}^{-1}$) should be properly constrained in order to avoid the trivial zero solution and/or degenerate solutions. In the following, we present the options of constraints and penalty terms appeared in literature:

1. $\boldsymbol{V}$ is unitary. Cardoso and Souloumiac proposed an efficient algorithm, namely Joint Approximate Diagonalization (JAD), which estimated $\boldsymbol{V}$ by successive Jacobi iterations [Cardoso and Souloumiac, 1996]. Moreau extended this algorithm by introducing a generalized JDC criterion [Moreau, 2001]. Zhang et al. estimated $\boldsymbol{V}$ by sequential Householder transforms [Zhang et al., 2009].

2. Each column vector of $\boldsymbol{V}$ has unit norm. This constraint is less restrictive than the former one and was adopted in [Wang et al., 2006, Dégerine and Kane, 2007, Fadaili et al., 2007].

3. The diagonal elements of the matrix $\boldsymbol{V}\boldsymbol{C}^{(k_0)}\boldsymbol{V}^{\mathsf{T}}$ must be equal to one, where $\boldsymbol{C}^{(k_0)}$ with $k_0 \in \{1, \ldots, K\}$ is a positive definite matrix in $\{\boldsymbol{C}^{(k)}\}$. In ICA, $\boldsymbol{C}^{(k_0)}$ could be the zero-delayed covariance matrix of the observations, and this constraint turns out to force the separated sources having unit variance. Vollgraf and Obermayer proposed a Quadratic DIAGonalization algorithm (QDIAG), which estimated each row vector of $\boldsymbol{V}$ alternately by matrix eigenvector computation [Vollgraf and Obermayer, 2006].

4. $\boldsymbol{V}$ has a unit determinant. This constraint can be guaranteed by expressing $\boldsymbol{V}$ as a product of a sequence of elementary unit-determinant matrices, such as Givens rotation matrices, hyperbolic rotation matrices and elementary triangular matrices. A merit of using these elementary matrices is that each matrix is determined by only one parameter, which generally can be estimated algebraically by means of a Jacobi-like optimization procedure. These methods include the LUJ1D and QRJ1D algorithms proposed by Afsari [Afsari, 2006], the J-DI algorithm proposed by Souloumiac [Souloumiac, 2009], the SL algorithm proposed by Sørensen et al. [Sørensen et al., 2009], and the coupled LU and QR factorizations based algorithms proposed by Maurandi et al. [Maurandi et al., 2013] and by Maurandi and Moreau [Maurandi and Moreau, 2014b], respectively. Several methods extended the ideas for complex-valued JDC problems [Guo et al., 2010, Gong et al., 2012, Mesloub et al., 2014, Maurandi and Moreau, 2014a, Maurandi et al., 2014], to cite a few.

5. $\boldsymbol{V}$ is constrained to be invertible. Ziehe et al. proposed an iterative method, namely Fast Frobenius DIAGonalization (FFDIAG) [Ziehe et al., 2004]. Given a nonsingular initialization $\boldsymbol{V}^{(0)}$, in the it-th iteration, $\boldsymbol{V}^{(it-1)}$ is updated by a strictly diagonally dominant matrix $\boldsymbol{U}$ (i.e. for every row of $\boldsymbol{U}$, the magnitude of the diagonal entry in a row is larger than the sum of the magnitudes of all the other non-diagonal entries in that row), therefore the product $\boldsymbol{V}^{(it)} = \boldsymbol{U}\boldsymbol{V}^{(it-1)}$ is guaranteed to be invertible. A complex-valued extension of FFDIAG was presented in [Xu et al., 2011]. Yeredor et al. derived a relative gradient approach and pointed out that for small enough step-sizes the invertibility of $\boldsymbol{V}$ was ensured [Yeredor et al., 2004]. Thereafter, Ghennioui et al. improved the former algorithm based on computing an optimal step-size [Ghennioui et al., 2009]. Trainini et al. extended the relative gradient algorithm for complex-valued matrices [Trainini et al., 2010]. Several variations of gradient-like methods were provided by Trainini and Moreau [Trainini and Moreau, 2012].

6. Several methods resorted to add penalty terms to criterion (1.27). In [Wang et al., 2005], Wang et al. introduced two penalty terms, namely $\|\boldsymbol{V}\boldsymbol{V}^{\mathsf{T}} - \boldsymbol{I}_N\|_F^2$ and $\|\operatorname{diag}(\boldsymbol{V} - \boldsymbol{I}_N)\|_F^2$, and proposed a gradient descent method, where diag(.) returns a matrix comprising only the diagonal elements of the input matrix. Li and Zhang utilized $\log|\det(\boldsymbol{V})|$ and performed the minimization with respect to each column vector of $\boldsymbol{V}$ alternately [Li and Zhang, 2007]. For a non-square matrix $\boldsymbol{V}$, Li et al. turned to use $\log|\det(\boldsymbol{V}\boldsymbol{V}^{\mathsf{T}})|$ and proposed a gradient decent algorithm [Li et al., 2007]. This penalty term was also adopted by Zhou et al. in [Zhou et al., 2009]. Pham and Congedo proposed a new intrinsic constraint $\sum_k(\boldsymbol{V}_{n,:}\boldsymbol{C}^{(k)}\boldsymbol{V}_{n,:}^{\mathsf{T}})^2 = 1$, and derived a pseudo Newton algorithm, where $\boldsymbol{V}_{n,:}$ is the n-th row vector of $\boldsymbol{V}$ [Pham and Congedo, 2009].

Besides the criterion $\Psi_1(\boldsymbol{A})$ in (1.27), Afsari proposed another indirect-fit criterion suitable for JDC, which was scale-invariant in $\boldsymbol{A}$ (and $\boldsymbol{V}$) [Afsari, 2006, Afsari, 2008]:

$$\Psi_2(\boldsymbol{A}) = \sum_{k=1}^{K} \left\| \boldsymbol{C}^{(k)} - \boldsymbol{A}\operatorname{diag}(\boldsymbol{A}^{-1}\boldsymbol{C}^{(k)}\boldsymbol{A}^{-\mathsf{T}})\boldsymbol{A}^{\mathsf{T}} \right\|_F^2 \tag{1.28}$$

The scale invariance means that the change of the scale of any columns of $\boldsymbol{A}$ does not affect the value of this criterion. Based on criterion (1.28), Afsari proposed two algorithms, namely LUJ2D and QRJ2D, by means of LU and QR matrix factorizations of $\boldsymbol{V}$, respectively [Afsari, 2006]. Recently, Yeredor et al. modified this criterion and used it in a different JDC problem, named sequentially drilled JDC [Yeredor et al., 2012].

1.3.3.2 Minimizing the direct-fit criterion

The direct-fit criterion is a measure of the squared difference between the given matrices $\boldsymbol{C}^{(k)}$ and their assumed model parameters in terms of estimating the joint transformation matrix $\boldsymbol{A}$ and K diagonal matrices $\boldsymbol{D}^{(k)}$ ($k \in \{1, \ldots, K\}$),

$$\Psi_3(\boldsymbol{A}, \{\boldsymbol{D}^{(k)}\}) = \sum_{k=1}^{K} \left\| \boldsymbol{C}^{(k)} - \boldsymbol{A}\boldsymbol{D}^{(k)}\boldsymbol{A}^{\mathsf{T}} \right\|_F^2 \tag{1.29}$$

This criterion was originally introduced by Wax and Sheinvald [Wax and Sheinvald, 1997]. They revealed that criterion (1.29) coincided with criterion (1.27) when $\boldsymbol{A}$ was constrained to be unitary. For a non-unitary matrix $\boldsymbol{A}$, Van der Veen proposed to minimize (1.29) via subspace fitting technique [Van der Veen, 2001]. An effective optimization scheme was proposed by Yeredor [Yeredor, 2002]. The minimization of (1.29) was performed alternately between each column vector of $\boldsymbol{A}$ and the diagonal matrices $\boldsymbol{D}^{(k)}$, giving birth to the well-known Alternating Columns and Diagonal Center (ACDC) algorithm. Recently, Zhang and Lou developed a recursive algorithm, which provided a compromise between performance and numerical complexity [Zhang and Lou, 2013]. Zeng and Feng proposed a novel algorithm based on the hybrid trust region method [Zeng and Feng, 2014]. In addition, criterion (1.29) does not require $\boldsymbol{A}$ to be square, and it can even handle the underdetermined case where $\boldsymbol{A}$ has more columns than rows [De Lathauwer and Castaing, 2008].

On the other hand, in the absence of noise, $\boldsymbol{A}$ can be extracted from the Generalized EigenValue Decomposition (GEVD) of two matrices $\boldsymbol{C}^{(1)}$ and $\boldsymbol{C}^{(2)}$ as follows:

$$\boldsymbol{C}^{(1)}(\boldsymbol{C}^{(2)})^{-1} = \boldsymbol{A}(\boldsymbol{D}^{(1)}(\boldsymbol{D}^{(2)})^{-1})\boldsymbol{A}^{-1} \tag{1.30}$$

where we assume that $\boldsymbol{D}^{(2)}$ (and hence $\boldsymbol{C}^{(2)}$) is nonsingular. When a noisy JDC problem involving more than two matrices is considered, Yeredor provided a suboptimal but closed-form (non-iterative) solution by deriving two "representative" matrices from $\{\boldsymbol{C}^{(k)}\}$ and computing the GEVD [Yeredor, 2005]. The algorithm is called Exact Joint Diagonalization (EJD). Based on EJD, Chabriel and Barrère proposed a more robust method, namely DIagonalization using Equivalent Matrices (DIEM) [Chabriel and Barrère, 2012]. DIEM requires that at least $K = N$ matrices $\boldsymbol{C}^{(k)}$ are available.

Additionally, each matrix $\boldsymbol{C}^{(k)} \in \mathbb{R}^{N\times N}$ can be regarded as a fontal matrix slice of a TO array $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{N\times N\times K}$. The minimization of criterion (1.29) is directly linked to the CPD of $\boldsymbol{\mathcal{C}}$, such that $\boldsymbol{\mathcal{C}} = [\![\boldsymbol{A}, \boldsymbol{A}, \boldsymbol{D}]\!]$, where $\boldsymbol{D}$ is defined in equation (1.24). As previously reviewed in section 1.2.3.1, this special case of CPD when two loading matrices coincide is called INDSCAL decomposition [Carroll and Chang, 1970]. Comon and Rajih proposed a method for INDSCAL decomposition in [Comon and Rajih, 2006, Appendix 7.2]. Albera and Karfoul derived a different Alternating Least Squares (ALS) algorithm (ALS) incorporating an Enhanced Line Search (ELS) procedure [Albera and Karfoul, 2009]. Trainini and Moreau proposed another ELS-ALS method for complex-valued matrices [Trainini and Moreau, 2011]. A damped Gaussian-Newton algorithm was provided by Koldovský et al. [Koldovský et al., 2011]. Moreover, since INDSCAL model is a special case of the CPD, one can attempt to decompose $\boldsymbol{\mathcal{C}}$ by general CPD $\boldsymbol{\mathcal{C}} = [\![\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{D}]\!]$, and suppose that eventually the two loading matrices $\boldsymbol{A}^{(1)}$ and $\boldsymbol{A}^{(2)}$ will converge to an estimate of $\boldsymbol{A}$. But in practice, some differences will remain between $\boldsymbol{A}^{(1)}$ and $\boldsymbol{A}^{(2)}$. Then the n-th column vector of $\boldsymbol{A}$, denoted by $\boldsymbol{a}_n$, can be estimated as the dominant left singular vector of the matrix $[\boldsymbol{a}_n^{(1)}\boldsymbol{a}_n^{(2)}] \in \mathbb{R}^{N\times 2}$, where $\boldsymbol{a}_n^{(1)}$ and $\boldsymbol{a}_n^{(2)}$ are the n-th column vectors of $\boldsymbol{A}^{(1)}$ and $\boldsymbol{A}^{(2)}$, respectively, and $n \in \{1, \ldots, N\}$ [De Lathauwer and Castaing, 2008]. An explicit link between the CP and JDC decompositions was established by De Lathauwer in [De Lathauwer, 2006].

1.3.3.3 Combination of the direct and indirect-fit criteria

In criterion (1.29), instead of fitting the set of matrices $\boldsymbol{C}^{(k)}$ by $\boldsymbol{A}$ and $\boldsymbol{D}^{(k)}$ directly, we can consider fitting a set of partially diagonalized matrices $\boldsymbol{V}^{(it)}\boldsymbol{C}^{(k)}(\boldsymbol{V}^{(it)})^{\mathsf{T}}$ at the it-th iteration by means of a residual joint transformation $\bar{\boldsymbol{A}}^{(it)}$ and K diagonal matrices $\boldsymbol{D}_v^{(k)}$, using the following criterion:

$$\Psi_4(\bar{\boldsymbol{A}}^{(it)}) = \sum_{k=1}^{K} \left\| \boldsymbol{V}^{(it)}\boldsymbol{C}^{(k)}(\boldsymbol{V}^{(it)})^{\mathsf{T}} - \bar{\boldsymbol{A}}^{(it)}\boldsymbol{D}_v^{(k)}(\bar{\boldsymbol{A}}^{(it)})^{\mathsf{T}} \right\|_F^2 \tag{1.31}$$

where it denotes the iteration index, where $\boldsymbol{V}^{(it)}$ is the obtained joint diagonalizer at the it-th iteration, and where $\boldsymbol{D}_v^{(k)} = \mathrm{diag}\{\boldsymbol{V}^{(it)}\boldsymbol{C}^{(k)}(\boldsymbol{V}^{(it)})^{\mathsf{T}}\}$. Criterion (1.31) can be regarded as the combination of the direct criterion (1.27) and the indirect-fit criterion (1.29). Each minimization step consists in one iteration of a direct-fit procedure with respect to $\bar{\boldsymbol{A}}^{(it)}$. In the direct-fit procedure, a Gaussian-Newton method is adopted for the purpose of fast convergence in a neighborhood of the true local minimum which is close to $\bar{\boldsymbol{A}}^{(it)} = \boldsymbol{I}_N$. By setting $\boldsymbol{I}_N$ as the initial point, only one iteration of the Gaussian-Newton optimization is applied for each step, because at such an initial point, the inverse of the Hessian matrix can be computed by solving distinct sets of (2×2) linear equations [Chabriel et al., 2014]. Once the residual joint transformation matrix $\bar{\boldsymbol{A}}^{(it)}$ is found, the joint diagonalizer $\boldsymbol{V}^{(it+1)}$ is computed by $\boldsymbol{V}^{(it+1)} = (\bar{\boldsymbol{A}}^{(it)})^{-1}\boldsymbol{V}^{(it)}$. This method was proposed by Tichavský and Yeredor, and called Uniformly Weighted Exhaustive Diagonalization with Gauss itErations (UWEDGE) [Tichavský and Yeredor, 2009].

1.3.3.4 Minimizing the approximate log-likelihood criterion

As already discussed in section 1.3.1, when the ICA model assumes statistically independent sources with variances that are constant within each block but varying between blocks, the jointly diagonalizable matrices $\boldsymbol{C}^{(k)}$ can be sample covariance matrices taken from distinct blocks of the observations [Pham and Cardoso, 2001]. Consequently, each $\boldsymbol{C}^{(k)}$ is a positive definite matrix. Pham utilized the following criterion, which measured the diagonality of positive definite matrices [Pham, 2001b]:

$$\Psi_5(\boldsymbol{A}) = \sum_{k=1}^{K} \log \frac{\det\{\mathrm{diag}(\boldsymbol{A}^{-1}\boldsymbol{C}^{(k)}\boldsymbol{A}^{-\mathsf{T}})\}}{\det\{\boldsymbol{A}^{-1}\boldsymbol{C}^{(k)}\boldsymbol{A}^{-\mathsf{T}}\}} \tag{1.32}$$

Minimizing (1.32) is approximately equivalent to maximizing a log-likelihood objective function for JDC. Historically, this criterion was first studied by Flury and Gautsch for seeking a unitary joint transformation matrix $\boldsymbol{A}$ [Flury and Gautsch, 1986]. In this case, the denominators in (1.32) are constant. Flury and Gautsch estimated $\boldsymbol{A}$ by successive Jacobi iterations. Pham considered the more general case that $\boldsymbol{A}$ was non-unitary. He proposed to maximize a lower bound of the decrease of (1.32) by means of sequential (2×2) transformations, and derived algebraic solutions for those transformation matrices [Pham, 2001b]. Thereafter, Joho provided a Newton algorithm [Joho, 2008]. Todros and Tabrikian proposed a computationally efficient algorithm by minimizing an approximation of criterion (1.32) [Todros and Tabrikian, 2010].

In many applications, the performance of the JDC algorithms can be substantially improved by introducing appropriate weighting in the optimization criterion. For example, the indirect-fit criterion (1.27) can be rewritten equivalently as follows:

$$\Psi_1(\boldsymbol{A}) = [\mathrm{f}(\boldsymbol{A})]^{\mathsf{T}}\mathrm{f}(\boldsymbol{A}) \tag{1.33}$$

where the function $\mathrm{f}(\boldsymbol{A})$ returns a $KN(N-1)/2$-dimensional column vector composed of all off-diagonal components below the main diagonal of the K matrices $\boldsymbol{A}^{-1}\boldsymbol{C}^{(k)}\boldsymbol{A}^{-\mathsf{T}}$. The weighted indirect-fit criterion can be derived by inserting a positive definite weight matrix $\boldsymbol{\Upsilon}$ in (1.33) as follows [Tichavský and Yeredor, 2009]:

$$\Psi_{w1}(\boldsymbol{A}) = [\mathrm{f}(\boldsymbol{A})]^{\mathsf{T}}\boldsymbol{\Upsilon}\,\mathrm{f}(\boldsymbol{A}) \tag{1.34}$$

A weighed direct-fit criterion can be found in [Yeredor, 2000]. In practice, it could be difficult to obtain an optimal weight matrix $\boldsymbol{\Upsilon}$ since the statistical characterization of the source components is not fully available. Comprehensive discussion on choosing the best weighting matrix for each criterion according to different statistical models of the sources is out of the scope of this thesis report. For more details, one may refer to the following references [Yeredor, 2000, Tichavský and Yeredor, 2009, Slapak and Yeredor, 2011, Yeredor, 2012].

1.4 Nonnegative matrix factorization

In the previous section we have addressed the ICA problem. The effectiveness of ICA relies on the statistical independence constraint of the source components. However, in ICA no constraint is made on the sign of the components either of the mixing matrix or of the sources. In other words, the negative values are allowed in the data representation model. Nevertheless, the negative values of both observations and source components are physically meaningless in a wide variety of real-world data. For example, the color and intensity of a pixel in an image is given by a nonnegative number; the frequency of a word that appears in a document is assigned by a nonnegative value; and a music note in a piece of music contributes a nonnegative amount to the signal power spectrum. In addition, some observed signal is most naturally characterized by the exclusively additive combination of the source components, for instance, the eye, nose and mouth of a face image, leading to a nonnegative mixing matrix as well. Imposing the nonnegativity constraint in some BSS problems allows us to obtain results in agreement with physical reality and to facilitate easier interpretation. However, we often lose the nonnegativity constraint during the ICA procedure, for example, when we perform whitening by means of PCA [Plumbley et al., 2010].

In this section, we discuss a specific method for the use of nonnegativity constraints in BSS problems, namely Nonnegative Matrix Factorization (NMF). Let us rewrite the linear BSS model (1.1) in the following matrix form:

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S} + \boldsymbol{E} \tag{1.35}$$

where $\boldsymbol{X} \in \mathbb{R}^{N\times M}$ is an observation matrix whose (n,m)-th component denotes the m-th realization of the n-th component of the observation vector $\boldsymbol{x}$ (1.1), where $\boldsymbol{A} \in \mathbb{R}^{N\times P}$ continues to express the mixing matrix, where $\boldsymbol{S} \in \mathbb{R}^{P\times M}$ is a source matrix whose

(p, m)-th component gives the value for the corresponding realization of the p-th source, and where the noise matrix $\boldsymbol{E} \in \mathbb{R}^{N \times M}$ can be similarly defined. We are interested in the conditions where both the mixing matrix $\boldsymbol{A}$ and the source matrix $\boldsymbol{S}$ have nonnegative components. Therefore, we can define the following NMF problem:

Problem 1.3. *Given a nonnegative observation matrix $\boldsymbol{X} \in \mathbb{R}_{+}^{N \times M}$, find a mixing matrix $\boldsymbol{A} \in \mathbb{R}^{N \times P}$ and a source matrix $\boldsymbol{S} \in \mathbb{R}^{P \times M}$, such that:*

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S} + \boldsymbol{E} \tag{1.36}$$

by minimizing the residual term $\boldsymbol{E}$, subject to $\boldsymbol{A}$ and $\boldsymbol{S}$ having nonnegative components.

NMF was initially investigated by Paatero and Tapper as the concept of positive matrix factorization [Paatero and Tapper, 1994, Paatero, 1997], and gained popularity through the well-known works of Lee and Seung [Lee and Seung, 1999]. Various methodologies and extensions of NMF have been constantly emerging during the last few years. The existing NMF algorithms can be divided into the following four categories, including the basic NMF, the constrained NMF, the structured NMF and the generalized NMF [Wang and Zhang, 2013]. NMF and its various extensions have become prominent tools in a wide range of areas, such as image classification [Das Gupta and Xiao, 2011], face recognition [Zafeiriou et al., 2006], music transcription [Févotte et al., 2009], spectroscopy separation [Van Benthem and Keenan, 2004], and pattern recognition [Cichocki et al., 2009]. Many systematic surveys of NMF have been addressed in literature, such as the review papers [Tropp, 2003, Chu et al., 2004, Berry et al., 2007, Buciu, 2008, Plumbley et al., 2010, Wang and Zhang, 2013, Zhou et al., 2014, Smaragdis et al., 2014], and an in-depth book [Cichocki et al., 2009].

In this section, we provide a concise overview of limited areas of NMF which are more relevant to the work described in this thesis, including basic NMF and one generalized NMF, namely semi-NMF. For a more broad discussion one may refer to the aforementioned literature.

1.4.1 Existence and uniqueness

Given a nonnegative matrix $\boldsymbol{X}$, its trivial NMF solution of form $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S}$ always exists, for instance $\boldsymbol{A} = \boldsymbol{X}$ and $\boldsymbol{S} = \boldsymbol{I}$. Vasiloglou et al. proved that every nonnegative matrix has a non-trivial NMF solution by relating NMF to the Completely Positive Matrix (CPM) factorization [Vasiloglou et al., 2009]. It shows that the set of CPMs forms a convex cone and the solution of NMF belongs to the CPM cone, leading to the conclusion that solving NMF is a convex optimization problem. However, finding a practical description of the CPM cone is still an open question. So although the problem could be reformulated as a convex problem, there is no algorithm known for solving it [Vasiloglou et al., 2009].

NMF is considered to be essentially unique when its solution is only subject to scale and permutation indeterminacies. Unfortunately, NMF solutions often suffer from additional rotation indeterminacy. For example, there are many ways to select a ratation matrix $\boldsymbol{R}$ which is not necessarily nonnegative or not necessarily equal to a product of a diagonal matrix and a permutation matrix, so that

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S} = \boldsymbol{A}\boldsymbol{R}\boldsymbol{R}^{-1}\boldsymbol{S} = \boldsymbol{A}_1\boldsymbol{S}_1 \tag{1.37}$$

and the rotated matrices $\boldsymbol{A}_1 = \boldsymbol{AR} \neq \boldsymbol{A}$ and $\boldsymbol{S}_1 = \boldsymbol{R}^{-1}\boldsymbol{S} \neq \boldsymbol{S}$ are still nonnegative [Cichocki et al., 2009]. That is to say, only the nonnegativity constraint is not sufficient to guarantee the uniqueness. Nevertheless, incorporating some extra constraints, such as separability, sparsity or symmetry, is sufficient to solve the NMF problem uniquely. The first sufficient condition of uniqueness for separable NMF was provided by Donoho and Stodden [Donoho and Stodden, 2003]. Laurberg et al. presented a stronger necessary and sufficient condition for the uniqueness based on the boundary closeness assumption of factors [Laurberg et al., 2008]. Gillis analyzed the uniqueness of sparse NMF from a geometric point of view [Gillis, 2012]. More recently Huang et al. derived new uniqueness results for symmetric and asymmetric NMF, which stated that a sufficient condition for uniqueness was that the conic hull of the latent factors was a superset of a particular second-order cone [Huang et al., 2014]. In addition checking this condition was shown to be Non-deterministic Polynomial time (NP)-complete. We omit these mathematical details in this report.

1.4.2 Basic NMF algorithms

The idea of basic NMF is to find efficient solutions to the NMF problem under the sole nonnegativity constraint. Due to the NP-hardness and lack of appropriate convex formulation of the NMF problem, the non-convex formulations are adopted in practice. Generally the non-convex formulations are relatively easy to solve but usually only local minima are achievable. Practical NMF algorithms perform the optimization by minimizing proper cost functions that measure the distance or discrepancy between the given matrix $\boldsymbol{X}$ and the factorization $\boldsymbol{AS}$. The optimization schemes differ from each other in the cost functions and the optimization procedures.

1.4.2.1 Cost functions

The Square of Euclidean Distance (SED) in equation (1.38) and the Generalized Kullback-Leibler Divergence (GKLD) in equation (1.39) are the most commonly used cost functions [Lee and Seung, 2000]:

$$\Psi_{\mathrm{F}}(\boldsymbol{A}, \boldsymbol{S}) = \| \boldsymbol{X} - \boldsymbol{AS} \|_F^2 = \sum_{n=1}^{N} \sum_{m=1}^{M} (X_{n,m} - [AS]_{n,m})^2 \tag{1.38}$$

$$\Psi_{\mathrm{KL}}(\boldsymbol{A}, \boldsymbol{S}) = \sum_{n=1}^{N} \sum_{m=1}^{M} (X_{n,m} \log \frac{X_{n,m}}{[AS]_{n,m}} - X_{n,m} + [AS]_{n,m}) \tag{1.39}$$

where $X_{n,m}$ is the (n,m)-th component of $\boldsymbol{X}$, and where $[AS]_{n,m}$ denotes the (n,m)-th component of the matrix product $\boldsymbol{AS}$. Other cost functions include Itakura-Saito Divergence (ISD) [Févotte et al., 2009], α-divergence [Cichocki et al., 2008], β-divergence [Kompass, 2007], γ-divergence [Cichocki and Amari, 2010], Earth mover's distance [Sandler and Lindenbaum, 2011] and many others. Different cost functions can be chosen according to the prior information of the probability distribution of the noises. For example, the minimizations of the SED, GKLD and ISD are equivalent to the maximum likelihood estimators under Gaussian additive, Poisson and multiplicative Gamma noises, respectively [Févotte and Idier, 2011]. Although generally these cost functions are not jointly

convex in both $\boldsymbol{A}$ and $\boldsymbol{S}$, they are separately convex in either $\boldsymbol{A}$ or $\boldsymbol{S}$, giving birth to a wide range of NMF algorithms based on an alternating minimization scheme.

1.4.2.2 Algorithms

Lee and Seung introduced the SED and GKLD cost functions, and applied Multiplicative Updates (MU) iteratively between $\boldsymbol{A}$ and $\boldsymbol{S}$ [Lee and Seung, 1999]. For example, the MU rule for SED is shown as follows:

$$\boldsymbol{A}^{(it+1)} = \boldsymbol{A}^{(it)} \boxdot [\boldsymbol{X}(\boldsymbol{S}^{(it)})^{\mathsf{T}}] \oslash [\boldsymbol{A}^{(it)}\boldsymbol{S}^{(it)}(\boldsymbol{S}^{(it)})^{\mathsf{T}}] \tag{1.40}$$

$$\boldsymbol{S}^{(it+1)} = \boldsymbol{S}^{(it)} \boxdot [(\boldsymbol{A}^{(it+1)})^{\mathsf{T}}\boldsymbol{X}] \oslash [(\boldsymbol{A}^{(it+1)})^{\mathsf{T}}\boldsymbol{A}^{(it+1)}\boldsymbol{S}^{(it)}] \tag{1.41}$$

where it is the iteration index, where $\boxdot$ denotes Hadamard product, and where $\oslash$ means matrix element-wise division. The MU rules can be seen as adaptive rescaled gradient descent algorithms and they have been widely used as a baseline due to their simplicity. Their multiplicative natures guarantee that the resulting matrix components cannot become negative. Berry et al. analyzed the convergence of the MU algorithms of Lee and Seung [Berry et al., 2007].

Considering the separate convexity of the SED cost function (1.38), the joint optimization can be converted into an alternate minimization of (1.38) with respect to $\boldsymbol{A}$ or $\boldsymbol{B}$ while fixing the other one in the least-square sense under the nonnegativity constraint. Such an alternate optimization scheme is referred to as Alternating Nonnegative Least Squares (ANLS) [Kim et al., 2014]. The following two-step algorithm is the simplest way to achieve ANLS [Berry et al., 2007]:

$$\boldsymbol{A}^{(it+1)} = \mathrm{P}_+\{\, \boldsymbol{X}(\boldsymbol{S}^{(it)})^{\mathsf{T}}[\boldsymbol{S}^{(it)}(\boldsymbol{S}^{(it)})^{\mathsf{T}}]^{-1} \,\} \tag{1.42}$$

$$\boldsymbol{S}^{(it+1)} = \mathrm{P}_+\{\, [(\boldsymbol{A}^{(it+1)})^{\mathsf{T}}\boldsymbol{A}^{(it+1)}]^{-1}(\boldsymbol{A}^{(it+1)})^{\mathsf{T}}\boldsymbol{X} \,\} \tag{1.43}$$

where $\mathrm{P}_+\{\cdot\}$ sets all negative values of the input matrix to zero. Kim et al. claimed that the projection of negative values onto the positive orthant can not guarantee the decrease of the cost function [Kim et al., 2007]. In practice, this procedure was reported to have good performance [Berry et al., 2007]. Rather than using the projection operation, the Nonnegativity constrained Least-Square (NLS) problem can be solved by means of the active set algorithm proposed by Lawson and Hanson [Lawson and Hanson, 1974]. Fast NLS algorithms were derived by Bro and De Jong [Bro and De Jong, 1997] and by Van Benthem and Keenan [Van Benthem and Keenan, 2004], respectively. By adopting the active set based NLS algorithm, Kim and Park proposed a framework of ANLS methods for NMF [Kim and Park, 2008].

In order to improve the performance, as well as to accelerate the convergence rate, many NMF methods were proposed based on sophisticated optimization schemes. Merritt and Zhang proposed an interior-point gradient method for solving the NLS problem [Merritt and Zhang, 2005]. Gonzalez and Zhang introduced a multiplicative regulatory factor leading to an accelerated MU algorithm [Gonzalez and Zhang, 2005]. Chu et al. derived a family of Newton-type approaches for NMF, based on a projected Newton method, the use of an alternating direction iterative Newton method and sequential quadratic programming, respectively [Chu et al., 2004]. Moreover, they introduced a way to parameterize a nonnegative matrix $\boldsymbol{A} \in \mathbb{R}_+^{N\times P}$ over the open set $\mathbb{R}^{N\times P}$ by means of a square change of variable: $\boldsymbol{A} = \boldsymbol{B}\boxdot\boldsymbol{B}$, where $\boldsymbol{B} \in \mathbb{R}^{N\times P}$ has unconstrained components.

This differentiable parametrization effectively transforms the nonnegativity constrained optimization problem into a problem with no constraint at all, leading to an alternating gradient descent method [Chu et al., 2004]. Lin proposed an ANLS algorithm by means of projected gradient methods with an adaptive gradient decent step-size, which achieved a fast convergence rate [Lin, 2007]. In order to obtain a more accurate solution, Zdunek and Cichocki proposed to use the second-order approximations in the Taylor expansion of the cost function, and proposed a family of projected quasi-Newton optimization approaches [Zdunek and Cichocki, 2007]. Li and Zhang developed a fixed-point NMF approach based on an exact block coordinate descent method, namely FastNMF [Li and Zhang, 2009]. Cichocki and Phan proposed a class of Hierarchical Alternating Least Squares (HALS) algorithms, giving birth to FastHALS NMF approaches [Cichocki and Phan, 2009]. It is known that a best rank-one approximation can be achieved in polynomial time. Gillis and Glineur proposed to identify an optimal rank-one NMF solution and subtract it from the input matrix. Then the similar procedure was repeated on the residual matrix under an upper bound constraint [Gillis and Glineur, 2010]. Guan et al. provided a Nesterov's optimal gradient method, in which the step-size was determined by the Lipschitz constant hence avoiding the classical time consuming line search procedure [Guan et al., 2012]. Moreover in order to corp with large-scale problems, many specific optimization approaches were developed, such as an interior-point method [Merritt and Zhang, 2005], local optimization [Cichocki and Phan, 2009], parallel computing [Liu et al., 2010], low-rank approximation [Zhou et al., 2012], Alternating Direction Method of Multipliers (ADMM) [Xu et al., 2012, Sun and Févotte, 2014], to cite only a few.

1.4.3 Semi-NMF algorithms

Basic NMF restricts every component in the observation matrix $\boldsymbol{X}$ to be nonnegative. When the matrix $\boldsymbol{X}$ is unconstrained, which means it may have mixed signs, Ding et al. suggested to restrict $\boldsymbol{A}$ to be nonnegative while placing no restriction on the signs of $\boldsymbol{S}$, leading to the following semi-NMF problem [Ding et al., 2010]:

Problem 1.4. *Given an observation matrix $\boldsymbol{X} \in \mathbb{R}^{N\times M}$, find two matrices $\boldsymbol{A} \in \mathbb{R}^{N\times P}$ and $\boldsymbol{S} \in \mathbb{R}^{P\times M}$, such that*

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S} + \boldsymbol{E} \tag{1.44}$$

by minimizing the residual term $\boldsymbol{E}$, subject to $\boldsymbol{A}$ having nonnegative components.

The motivation of semi-NMF was originated from the perspective of clustering with $\boldsymbol{A}$ denoting the cluster indicators and $\boldsymbol{S}$ representing the cluster centers [Li and Ding, 2006]. In this case, the SED cost function (1.38) can be viewed as an objective function of K-means clustering. On the other side, semi-NMF is also physically interpretable in some practical applications where the observation data is not always nonnegative, so the source components might also have some negative elements reflecting the phase information [Wang and Zhang, 2013].

By using the SED cost function (1.38), the semi-NMF algorithm of Ding et al. [Ding et al., 2010] iterates between finding the optimum $\boldsymbol{S}$ for a given $\boldsymbol{A}$, which is given by the classic least-square solution:

$$\boldsymbol{S}^{(it+1)} = [(\boldsymbol{A}^{(it)})^{\mathsf{T}}\boldsymbol{A}^{(it)}]^{-1}(\boldsymbol{A}^{(it)})^{\mathsf{T}}\boldsymbol{X} \tag{1.45}$$

and updating the estimate of $\boldsymbol{A}$ for a given $\boldsymbol{S}$ by the MU rule:

$$\boldsymbol{A}^{(it+1)} = \boldsymbol{A}^{(it)} \boxdot \sqrt{[\boldsymbol{X}(\boldsymbol{S}^{(it+1)})^{\mathsf{T}}]^{+} + \boldsymbol{A}^{(it)}[\boldsymbol{S}^{(it+1)})(\boldsymbol{S}^{(it+1)})^{\mathsf{T}}]^{-}} \oslash \sqrt{[\boldsymbol{X}(\boldsymbol{S}^{(it+1)})^{\mathsf{T}}]^{-} + \boldsymbol{A}^{(it)}[\boldsymbol{S}^{(it+1)})(\boldsymbol{S}^{(it+1)})^{\mathsf{T}}]^{+}} \tag{1.46}$$

where $\boldsymbol{U}^{+}$ and $\boldsymbol{U}^{-}$ correspond to the positive and negative parts of the input matrix $\boldsymbol{U}$, respectively, given by:

$$U^{+}_{m,n} = (|U_{m,n}| + U_{m,n})/2, \quad U^{-}_{m,n} = (|U_{m,n}| - U_{m,n})/2 \tag{1.47}$$

The convergence of this alternating minimization procedure was also proved. Park and Kim treated semi-NMF as a one-sided NMF problem and solved it with the NLS method [Park and Kim, 2006]. Wang and Li introduced a random projection technique to accelerate the semi-NMF [Wang and Li, 2010]. Kumar et al. proposed a semi-NMF algorithm based on a max-margin classification criterion [Kumar et al., 2011].

Additionally, semi-NMF finds its applications in template matching [Le Roux et al., 2008], image resolution enhancement [Bevilacqua et al., 2012], motion segmentation [Mo and Draper, 2012], graph theory [Mankad and Michailidis, 2013], and hyperspectral image unmixing [Yokoya et al., 2014].

1.5 Nonnegative independent component analysis

In order to achieve uniqueness of the semi-NMF solution, some additional constraints are necessary, such as mutual independence, sparsity or semi-orthogonality [Plumbley et al., 2010, Cichocki, 2013]. The mutual statistical independence, as well as the nonnegativity assumption of the source components leads to the following Nonnegative ICA (NICA) problem:

Problem 1.5. *Given M realizations of an observation vector $\boldsymbol{x} \in \mathbb{R}^N$, find a mixing matrix $\boldsymbol{A} \in \mathbb{R}^{N \times P}$ and the M corresponding realizations of a source vector $\boldsymbol{s} \in \mathbb{R}^P$, such that:*

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{\varepsilon} \tag{1.48}$$

where $\boldsymbol{s}$ has statistically independent and also nonnegative components, and where $\boldsymbol{\varepsilon} \in \mathbb{R}^N$ is an additive noise vector independent of $\boldsymbol{s}$.

1.5.1 Preprocessing and identifiablity

A source component s_i is called nonnegative if $p(s_i < 0) = 0$ and will be called well-grounded, if it has a non-vanishing pdf in any positive neighborhood of zero, such that for any $\delta > 0$ we have $p(s_i < \delta) > 0$ [Plumbley, 2002]. We assume that we have a noiseless observation vector $\boldsymbol{x}$ and the source components s_i have unit variance. Plumbley proposed to perform a whitening step on $\boldsymbol{x}$, yielding $\boldsymbol{z} = \boldsymbol{W}\boldsymbol{x}$, where $\boldsymbol{W} \in \mathbb{R}^{P \times N}$ is the classical whitening matrix. Then the whitened observation vector $\boldsymbol{z}$ can be written as $\boldsymbol{z} = \boldsymbol{U}\boldsymbol{s}$, where $\boldsymbol{U} \in \mathbb{R}^{P \times P}$ is an orthonormal matrix. Therefore, the NICA problem is reduced to finding an orthonormal matrix $\boldsymbol{V} \in \mathbb{R}^{P \times P}$, for which $\widehat{\boldsymbol{s}} = \boldsymbol{V}\boldsymbol{z}$ is equal to $\boldsymbol{s}$ up to scale and permutation indeterminacies.

Plumbley analyzed the identifiability condition for an NICA problem in [Plumbley, 2002]. Firstly, the nonnegativity of an orthonormal matrix is stated in the following lemma:

Lemma 1.1. *Let* $\boldsymbol{U}^{(0)} \in \mathbb{R}^{P\times P}$ *be an orthonormal matrix such that* $\boldsymbol{U}^{(0)}(\boldsymbol{U}^{(0)})^{\mathsf{T}} = (\boldsymbol{U}^{(0)})^{\mathsf{T}}\boldsymbol{U}^{(0)} = \boldsymbol{I}_P$. *Then all the elements of* $\boldsymbol{U}^{(0)}$ *are nonnegative if and only if* $\boldsymbol{U}^{(0)}$ *is a permutation matrix.*

Then based on lemma 1.1, we can obtain the following NICA identifiability condition:

Theorem 1.4. *Let* $\boldsymbol{s} \in \mathbb{R}^P$ *be a random vector of nonnegative and well-grounded independent source components, each with unit variance, and let*

$$\widehat{\boldsymbol{s}} = \boldsymbol{V}\boldsymbol{z} = (\boldsymbol{V}\boldsymbol{U})\boldsymbol{s} = \boldsymbol{U}^{(0)}\boldsymbol{s}$$

be an orthonormal rotation of $\boldsymbol{s}$, *where* $\boldsymbol{U}^{(0)}(\boldsymbol{U}^{(0)})^{\mathsf{T}} = (\boldsymbol{U}^{(0)})^{\mathsf{T}}\boldsymbol{U}^{(0)} = \boldsymbol{I}_P$. *Then* $\boldsymbol{U}^{(0)}$ *is a permutation matrix if and only if* $\widehat{\boldsymbol{s}}$ *is nonnegative with probability* 1.

Many practical algorithms perform optimizations by minimizing a cost function. The following corollary follows:

Corollary 1.1. *For the same conditions as theorem 1.4, suppose that* $\Psi(\boldsymbol{V})$ *is a cost function such that* $\Psi(\boldsymbol{V}) = 0$ *if and only if* $\widehat{\boldsymbol{s}} = \boldsymbol{V}\boldsymbol{z}$ *is nonnegative with probability* 1. *Then* $\Psi(\boldsymbol{V}) = 0$ *if and only if* $\boldsymbol{U}^{(0)} = \boldsymbol{V}\boldsymbol{U}$ *is a permutation matrix, such that* $\widehat{\boldsymbol{s}}$ *is a permutation of the source vector* $\boldsymbol{s}$.

1.5.2 NICA algorithms

In order to derive a suitable cost function, Plumbley suggested to project $\widehat{\boldsymbol{s}}$ onto the positive orthant by $\widehat{\boldsymbol{s}}^{+} = [\widehat{s}_1^{+}, \ldots, \widehat{s}_P^{+}]$ with $\widehat{s}_i^{+} = \max(0, \widehat{s}_i)$, and to construct a reestimate of $\boldsymbol{z} = \boldsymbol{V}^{\mathsf{T}}\widehat{\boldsymbol{s}}$ by $\widehat{\boldsymbol{z}} = \boldsymbol{V}^{\mathsf{T}}\widehat{\boldsymbol{s}}^{+}$ [Plumbley, 2002]. Then the cost function is given as follows:

$$\Psi_{\mathrm{NICA}}(\boldsymbol{V}) = \| \boldsymbol{z} - \widehat{\boldsymbol{z}} \|_F^2 = \left\| \boldsymbol{z} - \boldsymbol{V}^{\mathsf{T}}\widehat{\boldsymbol{s}}^{+} \right\|_F^2 \tag{1.49}$$

The value of (1.49) will be zero if $\boldsymbol{V}$ is obtained such that all the $\widehat{s}_i$ are nonnegative or $\widehat{\boldsymbol{s}} = \widehat{\boldsymbol{s}}^{+}$. Plumbley proposed to minimize criterion (1.49) by means of axis rotations and geodesic search over the Stiefel manifold of orthogonal matrices [Plumbley, 2003]. The convergence of the geodesic NICA algorithm was later proved in [Ye et al., 2006]. Yuan and Oja extended the FastICA algorithm for solving NICA problem [Yuan and Oja, 2004]. In [Plumbley and Oja, 2004], Plumbley and Oja pointed out that the stochastic gradient algorithm for minimizing (1.49) was actually a special case of the nonlinear PCA algorithm which was earlier investigated by Oja [Oja, 1997]. The same authors further showed that in the Stiefel manifold of orthogonal matrices, the cost function had no local minimum and it was a Lyapunov function for the matrix gradien flow. Consequently, they proposed a gradient descent algorithm which monotonically converged to the global minimum [Oja and Plumbley, 2004]. The global convergence of such an algorithm was proved by Ye [Ye, 2006]. In [Plumbley, 2005], Plumbley explored the geometrical methods to tackle the NICA problem, such as the manifold and Lie group of special orthogonal matrices. Winther and Petersen presented an empirical Bayesian framework for NICA

[Winther and Petersen, 2007]. Ouedraogo et al. developed an axis pair rotation method for NICA by means of Newton optimization [Ouedraogo et al., 2010]. A regularized gradient descend algorithm for NICA was presented in [Ouedraogo et al., 2011]. In many practical applications, the nonnegative source components are not well-grounded, Zheng et al. then derived an NICA algorithm minimizing the mutual information [Zheng et al., 2006]. Yu et al. formulated the NICA problem as a maximization problem of kurtosis under the nonnegativity constraint of the estimated sources, and solved it by means of a Lagrange multiplier method [Yu et al., 2013].

Furthermore, NICA plays important roles in image separation [Plumbley, 2003], music transcription [Plumbley, 2003, Dittmar and Uhle, 2004], hyperspectral image demixing [Bakir et al., 2006], gene module identification [Gong et al., 2007], ElectroEncephalography (EEG) / MagnetoEncephaloGraphy (MEG) source imaging [Valdés-Sosa et al., 2009], chemical spectra analysis [Shao et al., 2009], and remote sensing [Yu et al., 2012].

1.6 Semi-nonnegative ICA and nonnegative JDC

1.6.1 The semi-nonnegative ICA problem

In this thesis report, we tackle another important nonnegativity constrained ICA problem. Rather than imposing both nonnegativity and statistical independence on the source vector $\boldsymbol{s}$ as NICA, we propose to exploit the statistical independence of $\boldsymbol{s}$ and the nonnegativity of the mixing matrix $\boldsymbol{A}$, leading to what we call Semi-Nonnegative ICA (SeNICA) [Coloigner, 2012, Coloigner et al., 2014a]. Similarly to semi-NMF, the term *semi* refers to that only the mixing matrix $\boldsymbol{A}$ is constrained to have nonnegative components in the model. More precisely, the SeNICA problem is defined as follows:

Problem 1.6. *Given M realizations of an observation vector $\boldsymbol{x} \in \mathbb{R}^N$, find a mixing matrix $\boldsymbol{A} \in \mathbb{R}^{N \times P}$ and the M corresponding realizations of a source vector $\boldsymbol{s} \in \mathbb{R}^P$, such that:*

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{\varepsilon} \tag{1.50}$$

where $\boldsymbol{A}$ is constrained to have nonnegative components, where $\boldsymbol{s}$ has statistically independent components, and where $\boldsymbol{\varepsilon} \in \mathbb{R}^N$ is an additive noise vector independent of the source vector $\boldsymbol{s}$.

The differences between SeNICA and ICA, NICA, NMF, semi-NMF mainly consist in the assumptions chosen to solve the BSS problem $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{\varepsilon}$, which are summarized in table 1.1. The SeNICA problem is often encountered in BSS applications where only the mixing matrix is considered to be nonnegative. For instance, in Magnetic Resonance Spectroscopy (MRS), the column vectors of $\boldsymbol{A}$ represent the positive concentrations of the spectra of the source chemical components, while the source spectra $\boldsymbol{s}$ may have negative amplitudes, for example in the case of J-modulation [Sajda et al., 2004, Horská and Tkáč, 2012].

Of course, if the SeNICA problem 1.6 is noiseless, and if $\boldsymbol{A}$ and $\boldsymbol{s}$ fully satisfy the conditions listed in section 1.2.1, the identifiability of ICA under these conditions will imply that the explicit nonnegativity constraint on $\boldsymbol{A}$ would be unnecessary. However, in practice, these theoretical conditions can not be completely met due to the model

	NMF	Semi-NMF	ICA	NICA	SeNICA
Nonnegativity of $\boldsymbol{A}$	$\surd$	$\surd$			$\surd$
Nonnegativity of $\boldsymbol{s}$	$\surd$			$\surd$	
Statistical independence of $\boldsymbol{s}$			$\surd$	$\surd$	$\surd$
Well-groundedness of $\boldsymbol{s}$				$\surd$	

Table 1.1: Some different assumptions of NMF, semi-NMF, ICA, NICA and SeNICA for solving the BSS problem $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{\varepsilon}$.

error, the additional noise and the finite number of available samples. Hence, the complementary nonnegativity information should be explicitly considered in order to obtain results consistent with physical reality [Cologner, 2012]. Furthermore, SeNICA has already shown its good ability in separating MRS signals [Wang et al., 2013a, Cologner et al., 2014a] and in classification of 3D dose distribution for prediction of rectal bleeding in prostate cancer radiotherapy [Cologner et al., 2014b].

We proposed to solve the SeNICA problem by imposing the nonnegativity constraint through cumulant-based ICA algorithms. As discussed in section 1.2, compared with negentropy, likelihood estimation and mutual information, the cumulants avoid the complicated procedure of estimating the pdf and the time consuming integral computation, thus they are easier to compute. Following the study in section 1.3, the properties of cumulants allow us to solve the SeNICA problem by means of nonnegative JDC methods, which will be addressed in the next section.

1.6.2 Nonnegative JDC algorithms

The Nonnegative JDC (NJDC) problem is defined by imposing the nonnegativity constraint on the joint transformation matrix, as follows:

Problem 1.7. *Given K symmetric matrices $\boldsymbol{C}^{(k)} \in \mathbb{R}^{N\times N}$ and an integer P, find a joint transformation matrix $\boldsymbol{A} \in \mathbb{R}^{N\times P}$ and K diagonal matrices $\boldsymbol{D}^{(k)} \in \mathbb{R}^{P\times P}$ such that for each index $k \in \{1, 2, \cdots, K\}$, we have:*

$$\boldsymbol{C}^{(k)} = \boldsymbol{A}\boldsymbol{D}^{(k)}\boldsymbol{A}^{\mathsf{T}} \tag{1.51}$$

subject to $\boldsymbol{A}$ having nonnegative components.

The NJDC problem can be solved by minimizing the criterion functions presented in section 1.3 with additional nonnegativity constraint. The existing NJDC algorithms are mainly contributed by the french research team of Laboratoire Traitement du Signal et de l'Image (LTSI) in Rennes. Cologner et al. resorted to use the following two schemes in order to impose nonnegativity on $\boldsymbol{A}$ in the direct-fit JDC criterion (1.29) [Cologner, 2012, Cologner et al., 2014a, Cologner et al., 2014c]:

$$\boldsymbol{A} = \boldsymbol{B} \boxdot \boldsymbol{B} = \boldsymbol{B}^{\boxdot 2}, \quad \boldsymbol{A} = \exp(\boldsymbol{B}) \tag{1.52}$$

namely a square change of variable and an exponential change of variable, respectively, where $\boldsymbol{B} \in \mathbb{R}^{N\times P}$ and where $\exp(\cdot)$ denotes the element-wise exponential operator of the matrix argument. Then the nonnegativity constrained problem was transformed into a unconstrained problem. As previously reviewed in section 1.4.2, the square change of

variable has been originally adopted by Chu et al. for solving NMF problem [Chu et al., 2004]. In addition, its effectiveness in the context of nonnegative CPD (without symmetry constraints) was shown by Royer et al. [Royer et al., 2011, Royer, 2013]. Cologner et al. proposed to minimize the direct-fit criterion (1.29) with square and exponential changes of variable, respectively, by applying two fundamental optimization strategies: line search and trust region procedures. Regarding the linear search method, they developed gradient descent, conjugate gradient and Newton-like algorithms combined with a search for optimal step-sizes [Cologner et al., 2014c]. Moreover, two Alternating Least Squares (ALS) procedures also accelerated with an Enhanced Line Search (ELS) were derived [Cologner et al., 2010, Cologner et al., 2011b, Cologner et al., 2011a, Cologner et al., 2014a]. Regarding the trust region approach, Cologner et al. proposed a Levenberg Marquardt (LM) method based on an approximation of the Hessian inside a trust region [Cologner et al., 2011b, Cologner et al., 2014c]. Nevertheless, like all the methods based on line search and trust region strategies, those methods appear to be dependent on initialization, and they therefore require a multi-initialization procedure in practice, leading to an increase of numerical complexity [Cologner, 2012].

1.6.3 Contribution and outline of thesis

In this thesis report, we also adopt the square change of variable $\boldsymbol{A} = \boldsymbol{B}^{\boxdot 2}$ due to its simplicity. Then the NJDC problem 1.7 can be transformed into the following unconstrained one:

Problem 1.8. *Given K symmetric matrices $\boldsymbol{C}^{(k)} \in \mathbb{R}^{N\times N}$ and an integer P, find a matrix $\boldsymbol{B} \in \mathbb{R}^{N\times P}$ and K diagonal matrices $\boldsymbol{D}^{(k)} \in \mathbb{R}^{P\times P}$ such that for each index $k \in \{1, 2, \cdots, K\}$, we have:*

$$\boldsymbol{C}^{(k)} = \boldsymbol{B}^{\boxdot 2}\boldsymbol{D}^{(k)}(\boldsymbol{B}^{\boxdot 2})^{\mathsf{T}} \tag{1.53}$$

Then the joint transformation matrix $\boldsymbol{A} = \boldsymbol{B}^{\boxdot 2}$ is nonnegative for sure.

In order to overcome the drawback of existing NJDC methods, we resort to use Jacobi-like optimization procedures, which are known to be less sensitive to initialization. More precisely, the matrix $\boldsymbol{B}$ is decomposed as a product of elementary matrices:

$$\boldsymbol{B} = \prod_{i,j} \boldsymbol{\Theta}^{(i,j)}(\xi_{i,j}) \tag{1.54}$$

where $\boldsymbol{\Theta}^{(i,j)}(\xi_{i,j})$ is a unit-determinant elementary matrix determined by only one parameter $\xi_{i,j}$, and where (i, j) is the pivot index. The decomposition (1.54) can be achieved by polar, LU or QR matrix factorizations. In a Jacobi-like optimization procedure, each elementary matrix $\boldsymbol{\Theta}^{(i,j)}(\xi_{i,j})$ is estimated sequentially by minimizing a proper criterion function. In fact, in our case, the parameter of one elementary matrix can be simply estimated by searching the roots of a polynomial function. We proposed five Jacobi-like methods based on different factorizations of $\boldsymbol{B}$ incorporating different cost functions defined in section 1.3, namely $\text{JDC}^{+}_{\text{LU-3}}$ [Wang et al., 2013a], $\text{iJDC}^{+}_{\text{LU-1}}$ [Wang et al., 2014a], $\text{iJDC}^{+}_{\text{QR-1}}$ [Wang et al., 2014b, Wang et al., 2014a], $\text{JDC}^{+}_{\text{LU-1}}$ [Wang et al., 2012, Wang et al., 2014c], and $\text{JDC}^{+}_{\text{LU-2}}$ [Wang et al., 2014c]. Besides those five Jacobi-like methods

based on a square change of variable, we developed a sixth algorithm, based on the Alternating Direction Method of Multipliers (ADMM) [Boyd et al., 2010], that tackles the NJDC problem 1.7 by minimizing the direct-fit JDC criterion (1.29). Such a method, namely $\text{JDC}^{+}_{\text{ADMM-3}}$, estimates $\boldsymbol{A}$ directly, and is well suited for large-scale problems. The differences between these six algorithms are summarized in table 1.2. The names of the algorithms are straightforwardly determined by the cost functions, optimization schemes and preprocessing step that they used. Taking the algorithm $<\text{iJDC}^{+}_{\text{LU-1}}>$ for example, i) the prefix "i" means that it requires to inverse each input matrix $\boldsymbol{C}^{(k)}$ as a preprocessing step; ii) the superscript "+" represents the nonnegativity constraint; iii) the subscript "LU" denotes that the square change of variable $\boldsymbol{A} = \boldsymbol{B}^{\boxdot 2}$ is used and $\boldsymbol{B}$ is decomposed using the LU matrix factorization; iv) the ensuing subscript "1" indicates the first indirect-fit criterion (1.27), while "2" and "3" signify the second indirect-fit criterion (1.28) and the direct-fit criterion (1.29), respectively. Furthermore, in some practical SeNICA problems, the dimension of the observation space must be reduced. However, the classical dimension compression procedure, such as whitening by PCA, breaks the nonnegativity property of the compressed mixing matrix. We proposed a new nonnegative compression method in [Wang et al., 2014d], which guarantees the nonnegativity of the compressed mixing matrix.

This thesis report is organized as follows:

- **Chapter 2**: This chapter is devoted to the detailed description of the proposed NJDC methods. First of all, we introduce some necessary preliminaries for our methods, including LU and QR matrix factorizations, the Jacobi-like optimization principle and the concept of ADMM. Next, the proposed algorithms are presented. We separate the six algorithms into four groups for the purpose of convenience. For each group, we provide the reformulated cost functions and the comprehensive derivation of the algorithms. The assumptions on each group of algorithms as well as the practical issues are also addressed. A numerical complexity study of the methods proposed in this thesis is given at the end of this chapter.

- **Chapter 3**: Our algorithms are evaluated on random synthetic matrices. Several classical nonorthogonal JDC methods without nonnegativity constraint, and one NJDC method proposed by Coloigner et al. [Coloigner et al., 2014c], are tested as baseline algorithms. The performance is assessed in terms of the matrix estimation accuracy and the numerical complexity. The convergence property, the influence of Signal to Noise Ratio (SNR), the impact of dimensions, the effect of coherence, and the influence of the modulus of uniqueness, are extensively studied through Monte Carlo experiments. The obtained results show that the proposed algorithms offer better estimation accuracy by exploiting the nonnegativity *a priori*, especially in difficult contexts.

- **Chapter 4**: The efficiency of SeNICA methods is illustrated by solving three BSS problems from very different application domains. The first one which is rooted in the biomedical diagnostics, consists of separating synthetic MRS spectra. The second one is an automatic music transcription problem. It aims at estimating the musical notes and their attributes from a single-channel music record by decomposing its magnitude spectrogram. The third one is devoted to separating

	Cost function			Use a square change of variable $\boldsymbol{A} = \boldsymbol{B}^{\square 2}$			Assumption on $\boldsymbol{A}$	
	indirect criteria		direct criterion	Yes, then factorize $\boldsymbol{B}$ by		No	$\boldsymbol{A}$ is square	
	(1.27)	(1.28)	(1.29)	LU factorization	QR factorization	Estimate $\boldsymbol{A}$ by ADMM	Yes[1]	No
$\text{iJDC}^{+}_{\text{LU-1}}$	√			√			√	
$\text{iJDC}^{+}_{\text{QR-1}}$	√				√		√	
$\text{JDC}^{+}_{\text{LU-1}}$	√			√				√
$\text{JDC}^{+}_{\text{LU-2}}$		√		√				√
$\text{JDC}^{+}_{\text{LU-3}}$			√	√				√
$\text{JDC}^{+}_{\text{ADMM-3}}$			√			√		√

Table 1.2: The differences between the proposed six nonnegative JDC methods according to the cost functions, the optimization schemes and the assumption on $\boldsymbol{A}$. When we use the square change variable $\boldsymbol{A} = \boldsymbol{B}^{\square 2}$, the LU and QR matrix factorizations are used to decompose $\boldsymbol{B}$, leading to the first five Jacobi-like optimization methods. The sixth method based on ADMM optimization estimates $\boldsymbol{A}$ without change of variable. [1] Regarding $\text{iJDC}^{+}_{\text{LU-1}}$ and $\text{iJDC}^{+}_{\text{QR-1}}$, the joint transformation matrix $\boldsymbol{A}$ is constrained to be square. And a preprocessing step that inverses each input matrix $\boldsymbol{C}^{(k)}$ is required.

digital images which are degraded by the so-called show-through effect. Such a problem is usually caused by the seeping of ink from the reverse side or scanning a semi-transparent paper. The behavior of the proposed methods are compared with classical ICA, NICA, NMF and semi-NMF methods. The experimental results show that by fully exploiting the prior information of data, such as the nonnegativity of the mixing matrix and the statistical independence of the sources, the proposed methods achieve better estimation results.

- **Chapter 5**: This final section concludes the thesis report by providing a closing discussion on the contributions presented in the chapters as well as possible future extensions.

1.7 List of publications

1. **International conference without proceedings**

   [**Wang et al., 2012**] Wang, L., Albera, L., Shu, H. Z., and Senhadji, L. (2012). Decomposition of semi-nonnegative semi-symmetric three-way tensors based on LU matrix factorization. In *SIAM 2012 Conference on Applied Linear Algebra*, Valencia, Spain.

2. **International conferences with proceedings**

   [**Wang et al., 2013b**] Wang, L., Albera, L., Shu, H. Z., and Senhadji, L. (2013b). A new Jacobi-like nonnegative joint diagonalization by congruence. In *EUSIPCO'13, Proceedings of the XXI European Signal Processing Conference*, pages 1–5, Marrakech, Morocco.

   [**Wang et al., 2014b**] Wang, L., Albera, L., Kachenoura, A., Shu, H. Z., and Senhadji, L. (2014b). CP decomposition of semi-nonnegative semi-symmetric tensors based on QR matrix factorization. In *SAM'14, Proceedings of the Eighth IEEE Sensor Array and Multichannel Signal Processing Workshop*, pages 449–452, A Coruna, Spain. (invited paper).

   [**Wang et al., 2014d**] Wang, L., Kachenoura, A., Albera, L., Karfoul, A., Shu, H. Z., and Senhadji, L. (2014d). Nonnegative compression for semi-nonnegative independent component analysis. In *SAM'14, Proceedings of the Eighth IEEE Sensor Array and Multichannel Signal Processing Workshop*, pages 81–84, A Coruna, Spain.

3. **International journals**

   [**Wang et al., 2013a**] Wang, L., Albera, L., Kachenoura, A., Shu, H. Z., and Senhadji, L. (2013a). Nonnegative joint diagonalization by congruence based on LU matrix factorization. *IEEE Signal Processing Letters*, 20(8):807–810.

   [**Wang et al., 2014a**] Wang, L., Albera, L., Kachenoura, A., Shu, H. Z., and Senhadji, L. (2014a). Canonical polyadic decomposition of 3rd order semi-nonnegative semi-symmetric tensors using LU and QR matrix factorizations. Accepted for publication in *EURASIP Journal on Advances in Signal Processing*.

[Coloigner et al., 2014b] Coloigner, J., Fargeas, A., Kachenoura, A., Wang, L., Senhadji, L., De Crevoisier, R., Acosta, O., and Albera, L. (2014b). A novel classification method for prediction of rectal bleeding in prostate cancer radiotherapy based on a semi-nonnegative ICA of 3D planned dose distributions. Accepted for publication in *IEEE Journal of Biomedical and Health Informatics*.

[Wang et al., 2014c] Wang, L., Albera, L., Kachenoura, A., Shu, H. Z., and Senhadji, L. (2014c). Parameterization of the inverse of the nonnegative matrix for nonnegative joint diagonalization. In preparation for submission to *IEEE Transactions on Signal Processing*.

Chapter 2

Nonnegative joint diagonalization by congruence

This chapter is devoted to provide an in-depth scrutiny of the proposed six novel NJDC algorithms. The first five algorithms impose the nonnegativity constraint of the joint transformation matrix $\boldsymbol{A}$ by means of a square change of variable $\boldsymbol{A} = \boldsymbol{B}^{\boxdot 2}$ and estimate $\boldsymbol{B}$ using a Jacobi-like optimization procedure. They differ in the cost functions, matrix factorization schemes and preprocessing steps. The sixth algorithm estimates $\boldsymbol{A}$ directly through ADMM, which is an improved variant of the classical augmented Lagrangian method.

The remainder of this chapter is organized as follows. In section 2.1, we introduce some necessary preliminaries for the proposed methods, including LU and QR matrix factorizations, the Jacobi-like optimization principle and the concept of ADMM. Then the six algorithms are presented in the following order, $\text{JDC}^{+}_{\text{LU-3}}$ in section 2.2, $\text{iJDC}^{+}_{\text{LU-1}}$ and $\text{iJDC}^{+}_{\text{QR-1}}$ in section 2.3, $\text{JDC}^{+}_{\text{LU-1}}$ and $\text{JDC}^{+}_{\text{LU-2}}$ in section 2.4, and $\text{JDC}^{+}_{\text{ADMM-3}}$ in section 2.5. In each section, firstly we present the comprehensive derivation of the algorithms for estimating a square joint transformation matrix $\boldsymbol{A} \in \mathbb{R}_{+}^{N\times N}$. Secondly we discuss the possible generalization of the algorithms to the non-square case $\boldsymbol{A} \in \mathbb{R}_{+}^{N\times P}$ with $N > P$, as well as some practical issues. In addition, we provide a numerical complexity analysis of the methods proposed in this thesis in section 2.6.

2.1 Preliminaries

2.1.1 LU and QR matrix factorizations

Let us recall the following elementary unit-determinant matrices:

Definition 2.1. *A unit upper (or lower) triangular matrix is a upper (or lower, respectively) triangular matrix whose main diagonal elements are equal to* 1*.*

Definition 2.2. *An elementary upper triangular matrix with parameters* $\{i, j, u_{i,j}\}$ *and* $i < j$ *is a unit upper triangular matrix whose non-diagonal elements are zeros except the* (i, j)*-th entry, which is equal to* $u_{i,j}$*.*

$\boldsymbol{U}^{(i,j)}(u_{i,j})$ with $i < j$ and $i, j \in \{1, 2, \ldots, N\}$ denotes an elementary upper triangular matrix:

$$\boldsymbol{U}^{(i,j)}(u_{i,j}) = \begin{pmatrix} \boldsymbol{I}_{i-1} & \vdots & \mathbf{0} & \vdots & \mathbf{0} \\ \ldots & 1 & \ldots & u_{i,j} & \ldots \\ \mathbf{0} & \vdots & \boldsymbol{I}_{j-i-1} & \vdots & \mathbf{0} \\ \ldots & 0 & \ldots & 1 & \ldots \\ \mathbf{0} & \vdots & \mathbf{0} & \vdots & \boldsymbol{I}_{N-j} \end{pmatrix} \tag{2.1}$$

Definition 2.3. *An elementary lower triangular matrix with parameters* $\{i, j, \ell_{i,j}\}$ *and* $i > j$ *is a unit lower triangular matrix whose non-diagonal elements are zeros except the* (i, j)*-th entry, which is equal to* $\ell_{i,j}$*.*

$\boldsymbol{L}^{(i,j)}(\ell_{i,j})$ with $j < i$ and $i, j \in \{1, 2, \ldots, N\}$ corresponds to an elementary lower triangular matrix:

$$\boldsymbol{L}^{(i,j)}(\ell_{i,j}) = \begin{pmatrix} \boldsymbol{I}_{i-1} & \vdots & \mathbf{0} & \vdots & \mathbf{0} \\ \ldots & 1 & \ldots & 0 & \ldots \\ \mathbf{0} & \vdots & \boldsymbol{I}_{j-i-1} & \vdots & \mathbf{0} \\ \ldots & \ell_{i,j} & \ldots & 1 & \ldots \\ \mathbf{0} & \vdots & \mathbf{0} & \vdots & \boldsymbol{I}_{N-j} \end{pmatrix} \tag{2.2}$$

Definition 2.4. *A Givens rotation matrix with parameters* $\{i, j, \theta_{i,j}\}$ *and* $i < j$ *is equal to an identity matrix except for the* (i, i)*-th,* (j, j)*-th,* (i, j)*-th and* (j, i)*-th entries, which are equal to* $\cos(\theta_{i,j})$*,* $\cos(\theta_{i,j})$*,* $-\sin(\theta_{i,j})$ *and* $\sin(\theta_{i,j})$*, respectively.*

$\boldsymbol{Q}^{(i,j)}(\theta_{i,j})$ with $i < j$ and $i, j \in \{1, 2, \ldots, N\}$ indicates the corresponding Givens rotation matrix:

$$\boldsymbol{Q}^{(i,j)}(\theta_{i,j}) = \begin{pmatrix} \boldsymbol{I}_{i-1} & \vdots & \mathbf{0} & \vdots & \mathbf{0} \\ \ldots & \cos(\theta_{i,j}) & \ldots & -\sin(\theta_{i,j}) & \ldots \\ \mathbf{0} & \vdots & \boldsymbol{I}_{j-i-1} & \vdots & \mathbf{0} \\ \ldots & \sin(\theta_{i,j}) & \ldots & \cos(\theta_{i,j}) & \ldots \\ \mathbf{0} & \vdots & \mathbf{0} & \vdots & \boldsymbol{I}_{N-j} \end{pmatrix} \tag{2.3}$$

Now, let us consider the following lemmas:

Lemma 2.1. *Any* $(N \times N)$ *unit upper (or lower) triangular matrix can be factorized as a product of* $N(N-1)/2$ *elementary upper (or lower, respectively) triangular matrices.*

The proof of lemma 2.1 is straightforward by reducing the unit upper (or lower) triangular matrix into an identity matrix by means of Gaussian elimination [Golub and Van Loan, 1996, Chapter 3].

Lemma 2.2. *Any* $(N \times N)$ *orthonormal matrix can be factorized as a product of, at most,* $N(N-1)/2$ *Givens rotation matrices.*

The proof of lemma 2.2 can be found in [Vaidyanathan, 1993, Chapter 14]. Furthermore, the above two decompositions are not unique.

Then we review the following two classical matrix factorization methods for a non-singular matrix $\boldsymbol{B} \in \mathbb{R}^{N\times N}$ [Golub and Van Loan, 1996],

LU matrix factorization:

$$\boldsymbol{B} = \boldsymbol{L}\boldsymbol{U}\boldsymbol{\Lambda}_1\boldsymbol{\Pi}_1 \tag{2.4}$$

where $\boldsymbol{L} \in \mathbb{R}^{N\times N}$ is a unit lower triangular matrix, where $\boldsymbol{U} \in \mathbb{R}^{N\times N}$ is a unit upper triangular matrix, where $\boldsymbol{\Lambda}_1 \in \mathbb{R}^{N\times N}$ is a diagonal matrix, and where $\boldsymbol{\Pi}_1 \in \mathbb{R}^{N\times N}$ is a permutation matrix.

QR matrix factorization:

$$\boldsymbol{B} = \boldsymbol{Q}\boldsymbol{R}\boldsymbol{\Lambda}_2\boldsymbol{\Pi}_2 \tag{2.5}$$

where $\boldsymbol{Q} \in \mathbb{R}^{N\times N}$ is an orthonormal matrix, where $\boldsymbol{R} \in \mathbb{R}^{N\times N}$ is a unit upper triangular matrix, where $\boldsymbol{\Lambda}_2 \in \mathbb{R}^{N\times N}$ is a diagonal matrix, and where $\boldsymbol{\Pi}_2 \in \mathbb{R}^{N\times N}$ is a permutation matrix.

The LU and QR matrix decompositions are important linear algebra tools that are widely used in both scientific and engineering applications.

Now, let us return to the NJDC problem 1.8, where the nonnegativity constraint of the joint transformation matrix $\boldsymbol{A} \in \mathbb{R}_+^{N\times N}$ is imposed through a square change variable $\boldsymbol{A} = \boldsymbol{B}^{\boxdot 2}$. By means of the LU and QR matrix factorizations of $\boldsymbol{B} \in \mathbb{R}^{N\times N}$, the jointly diagonalizable matrices $\boldsymbol{C}^{(k)} = \boldsymbol{A}\boldsymbol{D}^{(k)}\boldsymbol{A}^{\mathsf{T}}$ can be rewritten in the following two equivalent forms, respectively:

$$\boldsymbol{C}^{(k)} = (\boldsymbol{L}\boldsymbol{U})^{\boxdot 2}(\boldsymbol{\Lambda}_1^{\boxdot 2}\,\boldsymbol{\Pi}_1\boldsymbol{D}^{(k)}\boldsymbol{\Pi}_1^{\mathsf{T}}\boldsymbol{\Lambda}_1^{\boxdot 2})[(\boldsymbol{L}\boldsymbol{U})^{\boxdot 2}]^{\mathsf{T}} = (\boldsymbol{L}\boldsymbol{U})^{\boxdot 2}\,\boldsymbol{D}_1^{(k)}[(\boldsymbol{L}\boldsymbol{U})^{\boxdot 2}]^{\mathsf{T}} \tag{2.6}$$

$$\boldsymbol{C}^{(k)} = (\boldsymbol{Q}\boldsymbol{R})^{\boxdot 2}(\boldsymbol{\Lambda}_2^{\boxdot 2}\,\boldsymbol{\Pi}_2\boldsymbol{D}^{(k)}\boldsymbol{\Pi}_2^{\mathsf{T}}\boldsymbol{\Lambda}_2^{\boxdot 2})[(\boldsymbol{Q}\boldsymbol{R})^{\boxdot 2}]^{\mathsf{T}} = (\boldsymbol{Q}\boldsymbol{R})^{\boxdot 2}\,\boldsymbol{D}_2^{(k)}[(\boldsymbol{Q}\boldsymbol{R})^{\boxdot 2}]^{\mathsf{T}} \tag{2.7}$$

where:

$$\boldsymbol{D}_1^{(k)} = \boldsymbol{\Lambda}_1^{\boxdot 2}\,\boldsymbol{\Pi}_1\boldsymbol{D}^{(k)}\boldsymbol{\Pi}_1^{\mathsf{T}}\boldsymbol{\Lambda}_1^{\boxdot 2}, \quad \boldsymbol{D}_2^{(k)} = \boldsymbol{\Lambda}_2^{\boxdot 2}\,\boldsymbol{\Pi}_2\boldsymbol{D}^{(k)}\boldsymbol{\Pi}_2^{\mathsf{T}}\boldsymbol{\Lambda}_2^{\boxdot 2} \tag{2.8}$$

It is straightforward to verify that $\boldsymbol{D}_1^{(k)}$ and $\boldsymbol{D}_2^{(k)}$ are also diagonal matrices. Therefore, due to the scale and permutation indeterminacies of the JDC problem, the Hadamard square root of the joint transformation matrix, say $\boldsymbol{B}$, can be expressed as $\boldsymbol{B} = \boldsymbol{L}\boldsymbol{U}$ and $\boldsymbol{B} = \boldsymbol{Q}\boldsymbol{R}$ without loss of generality. Moreover, by incorporating lemma 2.1 and lemma 2.2, we obtain the following two elementary factorizations of $\boldsymbol{B}$:

$$\boldsymbol{B} = \prod_{j=1}^{N}\prod_{i=j+1}^{N}\boldsymbol{L}^{(i,j)}(\ell_{i,j})\prod_{i=1}^{N}\prod_{j=i+1}^{N}\boldsymbol{U}^{(i,j)}(u_{i,j}) \tag{2.9}$$

$$\boldsymbol{B} = \prod_{i=1}^{N}\prod_{j=i+1}^{N}\boldsymbol{Q}^{(i,j)}(\theta_{i,j})\prod_{i=1}^{N}\prod_{j=i+1}^{N}\boldsymbol{U}^{(i,j)}(u'_{i,j}) \tag{2.10}$$

where $\boldsymbol{L}^{(i,j)}(\ell_{i,j}) \in \mathbb{R}^{N\times N}$, $\boldsymbol{U}^{(i,j)}(u_{i,j}) \in \mathbb{R}^{N\times N}$ and $\boldsymbol{Q}^{(i,j)}(\theta_{i,j}) \in \mathbb{R}^{N\times N}$ are an elementary lower triangular matrix, an elementary upper triangular matrix and a Givens rotation matrix, respectively. As a consequence, estimating $\boldsymbol{B}$ for the NJDC problem 1.8 is converted to the estimate of $N(N-1)$ parameters: $\ell_{i,j}$ and $u_{i,j}$ for the LU decomposition (2.9), or $\theta_{i,j}$ and $u'_{i,j}$ for the QR decomposition (2.10). It is noteworthy that by using such elementary decompositions, the trivial zero solution of the NJDC problem is avoided, since $\boldsymbol{B}$ is implicitly constrained to have a unit determinant. Instead of simultaneously computing these $N(N-1)$ parameters, we propose to adopt a Jacobi-like procedure which performs $N(N-1)$ sequential optimizations.

2.1.2 A Jacobi-like optimization procedure

Most NJDC algorithms presented in this chapter are Jacobi-like. Therefore we highlight in this section the principle of a Jacobi-like optimization procedure.

The purpose of the NJDC problem 1.8 is to find the Hadamard square root $\boldsymbol{B}$ of the nonnegative joint transformation matrix $\boldsymbol{A}$. We aim at estimating the optimal matrix $\boldsymbol{B}$ by minimizing a proper criterion function Ψ with respect to $\boldsymbol{B}$. $\Psi(\boldsymbol{B})$ could be reformulated from the JDC criteria presented in section 1.3.3 incorporating the square change of variable $\boldsymbol{A} = \boldsymbol{B}^{\boxdot 2}$. According to the discussion in section 2.1.1, $\boldsymbol{B}$ can be decomposed as a product of elementary matrices:

$$\boldsymbol{B} = \prod_{i,j} \boldsymbol{\Theta}^{(i,j)}(\xi_{i,j}) \tag{2.11}$$

where $\boldsymbol{\Theta}^{(i,j)}(\xi_{i,j})$ can be elementary triangular matrices or Given rotation matrices. Instead of computing these elementary matrices simultaneously, a Jacobi-like optimization procedure consists of estimating these elementary matrices successively and building up $\boldsymbol{B}$ gradually as the product of all the estimated elementary matrices [De Lathauwer, 2010]. More precisely, we can start by an initialization of $\boldsymbol{B}$, namely $\boldsymbol{B}^{(0)}$, where the superscript (0) denotes the iteration number. At the it-th Jacobi-like iteration, we estimate one of the elementary matrices, namely $\boldsymbol{\Theta}^{(i,j)}(\xi_{i,j})$, with a specific (i,j) index couple. The optimal parameter $\xi_{i,j}$ is chosen as the global minimum of a local cost function as follows:

$$\xi_{i,j}^{\text{opt}} = \underset{\xi_{i,j}}{\operatorname{argmin}}\, \Psi(\boldsymbol{B}^{(it-1)}\boldsymbol{\Theta}^{(i,j)}(\xi_{i,j})) \tag{2.12}$$

where $\boldsymbol{B}^{(it-1)}$ is the estimate of $\boldsymbol{B}$ at the previous iteration thus it is a constant term. Compared to the minimization of the global criterion $\Psi(\boldsymbol{B})$ with respect to $\boldsymbol{B}$ directly, generally the local optimization (2.12) is much easier to solve. Once $\xi_{i,j}^{\text{opt}}$ is obtained, $\boldsymbol{B}^{(it-1)}$ is updated as follows:

$$\boldsymbol{B}^{(it)} = \boldsymbol{B}^{(it-1)}\boldsymbol{\Theta}^{(i,j)}(\xi_{i,j}^{\text{opt}}) \tag{2.13}$$

Then the $(it+1)$-th Jacobi-like iteration continues to compute $\boldsymbol{\Theta}^{(i,j)}(\xi_{i,j})$ with the next (i,j) index. Such a procedure is repeated until the global criterion $\Psi(\boldsymbol{B})$ is minimized. In the classical Jacobi-like iteration, the choice of the order of the index (i,j) is crucial. From the standpoint that the decrease of the cost function shall be as large as possible in each Jacobi-like iteration, it makes sense to choose the pair (i',j') for which $\Psi(\boldsymbol{B}^{(it-1)}\boldsymbol{\Theta}^{(i',j')}(\xi_{i',j'}))$ is minimal among all potential (i,j) indices. However, finding the optimal (i,j) index in each iteration is computationally expensive. Therefore, in practice, when going through the different subproblems (2.12), one rather follows a fixed (i,j) sequence, for example:

$$(1,2),(1,3),\ldots,(1,N),(2,1),(2,3),\ldots,(2,N),\ldots,(N-1,N) \tag{2.14}$$

This technique is called the cyclic Jacobi-like algorithm [De Lathauwer, 2010]. The pseudo-code of the cyclic Jacobi-like optimization procedure for a NJDC problem is outlined in the appendix C.1.

2.1.3 The alternating direction method of multipliers

In a finite-dimensional setting, ADMM [Glowinski and Marroco, 1975,Gabay and Mercier, 1976,Boyd et al., 2010] is a method that proposes to combine the decomposability of the dual ascent method [Shor, 1985, Bertsekas, 1999, Nedić and Ozdaglar, 2010] with the superior convergence properties of the method of multipliers [Hestenes, 1969, Miele et al., 1972, Bertsekas, 1996]. The algorithm solves structured convex problems of the following form:

$$\begin{aligned} &\underset{\boldsymbol{x},\boldsymbol{y}}{\text{minimize}} \ \mathrm{f}(\boldsymbol{x}) + \mathrm{g}(\boldsymbol{y}) \\ &\text{subject to } \boldsymbol{P}\boldsymbol{x} + \boldsymbol{Q}\boldsymbol{y} = \boldsymbol{c} \end{aligned} \tag{2.15}$$

where $\mathrm{f}(\cdot)$ and $\mathrm{g}(\cdot)$ are two convex functions defined on closed convex subsets of real numbers of dimensions N_1 and N_2, respectively, with $\boldsymbol{x} \in \mathbb{R}^{N_1}$, $\boldsymbol{y} \in \mathbb{R}^{N_2}$, $\boldsymbol{P} \in \mathbb{R}^{N_3 \times N_1}$, $\boldsymbol{Q} \in \mathbb{R}^{N_3 \times N_2}$ and $\boldsymbol{c} \in \mathbb{R}^{N_3}$. The optimal value of the problem (2.15) will be denoted as follows:

$$p^* = \inf\{\mathrm{f}(\boldsymbol{x}) + \mathrm{g}(\boldsymbol{y}) | \boldsymbol{P}\boldsymbol{x} + \boldsymbol{Q}\boldsymbol{y} = \boldsymbol{c}\} \tag{2.16}$$

where $\inf\{\cdot\}$ denotes infimum. The augmented Lagrangian function of (2.15) is formed as follows:

$$\mathrm{L}_\rho(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) = \mathrm{f}(\boldsymbol{x}) + \mathrm{g}(\boldsymbol{y}) + \boldsymbol{\lambda}^\mathsf{T}(\boldsymbol{P}\boldsymbol{x} + \boldsymbol{Q}\boldsymbol{y} - \boldsymbol{c}) + \frac{\rho}{2} \|\boldsymbol{P}\boldsymbol{x} + \boldsymbol{Q}\boldsymbol{y} - \boldsymbol{c}\|_F^2 \tag{2.17}$$

where $\boldsymbol{\lambda}$ is a Lagrangian multiplier vector and $\rho > 0$ is a penalty parameter. ADMM consists of the iterations:

$$\boldsymbol{x}^{(it)} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \, \mathrm{L}_\rho(\, \boldsymbol{x}, \boldsymbol{y}^{(it-1)}, \boldsymbol{\lambda}^{(it-1)}\,) \tag{2.18}$$

$$\boldsymbol{y}^{(it)} = \underset{\boldsymbol{y}}{\operatorname{argmin}} \, \mathrm{L}_\rho(\, \boldsymbol{x}^{(it)}, \boldsymbol{y}, \boldsymbol{\lambda}^{(it-1)}\,) \tag{2.19}$$

$$\boldsymbol{\lambda}^{(it)} = \boldsymbol{\lambda}^{(it-1)} + \rho(\, \boldsymbol{P}\boldsymbol{x}^{(it)} + \boldsymbol{Q}\boldsymbol{y}^{(it)} - \boldsymbol{c}\,) \tag{2.20}$$

The ADMM algorithm resembles the dual ascent method and the method of multipliers [Boyd et al., 2010]. It includes an $\boldsymbol{x}$-minimization step (2.18), a $\boldsymbol{y}$-minimization step (2.19), and a dual variable update step (2.20). It is noteworthy that the classical Augmented Lagrangian Multiplier (ALM) method [Hestenes, 1969, Rockafellar, 1973, Fortin and Glowinski, 1983] minimizes (2.17) jointly with respect to both $\boldsymbol{x}$ and $\boldsymbol{y}$. For example, ALM replaces the steps (2.18) and (2.19) by

$$[\boldsymbol{x}^{(it)}, \boldsymbol{y}^{(it)}] = \underset{\boldsymbol{x},\boldsymbol{y}}{\operatorname{argmin}} \, \mathrm{L}_\rho(\, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}^{(it-1)}\,) \tag{2.21}$$

which could be more difficult to solve since it involves both $\mathrm{f}(\boldsymbol{x})$ and $\mathrm{g}(\boldsymbol{y})$. On the other side, in ADMM, $\boldsymbol{x}$ and $\boldsymbol{y}$ are updated in an alternating manner, which specifies the term "alternating direction". ADMM can be regarded as a special version of the method of multipliers where a single Gauss-Seidel pass over $\boldsymbol{x}$ and $\boldsymbol{y}$ is used instead of the usual joint minimization [Boyd et al., 2010]. The convergence analysis for ADMM was addressed in [Boyd et al., 2010] and references therein.

In the following three sections, we present five NJDC algorithms for solving problem 1.8 by means of a square change of variable and a Jacobi-like optimization procedure. For the sake of convenience, firstly we assume that the joint transformation matrix $\boldsymbol{A} \in \mathbb{R}_{+}^{N\times P}$ in problem 1.8 is a square matrix such that $N = P$. Given K symmetric matrices $\boldsymbol{C}^{(k)} \in \mathbb{R}^{N\times N}$, each algorithm is derived for the purpose of estimating $\boldsymbol{A} \in \mathbb{R}_{+}^{N\times N}$. Secondly, we discuss the possible extensions of these algorithms in order to seek a non-square matrix $\boldsymbol{A} \in \mathbb{R}_{+}^{N\times P}$ where $N > P$.

2.2 The $\text{JDC}^{+}_{\text{LU-3}}$ algorithm

2.2.1 Algorithm derivation

Let us impose the nonnegativity constraint in the JDC criteria presented in section 1.3.3. The indirect-fit criteria (1.27) and (1.28) seem difficult to handle since it is quite complicated to impose the nonnegativity constraint on $\boldsymbol{A}$, meanwhile to perform the optimization on its inverse $\boldsymbol{A}^{-1}$. In other words, it is difficult to parameterize the set of matrices which are the inverse of nonnegative matrices. On the other side, the direct-fit JDC criterion (1.29) carries out the optimization on $\boldsymbol{A}$ directly. Therein the nonnegativity of $\boldsymbol{A}$ can be enforced straightforwardly. Therefore, the first and natural idea is to incorporate the square change of variable in the direct-fit JDC criterion (1.29), leading to the following nonnegativity constrained direct-ft JDC criterion:

$$\Psi_3^{+}(\boldsymbol{B}, \{\boldsymbol{D}^{(k)}\}) = \sum_{k=1}^{K} \left\| \boldsymbol{C}^{(k)} - (\boldsymbol{B}^{\boxdot 2})\boldsymbol{D}^{(k)}(\boldsymbol{B}^{\boxdot 2})^{\mathsf{T}} \right\|_F^2 \tag{2.22}$$

Minimizing (2.22) with respect to both $\boldsymbol{B}$ and the set $\{\boldsymbol{D}^{(k)}\}$ is difficult. Consequently, as Yeredor's ACDC algorithm [Yeredor, 2002], the proposed $\text{JDC}^{+}_{\text{LU-3}}$ algorithm performs the minimization of (2.22) alternately with respect to $\boldsymbol{B}$ and $\{\boldsymbol{D}^{(k)}\}$. In order to estimate $\boldsymbol{B}$, by means of the LU matrix factorization (2.9) and the Jacobi-like optimization procedure, the high dimensional optimization is reduced to search a sequence of sparse triangular matrices. Therefore, $\text{JDC}^{+}_{\text{LU-3}}$ consists of the following two minimization schemes:

1. The AC (Alternating Columns) phase minimizes (2.22) with respect to $\boldsymbol{B}$ via Jacobi-like iterations. In each iteration, one elementary triangular matrix factor $\boldsymbol{L}^{(i,j)}(\ell_{i,j})$ or $\boldsymbol{U}^{(i,j)}(u_{i,j})$ of $\boldsymbol{B}$ is identified, while keeping its other factors as well as the diagonal matrices $\{\boldsymbol{D}^{(k)}\}$ fixed. For example,

$$\ell_{i,j}^{(it)} = \underset{\ell_{i,j}}{\operatorname{argmin}}\, \Psi_3^{+}(\, \boldsymbol{B}^{(it-1)}\boldsymbol{L}^{(i,j)}(\ell_{i,j}), \{\boldsymbol{D}^{(k,it-1)}\}\,) \tag{2.23}$$

   The sequential updates of $\boldsymbol{B}$ by right multiplying it by $\boldsymbol{L}^{(i,j)}(\ell_{i,j})$ or $\boldsymbol{U}^{(i,j)}(u_{i,j})$ with a different index couple (i, j) affects the j-th column vector of $\boldsymbol{B}$ sequentially. Thus we retain the terminology "Alternating Columns phase" from the ACDC algorithm.

2. The DC (Diagonal Centers) phase minimizes (2.22) with respect to the diagonal matrix set $\{\boldsymbol{D}^{(k)}\}$ while keeping $\boldsymbol{B}$ fixed:

$$\{\boldsymbol{D}^{(k,it)}\} = \underset{\{\boldsymbol{D}^{(k)}\}}{\operatorname{argmin}}\, \Psi_3^{+}(\boldsymbol{B}^{(it-1)}, \{\boldsymbol{D}^{(k)}\}) \tag{2.24}$$

Hence, $\text{JDC}^{+}_{\text{LU-3}}$ is an alternating optimization approach including a Jacobi-like optimization phase.

2.2.1.1 AC phase

In this phase, we minimize (2.22) with respect to $\boldsymbol{L}^{(i,j)}(\ell_{i,j})$ and $\boldsymbol{U}^{(i,j)}(u_{i,j})$. In order to simplify the notations, we define an elementary triangular matrix $\boldsymbol{T}^{(i,j)}(t_{i,j})$ such that:

$$\boldsymbol{T}^{(i,j)}(t_{i,j}) = \begin{cases} \boldsymbol{L}^{(i,j)}(\ell_{i,j}), & \text{if } i > j \\ \boldsymbol{U}^{(i,j)}(u_{i,j}), & \text{if } i < j \end{cases} \tag{2.25}$$

Consequently, the LU decomposition (2.9) of $\boldsymbol{B}$ can be simply denoted as follows:

$$\boldsymbol{B} = \prod_{i \neq j}^{N(N-1)} \boldsymbol{T}^{(i,j)}(t_{i,j}) \tag{2.26}$$

Instead of simultaneously computing these $N(N-1)$ matrices, each elementary triangular matrix with a selected (i,j) index is identified sequentially. For the sake of convenience, we use $\widehat{\boldsymbol{A}}$, $\widehat{\boldsymbol{B}}$ and $\widehat{\boldsymbol{D}}^{(k)}$ to denote the estimate of $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{D}^{(k)}$ at $(it-1)$-th iteration instead of $\boldsymbol{A}^{(it-1)}$, $\boldsymbol{B}^{(it-1)}$ and $\boldsymbol{D}^{(k,it-1)}$, respectively. At the current iteration, in other words, at the it-th iteration, the update of $\widehat{\boldsymbol{B}}$ by one $\boldsymbol{T}^{(i,j)}(t_{i,j})$, denoted by $\widehat{\boldsymbol{B}}^{(\text{new})}$, is defined as follows:

$$\widehat{\boldsymbol{B}}^{(\text{new})} = \widehat{\boldsymbol{B}}\boldsymbol{T}^{(i,j)}(t_{i,j}) \tag{2.27}$$

Inserting (2.27) into the optimization (2.23), the estimation of $t_{i,j}$ only consists of minimizing the following local cost function:

$$\Psi^{+}_{3,\text{AC}}(t_{i,j}) = \sum_{k=1}^{K} \left\| \boldsymbol{C}^{(k)} - [(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}]\, \widehat{\boldsymbol{D}}^{(k)} [(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}]^{\mathsf{T}} \right\|_F^2 \tag{2.28}$$

The expression of the Hadamard square of the matrix $\widehat{\boldsymbol{B}}^{(\text{new})}$ is shown in the following proposition.

Proposition 2.1. *$\widehat{\boldsymbol{A}}^{(new)} = (\widehat{\boldsymbol{B}}^{(new)})^{\boxdot 2} = (\widehat{\boldsymbol{B}}\boldsymbol{T}^{(i,j)}(t_{i,j}))^{\boxdot 2}$ can be expressed as a function of $t_{i,j}$ as follows:*

$$\widehat{\boldsymbol{A}}^{(new)} = (\widehat{\boldsymbol{B}}^{(new)})^{\boxdot 2} = \widehat{\boldsymbol{B}}^{\boxdot 2}\boldsymbol{T}^{(i,j)}(t_{i,j}^2) + 2\, t_{i,j}(\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)\boldsymbol{e}_j^{\mathsf{T}} \tag{2.29}$$

where $\widehat{\boldsymbol{b}}_i$ and $\widehat{\boldsymbol{b}}_j$ denote the i-th and j-th column vectors of $\widehat{\boldsymbol{B}}$, respectively, and where $\boldsymbol{e}_j$ is the j-th column vector of the identity matrix $\boldsymbol{I}_N$.

The proof of proposition 2.1 is provided in appendix A.1. Therefore, the cost function (2.28) can be rewritten as an explicit function of $t_{i,j}$ as follows:

$$\Psi^{+}_{3,\text{AC}}(t_{i,j}) = \sum_{k=1}^{K} \left\| \boldsymbol{F}^{(k,4)} t_{i,j}^4 + \boldsymbol{F}^{(k,3)} t_{i,j}^3 + \boldsymbol{F}^{(k,2)} t_{i,j}^2 + \boldsymbol{F}^{(k,1)} t_{i,j} + \boldsymbol{F}^{(k,0)} \right\|_F^2 \tag{2.30}$$

where:

$$\boldsymbol{F}^{(k,4)} = -\widehat{d}_j^{(k)} \left[(\widehat{\boldsymbol{b}}_i^{\boxdot 2})(\widehat{\boldsymbol{b}}_i^{\boxdot 2})^{\mathsf{T}} \right] \tag{2.31}$$

$$\boldsymbol{F}^{(k,3)} = -2\widehat{d}_j^{(k)} \left[(\widehat{\boldsymbol{b}}_i^{\boxdot 2})(\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)^{\mathsf{T}} + (\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)(\widehat{\boldsymbol{b}}_i^{\boxdot 2})^{\mathsf{T}} \right] \tag{2.32}$$

$$\boldsymbol{F}^{(k,2)} = -\widehat{d}_j^{(k)} \left[(\widehat{\boldsymbol{b}}_i^{\boxdot 2})(\widehat{\boldsymbol{b}}_j^{\boxdot 2})^{\mathsf{T}} + (\widehat{\boldsymbol{b}}_j^{\boxdot 2})(\widehat{\boldsymbol{b}}_i^{\boxdot 2})^{\mathsf{T}} + 4(\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)(\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)^{\mathsf{T}} \right] \tag{2.33}$$

$$\boldsymbol{F}^{(k,1)} = -2\widehat{d}_j^{(k)} \left[(\widehat{\boldsymbol{b}}_j^{\boxdot 2})(\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)^{\mathsf{T}} + (\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)(\widehat{\boldsymbol{b}}_j^{\boxdot 2})^{\mathsf{T}} \right] \tag{2.34}$$

$$\boldsymbol{F}^{(k,0)} = \boldsymbol{C}^{(k)} - (\widehat{\boldsymbol{B}}^{\boxdot 2})\boldsymbol{D}^{(k)}(\widehat{\boldsymbol{B}}^{\boxdot 2})^{\mathsf{T}} \tag{2.35}$$

and where $\widehat{d}_j^{(k)}$ denotes the (j,j)-th entry of $\widehat{\boldsymbol{D}}^{(k)}$. (2.30) can be expressed in the following compact matrix form:

$$\Psi^{+}_{3,\text{AC}}(t_{i,j}) = \sum_{k=1}^{K} \left\| \boldsymbol{F}^{(k)} \boldsymbol{\tau}_{i,j} \right\|_F^2 = \boldsymbol{\tau}_{i,j}^{\mathsf{T}} \boldsymbol{G} \boldsymbol{\tau}_{i,j} \tag{2.36}$$

where:

$$\boldsymbol{G} = \sum_{k=1}^{K} (\boldsymbol{F}^{(k)})^{\mathsf{T}} \boldsymbol{F}^{(k)} \tag{2.37}$$

is a (5×5) symmetric coefficient matrix, and where $\boldsymbol{F}^{(k)}$ is a $(N^2 \times 5)$ matrix defined as follows:

$$\boldsymbol{F}^{(k)} = \left[\text{vec}(\boldsymbol{F}^{(k,4)}), \text{vec}(\boldsymbol{F}^{(k,3)}), \text{vec}(\boldsymbol{F}^{(k,2)}), \text{vec}(\boldsymbol{F}^{(k,1)}), \text{vec}(\boldsymbol{F}^{(k,0)}) \right] \tag{2.38}$$

in which vec(.) reshapes a matrix into a column vector by stacking its columns vertically. $\boldsymbol{\tau}_{i,j}$ is a 5-dimensional parameter vector defined as follows:

$$\boldsymbol{\tau}_{i,j} = \left[t_{i,j}^4,\, t_{i,j}^3,\, t_{i,j}^2,\, t_{i,j},\, 1 \right]^{\mathsf{T}} \tag{2.39}$$

Cost function (2.36) shows that $\Psi^{+}_{3,\text{AC}}(t_{i,j})$ is a 8-th degree polynomial in $t_{i,j}$. The global minimum $t_{i,j}$ can be obtained by computing the roots of its derivative and selecting the root yielding the smallest value of (2.36). Once the optimal $t_{i,j}$ is computed, the matrix $\widehat{\boldsymbol{B}}^{(\text{new})}$ is computed using equation (2.27), and $\widehat{\boldsymbol{A}}^{(\text{new})}$ is given by $\widehat{\boldsymbol{A}}^{(\text{new})} = (\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}$. Then the AC phase is repeated to estimate $\boldsymbol{T}^{(i,j)}(t_{i,j})$ with the next (i,j) index.

2.2.1.2 DC phase

In this phase, the minimization of (2.24) with respect to the diagonal matrices $\{\boldsymbol{D}^{(k)}\}$ can be separated into K distinct linear least square subproblems:

$$\Psi^{(k)}_{3,\text{DC}}(\boldsymbol{D}^{(k)}) = \left\| \boldsymbol{C}^{(k)} - \widehat{\boldsymbol{A}} \boldsymbol{D}^{(k)} \widehat{\boldsymbol{A}}^{\mathsf{T}} \right\|_F^2, \quad k \in \{1, 2, \ldots, K\} \tag{2.40}$$

The optimal solution of $\widehat{\boldsymbol{D}}^{(k,\text{new})}$ given by Yeredor in [Yeredor, 2002] is:

$$\widehat{\boldsymbol{D}}^{(k,\text{new})} = \text{Diag}\left((\widehat{\boldsymbol{A}} \odot \widehat{\boldsymbol{A}})^{\sharp} \boldsymbol{c}^{(k)} \right) \tag{2.41}$$

where $\text{Diag}(\cdot)$ returns a diagonal matrix whose diagonal elements are given by the input vector, where $\boldsymbol{c}^{(k)} = \text{vec}(\boldsymbol{C}^{(k)})$, and where $\widehat{\boldsymbol{A}}$ is the current estimate of the joint transformation matrix $\boldsymbol{A}$.

In this context, one full iteration is referred to the combination of one full AC phase and one DC phase. In practice, several iterations are necessary to ensure convergence. One can stop the algorithm when the value of the global cost function (2.22), or the decrease of (2.22) between two successive iterations falls below a fixed positive threshold. Such a stopping criterion is guaranteed to be met since (2.22) is non-increasing in each sweep. We observed empirically that the proposed algorithm converges linearly.

2.2.2 Generalization to the non-square case

In a JDC problem, a non-square joint transformation matrix $\boldsymbol{A} \in \mathbb{R}_{+}^{N \times P}$ and K diagonal matrices $\boldsymbol{D}^{(k)} \in \mathbb{R}^{P \times P}$ with $N > P$ are frequently encountered. Let us recall that $\boldsymbol{A} = \boldsymbol{B}^{\boxdot 2}$. Based on the LU decomposition of $\boldsymbol{B} \in \mathbb{R}^{N \times P}$, which yields $\boldsymbol{B} = \boldsymbol{L}\boldsymbol{U}$, where $\boldsymbol{L} \in \mathbb{R}^{N \times P}$ is a unit lower triangular matrix, and where $\boldsymbol{U} \in \mathbb{R}^{P \times P}$ is a unit upper triangular matrix, the $\text{JDC}^{+}_{\text{LU-3}}$ algorithm can handle non-square matrices directly. Theoretically, $\boldsymbol{L}$ and $\boldsymbol{U}$ can be decomposed as follows:

$$\boldsymbol{L} = \prod_{j=1}^{P} \prod_{i=j+1}^{N} \boldsymbol{L}^{(i,j)}(\ell_{i,j}), \quad \boldsymbol{U} = \prod_{i=1}^{P} \prod_{j=i+1}^{P} \boldsymbol{U}^{(i,j)}(u_{i,j}) \tag{2.42}$$

where the sizes of all the $\boldsymbol{L}^{(i,j)}(\ell_{i,j})$ and $\boldsymbol{U}^{(i,j)}(u_{i,j})$ matrices are $(N \times N)$ and $(P \times P)$, respectively, except that of $\boldsymbol{L}^{(N,P)}(\ell_{N,P})$ which is $(N \times P)$. Below we give an example:

$$\begin{pmatrix} 1.0000 & 0.1176 \\ 0.3750 & 1.0441 \\ 0.5000 & 1.8966 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0.3750 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.5 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1.8378 \end{pmatrix} \begin{pmatrix} 1 & 0.1176 \\ 0 & 1 \end{pmatrix}$$

Indeed, we can treat all the $\boldsymbol{L}^{(i,j)}(\ell_{i,j})$ and $\boldsymbol{U}^{(i,j)}(u_{i,j})$ matrices as $(N \times N)$ matrices, and choose the first P column vectors of their product as the final result. In the above example, we actually estimate four (3×3) elementary triangular matrices and choose the first two column vectors of the product as the result:

$$\begin{pmatrix} 1.0000 & 0.1176 & 0 \\ 0.3750 & 1.0441 & 0 \\ 0.5000 & 1.8966 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0.3750 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.5 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1.8378 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0.1176 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

2.2.3 Practical issues

In practice, we observe that if each input matrix $\boldsymbol{C}^{(k)}$ is almost exactly jointly diagonalizable due to a high SNR, the classical non-constrained JDC methods can also give a nonnegative $\boldsymbol{A}$ with high probability. In this situation, imposing the nonnegativity constraint explicitly is unnecessary and increases the computational burden. Therefore, we propose to relax the nonnegativity constraint by directly decomposing $\boldsymbol{A}$ into elementary LU form instead of using the decompositions of $\boldsymbol{B}$ as follows:

$$\boldsymbol{A} = \prod_{j=1}^{N} \prod_{i=j+1}^{N} \boldsymbol{L}^{(i,j)}(\ell_{i,j}) \prod_{i=1}^{N} \prod_{j=i+1}^{N} \boldsymbol{U}^{(i,j)}(u_{i,j}) \overset{\text{def}}{=} \prod_{i \neq j}^{N(N-1)} \boldsymbol{T}^{(i,j)}(t_{i,j}) \tag{2.43}$$

By inserting (2.43) into the original direct-fit JDC criterion (1.29), each parameter $t_{i,j}$ can be found by minimizing the following 4-th degree polynomial:

$$\Psi_{3,\text{AC}}(t_{i,j}) = \sum_{k=1}^{K} \left\| \boldsymbol{F}^{(k,4)} t_{i,j}^2 + (\boldsymbol{F}^{(k,2)} + 4\widehat{d}_j^{(k)} (\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)(\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)^{\mathsf{T}}) t_{i,j} + \boldsymbol{F}^{(k,0)} \right\|_F^2 \quad (2.44)$$

The derivation of the compact matrix form of (2.44) is omitted in this report. In practice, it is suggested to compute $t_{i,j}$ by minimizing (2.44) first. If all the elements in the j-th column vector of $\widehat{\boldsymbol{A}}\boldsymbol{T}^{(i,j)}(t_{i,j})$ have the same sign ς, the j-th column vector of $\widehat{\boldsymbol{A}}^{(\text{new})}$ is replaced by $\varsigma(\widehat{\boldsymbol{a}}_j + t_{i,j}\widehat{\boldsymbol{a}}_i)$, while the other column vectors remain unchanged, where $\widehat{\boldsymbol{a}}_i$ and $\widehat{\boldsymbol{a}}_j$ are the i-th and j-th column vectors of $\widehat{\boldsymbol{A}}$, respectively. Otherwise, $t_{i,j}$ is computed by minimizing (2.36). $\widehat{\boldsymbol{A}}$ is then updated using proposition 2.1. The pseudo-code of JDC$^{+}_{\text{LU-3}}$ is given in the appendix C.2 .

2.3 The iJDC$^{+}_{\text{LU-1}}$ and iJDC$^{+}_{\text{QR-1}}$ algorithms

2.3.1 Algorithm derivation

The JDC$^{+}_{\text{LU-3}}$ algorithm is conceptually simple. However, the alternating nature of such a method leads to some drawbacks. For example, its performance appears to depend on initialization, therefore in practice it requires a multi-initialization procedure, leading to an increase of numerical complexity. Moreover, similar to the ACDC algorithm, JDC$^{+}_{\text{LU-3}}$ converges linearly, and occasionally its convergence can be slow. In fact, in many practical ICA problems, the estimate of these diagonal matrices $\boldsymbol{D}^{(k)}$ is not necessary. Therefore, without estimating $\boldsymbol{D}^{(k)}$, we propose to use the indirect-fit criterion (1.27) in order to derive a *pure* Jacobi-like NJDC algorithm, inspired by the fast convergence property of this kind of procedure. Furthermore, we would like to use both LU and QR matrix factorizations for the Jacobi-like procedure in order to study the influence of the used matrix factorization. We also would like to avoid the inverse operator in criterion (1.27). Now let us consider the structure of $\boldsymbol{C}^{(k)} = \boldsymbol{A}\boldsymbol{D}^{(k)}\boldsymbol{A}^{\mathsf{T}}$ with the following two assumptions:

1. $\boldsymbol{A} \in \mathbb{R}_{+}^{N\times N}$ is nonsingular;
2. Each $\boldsymbol{D}^{(k)} \in \mathbb{R}^{N\times N}$ is nonsingular which means that its main diagonal does not contain any zero entry.

Then each matrix $\boldsymbol{C}^{(k)}$ is nonsingular and its inverse can be expressed as follows:

$$\left(\boldsymbol{C}^{(k)}\right)^{-1} = \boldsymbol{A}^{-\mathsf{T}} \left(\boldsymbol{D}^{(k)}\right)^{-1} \boldsymbol{A}^{-1} \quad (2.45)$$

We use $\boldsymbol{C}^{(k,-1)}$ to denote $(\boldsymbol{C}^{(k)})^{-1}$ for simplicity. Equation (2.45) shows that $\boldsymbol{C}^{(k,-1)}$ also preserves the jointly diagonalizable structure. Furthermore, instead of $\boldsymbol{A}^{-1}$, $\boldsymbol{A}$ serves as the joint diagonalizer. As a consequence, by means of a preprocessing step that inverses each input matrices $\boldsymbol{C}^{(k)}$, $\boldsymbol{A}$ can be estimated by minimizing the following criterion which is modified from (1.27):

$$\Psi_{1'}(\boldsymbol{A}) = \sum_{k=1}^{K} \left\| \operatorname{off}(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{C}^{(k,-1)}\boldsymbol{A}) \right\|_F^2 \quad (2.46)$$

By such a manipulation, most JDC algorithms based on criterion (1.27) can now estimate $\boldsymbol{A}$ directly. However, none of them can guarantee the nonnegativity of $\boldsymbol{A}$. In order to impose the nonnegativity constraint on $\boldsymbol{A}$, we resort to use a square change of variable $\boldsymbol{A} = \boldsymbol{B}^{\boxdot 2}$. Then cost function (2.46) can be reformulated including the nonnegativity constraint as follows:

$$\Psi^{+}_{1'}(\boldsymbol{B}) = \sum_{k=1}^{K} \left\| \operatorname{off}\left((\boldsymbol{B}^{\boxdot 2})^{\mathsf{T}} \boldsymbol{C}^{(k,-1)} \boldsymbol{B}^{\boxdot 2} \right) \right\|_F^2 \tag{2.47}$$

By means of the LU and QR matrix factorizations of $\boldsymbol{B}$, the minimization of (2.47) with respect to $\boldsymbol{B}$ is converted to the estimate of the following two sets of $N(N-1)$ parameters, respectively,

1. $\ell_{i,j}$ and $u_{i,j}$ for the LU decomposition (2.9);
2. $\theta_{i,j}$ and $u'_{i,j}$ for the QR decomposition (2.10).

Instead of simultaneously computing these parameters, we propose two Jacobi-like procedures which perform a series of $N(N-1)$ sequential optimizations. This yields two new algorithms: *i*) the first algorithm, named iJDC$^{+}_{\text{LU-1}}$, estimates each $\ell_{i,j}$ and $u_{i,j}$ successively, and *ii*) the second one, called iJDC$^{+}_{\text{QR-1}}$, estimates each $\theta_{i,j}$ and $u'_{i,j}$ sequentially. Now, the difficulty is how to estimate these of parameters, namely $\boldsymbol{L}^{(i,j)}(\ell_{i,j})$ and $\boldsymbol{U}^{(i,j)}(u_{i,j})$ for iJDC$^{+}_{\text{LU-1}}$, and $\boldsymbol{Q}^{(i,j)}(\theta_{i,j})$ and $\boldsymbol{U}^{(i,j)}(u'_{i,j})$ for iJDC$^{+}_{\text{QR-1}}$. Two points should be noted here:

1. For the iJDC$^{+}_{\text{LU-1}}$ algorithm, $\boldsymbol{L}^{(i,j)}(\ell_{i,j})$ and $\boldsymbol{U}^{(i,j)}(u_{i,j})$ belong to the same category of matrices, therefore they can be estimated by the same algorithmic procedure just with an emphasis on the relation between the i and j indices ($j < i$ for $\boldsymbol{L}^{(i,j)}(\ell_{i,j})$ and $i < j$ for $\boldsymbol{U}^{(i,j)}(u_{i,j})$). Hence, we adopt the notation $\boldsymbol{T}^{(i,j)}(t_{i,j})$ to denote both $\boldsymbol{L}^{(i,j)}(\ell_{i,j})$ and $\boldsymbol{U}^{(i,j)}(u_{i,j})$, as we defined it earlier in equation (2.26).
2. Both iJDC$^{+}_{\text{LU-1}}$ and iJDC$^{+}_{\text{QR-1}}$ algorithms consist of estimating the elementary triangular matrices $\boldsymbol{U}^{(i,j)}(u_{i,j})$. But the resulting parameters could be different. Additionally the procedures of estimating these matrices for both algorithms are identical.

Consequently, the principal problem is reduced to the estimation of two kinds of parameters, namely the elementary triangular matrices $\boldsymbol{T}^{(i,j)}(t_{i,j})$ and the Givens rotation matrices $\boldsymbol{Q}^{(i,j)}(\theta_{i,j})$, by means of Jacobi-like optimization procedures.

2.3.1.1 Minimization with respect to $\boldsymbol{T}^{(i,j)}(t_{i,j})$

In this section, we estimate $\boldsymbol{T}^{(i,j)}(t_{i,j})$ for $i \neq j$ by solving the following optimization problem:

$$t_{i,j} = \underset{t_{i,j}}{\operatorname{argmin}}\, \Psi^{+}_{1'}(\widehat{\boldsymbol{B}}\boldsymbol{T}^{(i,j)}(t_{i,j})) \tag{2.48}$$

The update of $\widehat{\boldsymbol{B}}$ is defined in equation (2.27) by $\widehat{\boldsymbol{B}}^{(\text{new})} = \widehat{\boldsymbol{B}}\boldsymbol{T}^{(i,j)}(t_{i,j})$. Inserting $\widehat{\boldsymbol{B}}^{(\text{new})}$ into the optimization (2.48), the optimal $t_{i,j}$ is the global minimum of the following local cost function:

$$\Psi^{+}_{1'}(t_{i,j}) = \sum_{k=1}^{K} \left\| \operatorname{off}\left\{ [(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}]^{\mathsf{T}} \boldsymbol{C}^{(k,-1)} [(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}] \right\} \right\|_F^2 \tag{2.49}$$

We define the K matrices $\widehat{\boldsymbol{C}}^{(k,\text{new})} \in \mathbb{R}^{N\times N}$ as follows:

$$\widehat{\boldsymbol{C}}^{(k,\text{new})} = \left[(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}\right]^{\mathsf{T}} \boldsymbol{C}^{(k,-1)} \left[(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}\right] \quad k \in \{1,2,\ldots,K\} \tag{2.50}$$

The cost function (2.49) actually minimizes the total sum of squares of the off-diagonal components of $\widehat{\boldsymbol{C}}^{(k,\text{new})}$. The explicit expression of $(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}$ as a function of $t_{i,j}$ is provided in (2.29) in proposition 2.1. Inserting equation (2.29) into (2.50), we have:

$$\begin{aligned}\widehat{\boldsymbol{C}}^{(k,\text{new})} =& \underbrace{\boldsymbol{T}^{(i,j)}(t_{i,j}^2)^{\mathsf{T}}\widehat{\boldsymbol{C}}^{(k)}\,\boldsymbol{T}^{(i,j)}(t_{i,j}^2)}_{①} \\ &+ \underbrace{t_{i,j}\,\boldsymbol{T}^{(i,j)}(t_{i,j}^2)^{\mathsf{T}}\widehat{\boldsymbol{c}}^{(k,1)}\boldsymbol{e}_j^{\mathsf{T}}}_{②} + \underbrace{t_{i,j}\,\boldsymbol{e}_j\widehat{\boldsymbol{c}}^{(k,2)}\boldsymbol{T}^{(i,j)}(t_{i,j}^2)}_{③} + \underbrace{t_{i,j}^2\,\widehat{c}^{(k,3)}\boldsymbol{e}_j\boldsymbol{e}_j^{\mathsf{T}}}_{④}\end{aligned} \tag{2.51}$$

where $\widehat{\boldsymbol{C}}^{(k)} = \widehat{\boldsymbol{A}}^{\mathsf{T}}\boldsymbol{C}^{(k,-1)}\widehat{\boldsymbol{A}}$ is a $(N\times N)$ constant matrix, where $\widehat{\boldsymbol{c}}^{(k,1)} = 2\,\widehat{\boldsymbol{A}}^{\mathsf{T}}\boldsymbol{C}^{(k,-1)}(\widehat{\boldsymbol{b}}_i\boxdot\widehat{\boldsymbol{b}}_j)$ is a $(N\times 1)$ constant column vector, where $\widehat{\boldsymbol{c}}^{(k,2)} = 2\,(\widehat{\boldsymbol{b}}_i\boxdot\widehat{\boldsymbol{b}}_j)^{\mathsf{T}}\boldsymbol{C}^{(k,-1)}\widehat{\boldsymbol{A}}$ is a $(1\times N)$ constant row vector, and where $\widehat{c}^{(k,3)} = 4\,(\widehat{\boldsymbol{b}}_i\boxdot\widehat{\boldsymbol{b}}_j)^{\mathsf{T}}\boldsymbol{C}^{(k,-1)}(\widehat{\boldsymbol{b}}_i\boxdot\widehat{\boldsymbol{b}}_j)$ is a constant scalar. The term ① in (2.51) transforms the j-th column vector and the j-th row vector of $\widehat{\boldsymbol{C}}^{(k)}$. The term ② in (2.51) is a zero matrix except its j-th column vector containing non-zero elements, while the term ③ in (2.51) contains non-zero entries only on its j-th row vector. Eventually the term ④ in (2.51) is a zero matrix except its (j,j)-th component being non-zero. In addition, $\widehat{\boldsymbol{C}}^{(k,\text{new})} = ①+②+③+④$ is a $(N\times N)$ symmetric matrix. Hence (2.51) shows that only the j-th column vector and the j-th row vector of $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ involve the parameter $t_{i,j}$, while the other components remain constant. Therefore, considering the symmetry of $\widehat{\boldsymbol{C}}^{(k,\text{new})}$, the minimization of the cost function (2.49) is equivalent to minimizing the total sum of the squares of the elements of the j-th column vectors of $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ except their (j,j)-th elements with $k \in \{1,\cdots,K\}$. The required components of $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ can be expressed by the following proposition.

Proposition 2.2. *The elements of the j-th column vector except the (j,j)-th entry of $\widehat{\boldsymbol{C}}^{(k,new)}$ is a second degree polynomial function in $t_{i,j}$ as follows, for every n value different of j:*

$$\widehat{C}_{n,j}^{(k,new)} = \widehat{C}_{n,i}^{(k)}t_{i,j}^2 + \widehat{c}_n^{(k,1)}t_{i,j} + \widehat{C}_{n,j}^{(k)} \tag{2.52}$$

where $\widehat{C}_{n,i}^{(k)}$ and $\widehat{C}_{n,j}^{(k)}$ are the (n,i)-th and (n,j)-th components of matrix $\widehat{\boldsymbol{C}}^{(k)}$, respectively, and $\widehat{c}_n^{(k,1)}$ is the n-th element of vector $\widehat{\boldsymbol{c}}^{(k,1)}$.

The proof of this proposition is addressed in appendix A.2. Proposition 2.2 shows that the minimization of the cost function (2.49) can be expressed in the following compact matrix form:

$$\Psi_{1'}^{+}(t_{i,j}) = \sum_{k=1}^{K}\left\|\,\boldsymbol{E}^{(k)}\boldsymbol{\tau}_{i,j}\,\right\|_F^2 = \boldsymbol{\tau}_{i,j}^{\mathsf{T}}\,\boldsymbol{Q}_E\,\boldsymbol{\tau}_{i,j} \tag{2.53}$$

where $\boldsymbol{Q}_E$ is defined as a (3×3) symmetric coefficient matrix:

$$\boldsymbol{Q}_E = \sum_{k=1}^{K}(\boldsymbol{E}^{(k)})^{\mathsf{T}}\boldsymbol{E}^{(k)} \tag{2.54}$$

with $\boldsymbol{E}^{(k)}$ a $((N-1)\times 3)$ matrix defined as follows: its first column contains the i-th column of $\widehat{\boldsymbol{C}}^{(k)}$ without the j-th element, the second column contains vector $\widehat{\boldsymbol{c}}^{(k,1)}$ without the j-th entry and the third column contains the j-th column of $\widehat{\boldsymbol{C}}^{(k)}$ without the j-th component. $\boldsymbol{\tau}_{i,j}$ is a 3-dimensional parameter vector defined as follows:

$$\boldsymbol{\tau}_{i,j} = \left[t_{i,j}^2,\, t_{i,j},\, 1\right]^{\mathsf{T}} \tag{2.55}$$

Equation (2.53) shows that $\Psi^{+}_{1'}(t_{i,j})$ is a fourth degree polynomial function. The global minimum $t_{i,j}$ can be obtained by computing the roots of its derivative and selecting the one yielding the smallest value of (2.53). Once the optimal $t_{i,j}$ is computed, $\widehat{\boldsymbol{B}}^{(\text{new})}$ is updated by means of equation (2.27) and the joint diagonalizer $\widehat{\boldsymbol{A}}$ is updated by computing $(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}$. The K matrices $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ are computed by means of equation (2.51) and are used as the input matrices for the next Jacobi-like iteration. Then the same procedure is repeated to compute the next $t_{i,j}$ with another index couple (i,j). The processing of all the $N(N-1)$ parameters $t_{i,j}$ with different (i,j) couples, more precisely $\ell_{i,j}$ with $i>j$ and $u_{i,j}$ with $i<j$, is called a LU sweep. The proposed iJDC$^{+}_{\text{LU-1}}$ algorithm requires several LU sweeps in practice.

2.3.1.2 Minimization with respect to $\boldsymbol{Q}^{(i,j)}(\theta_{i,j})$

Now we identify $\boldsymbol{Q}^{(i,j)}(\theta_{i,j})$ with $i<j$ and $i,j\in\{1,\ldots,N\}$ by solving the following optimization problem:

$$\theta_{i,j} = \underset{\theta_{i,j}}{\operatorname{argmin}}\, \Psi^{+}_{1'}(\widehat{\boldsymbol{B}}\boldsymbol{Q}^{(i,j)}(\theta_{i,j})) \tag{2.56}$$

The update of $\widehat{\boldsymbol{B}}$ is now defined as follows:

$$\widehat{\boldsymbol{B}}^{(\text{new})} = \widehat{\boldsymbol{B}}\boldsymbol{Q}^{(i,j)}(\theta_{i,j}) \tag{2.57}$$

Similarly, putting (2.57) into the optimization (2.56), the optimal $\theta_{i,j}$ is the global minimum of the following local cost function:

$$\Psi^{+}_{1'}(\theta_{i,j}) = \sum_{k=1}^{K} \left\| \operatorname{off}\left\{ [(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}]^{\mathsf{T}} \boldsymbol{C}^{(k,-1)} [(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}] \right\} \right\|_F^2 \tag{2.58}$$

We define the K matrices $\widehat{\boldsymbol{C}}^{(k,\text{new})} \in \mathbb{R}^{N\times N}$ as follows:

$$\widehat{\boldsymbol{C}}^{(k,\text{new})} = \left[(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}\right]^{\mathsf{T}} \boldsymbol{C}^{(k,-1)} \left[(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}\right], \quad k\in\{1,2,\ldots,K\} \tag{2.59}$$

The cost function (2.58) minimizes the total sum of squares of the off-diagonal components of $\widehat{\boldsymbol{C}}^{(k,\text{new})}$. The explicit expression of $(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}$ as a function of $\theta_{i,j}$ now can be rewritten as shown in the following proposition.

Proposition 2.3. *$\widehat{\boldsymbol{A}}^{(new)} = (\widehat{\boldsymbol{B}}^{(new)})^{\boxdot 2} = (\widehat{\boldsymbol{B}}\boldsymbol{Q}^{(i,j)}(\theta_{i,j}))^{\boxdot 2}$ can be written as a function of $\theta_{i,j}$ as follows:*

$$\widehat{\boldsymbol{A}}^{(new)} = (\widehat{\boldsymbol{B}}^{(new)})^{\boxdot 2} = \widehat{\boldsymbol{B}}^{\boxdot 2}\,(\boldsymbol{Q}^{(i,j)}(\theta_{i,j}))^{\boxdot 2} + \sin(2\theta_{i,j})(\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)(\boldsymbol{e}_i^{\mathsf{T}} - \boldsymbol{e}_j^{\mathsf{T}}) \tag{2.60}$$

where $\widehat{\boldsymbol{b}}_i$ and $\widehat{\boldsymbol{b}}_j$ denote the i-th and j-th column vectors of $\widehat{\boldsymbol{B}}$, respectively, and where $\boldsymbol{e}_i$ and $\boldsymbol{e}_j$ are the i-th and j-th column vectors of the identity matrix $\boldsymbol{I}_N$, respectively.

The proof of proposition 2.3 is provided in appendix A.3. Inserting (2.60) into (2.59), we obtain:

$$\begin{aligned}\widehat{\boldsymbol{C}}^{(k,\text{new})} =& \underbrace{[(\boldsymbol{Q}^{(i,j)}(\theta_{i,j}))^{\boxdot 2}]^{\mathsf{T}}\widehat{\boldsymbol{C}}^{(k)}(\boldsymbol{Q}^{(i,j)}(\theta_{i,j}))^{\boxdot 2}}_{①} + \underbrace{\sin(2\theta_{i,j})[(\boldsymbol{Q}^{(i,j)}(\theta_{i,j}))^{\boxdot 2}]^{\mathsf{T}}\widehat{\boldsymbol{c}}^{(k,1)}(\boldsymbol{e}_i^{\mathsf{T}} - \boldsymbol{e}_j^{\mathsf{T}})}_{②} \\ &+ \underbrace{\sin(2\theta_{i,j})(\boldsymbol{e}_i - \boldsymbol{e}_j)\widehat{\boldsymbol{c}}^{(k,2)}(\boldsymbol{Q}^{(i,j)}(\theta_{i,j}))^{\boxdot 2}}_{③} + \underbrace{\sin^2(2\theta_{i,j})\widehat{c}^{(k,3)}(\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i^{\mathsf{T}} - \boldsymbol{e}_j^{\mathsf{T}})}_{④}\end{aligned} \tag{2.61}$$

where $\widehat{\boldsymbol{C}}^{(k)} = \widehat{\boldsymbol{A}}^{\mathsf{T}}\boldsymbol{C}^{(k,-1)}\widehat{\boldsymbol{A}}$ is a $(N \times N)$ constant matrix, where $\widehat{\boldsymbol{c}}^{(k,1)} = \widehat{\boldsymbol{A}}^{\mathsf{T}}\boldsymbol{C}^{(k,-1)}(\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)$ is a $(N \times 1)$ constant column vector, where $\widehat{\boldsymbol{c}}^{(k,2)} = (\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)^{\mathsf{T}}\boldsymbol{C}^{(k,-1)}\widehat{\boldsymbol{A}}$ is a $(1 \times N)$ constant row vector, and where $\widehat{c}^{(k,3)} = (\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)^{\mathsf{T}}\boldsymbol{C}^{(k,-1)}(\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)$ is a constant scalar. The term ① in (2.61) transforms the i-th and j-th column vectors and the i-th and j-th row vectors of $\widehat{\boldsymbol{C}}^{(k)}$. The term ② in (2.61) is a zero matrix except its i-th and j-th column vectors containing non-zero elements, while the term ③ in (2.61) contains non-zero entries only on its i-th and j-th row vectors. And the term ④ in (2.61) is a zero matrix except its (i,i)-th, (j,j)-th, (i,j)-th and (j,i)-th components being non-zero. $\widehat{\boldsymbol{C}}^{(k,\text{new})} =$ ①+②+③+④ is a $(N \times N)$ symmetric matrix. Hence (2.61) shows that only the i-th and j-th column vectors and the i-th and j-th row vectors of $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ involve the parameter $\theta_{i,j}$, while the other components remain constant. It is noteworthy that the (i,j)-th and (j,i)-th components are twice affected by the transformation. Considering the symmetry of $\widehat{\boldsymbol{C}}^{(k,\text{new})}$, we propose to minimize the sum of the squares of the (i,j)-th entries of the K matrices $\widehat{\boldsymbol{C}}^{(k,\text{new})}$, instead of minimizing all the off-diagonal entries. Although minimizing this quantity is not equivalent to minimizing the original cost function (2.58), such a simplified minimization scheme is commonly adopted by many algorithms, such as in [Wang et al., 2007, Souloumiac, 2009, Guo et al., 2010, Luciani and Albera, 2014]. Therefore, $u_{i,j}$ is estimated by minimizing an approximate of cost function (2.58), denoted by $\widehat{\Psi}^{+}_{1'}(\theta_{i,j})$. The (i,j)-th component of $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ is expressed in the following proposition.

Proposition 2.4. *The (i,j)-th entry of $\widehat{\boldsymbol{C}}^{(k,new)}$ can be expressed as a function of $\theta_{i,j}$ as follows:*

$$\begin{aligned}\widehat{C}_{i,j}^{(k,new)} =& -\sin^2(2\theta_{i,j})\widehat{c}^{(k,3)} \\ &+ \sin^2(\theta_{i,j})(\widehat{C}_{i,i}^{(k)}\cos^2(\theta_{i,j}) + \widehat{C}_{j,i}^{(k)}\sin^2(\theta_{i,j})) \\ &+ \cos^2(\theta_{i,j})(\widehat{C}_{i,j}^{(k)}\cos^2(\theta_{i,j}) + \widehat{C}_{j,j}^{(k)}\sin^2(\theta_{i,j})) \\ &+ \sin(2\theta_{i,j})(\widehat{c}_i^{(k,1)}\cos^2(\theta_{i,j}) + \widehat{c}_j^{(k,1)}\sin^2(\theta_{i,j})) \\ &- \sin(2\theta_{i,j})(\widehat{c}_j^{(k,2)}\cos^2(\theta_{i,j}) + \widehat{c}_i^{(k,2)}\sin^2(\theta_{i,j}))\end{aligned} \tag{2.62}$$

where $\widehat{C}_{i,i}^{(k)}$, $\widehat{C}_{j,j}^{(k)}$, $\widehat{C}_{i,j}^{(k)}$ and $\widehat{C}_{j,i}^{(k)}$ are the (i,i)-th, (j,j)-th, (i,j)-th and (j,i)-th components of matrix $\widehat{\boldsymbol{C}}^{(k)}$, respectively, and where $\widehat{c}_i^{(k,q)}$ and $\widehat{c}_j^{(k,q)}$ are the i-th and j-th elements of vector $\widehat{\boldsymbol{c}}^{(k,q)}$ with $q \in \{1,2\}$, respectively.

The proof of proposition 2.4 is given in appendix A.4. In order to simplify the notation of (2.62), we resort to the Weierstrass change of variable: $\vartheta_{i,j} = \tan(\theta_{i,j})$. Then we obtain:

$$\sin(2\theta_{i,j}) = \frac{2\vartheta_{i,j}}{1+\vartheta_{i,j}^2},\ \cos(2\theta_{i,j}) = \frac{1-\vartheta_{i,j}^2}{1+\vartheta_{i,j}^2},\ \sin^2(\theta_{i,j}) = \frac{\vartheta_{i,j}^2}{1+\vartheta_{i,j}^2},\ \cos^2(\theta_{i,j}) = \frac{1}{1+\vartheta_{i,j}^2} \tag{2.63}$$

By substituting (2.63) into (2.62), we obtain an alternative expression of the (i,j)-th entry of $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ which is described in the following proposition.

Proposition 2.5. *The (i,j)-th entry of $\widehat{\boldsymbol{C}}^{(k,new)}$ can be expressed by a rational function of $\vartheta_{i,j}$ as follows:*

$$\widehat{C}_{i,j}^{(k,new)} = \frac{f_4^{(k)}\vartheta_{i,j}^4 + f_3^{(k)}\vartheta_{i,j}^3 + f_2^{(k)}\vartheta_{i,j}^2 + f_1^{(k)}\vartheta_{i,j} + f_0^{(k)}}{(1+\vartheta_{i,j}^2)^2} \tag{2.64}$$

where $f_4^{(k)} = \widehat{C}_{j,i}^{(k)}$, $f_3^{(k)} = -2\widehat{c}_i^{(k,1)}$, $f_2^{(k)} = \widehat{C}_{i,i}^{(k)} + \widehat{C}_{j,j}^{(k)} + 2\widehat{c}_j^{(k,2)} - 4\widehat{c}^{(k,3)}$, $f_1^{(k)} = 2\widehat{c}_i^{(k,2)} - \widehat{c}_j^{(k,1)}$ and $f_0^{(k)} = \widehat{C}_{j,j}^{(k)}$.

The proof of proposition 2.5 is quite easy and therefore omitted. By means of proposition 2.5, the approximate cost function $\widehat{\Psi}^{+}_{1'}(\theta_{i,j})$ is then transformed into $\widehat{\Psi}^{+}_{1'}(\vartheta_{i,j})$. Recall that $\widehat{\Psi}^{+}_{1'}(\vartheta_{i,j})$ is the total sum of the squares of the (i,j)-th entries of the K matrices $\widehat{\boldsymbol{C}}^{(k,\text{new})}$. Moreover equation (2.64) easily shows that $\widehat{\Psi}^{+}_{1'}(\vartheta_{i,j})$ is a rational function in $\vartheta_{i,j}$, where the degrees of the numerator and the denominator are 8 and 8, respectively. $\widehat{\Psi}^{+}_{1'}(\vartheta_{i,j})$ can be expressed in the following compact matrix form:

$$\widehat{\Psi}^{+}_{1'}(\vartheta_{i,j}) = \sum_{k=1}^{K} \left\| (\boldsymbol{f}^{(k)})^{\mathsf{T}} \boldsymbol{\vartheta}_{i,j} \right\|_F^2 = \boldsymbol{\vartheta}_{i,j}^{\mathsf{T}} \boldsymbol{Q}_F \boldsymbol{\vartheta}_{i,j} \tag{2.65}$$

where $\boldsymbol{Q}_F$ is a (5×5) symmetric coefficient matrix defined as follows:

$$\boldsymbol{Q}_F = \sum_{k=1}^{K} \boldsymbol{f}^{(k)} (\boldsymbol{f}^{(k)})^{\mathsf{T}} \tag{2.66}$$

where $\boldsymbol{f}^{(k)}$ is a 5-dimensional vector:

$$\boldsymbol{f}^{(k)} = \left[f_4^{(k)},\, f_3^{(k)},\, f_2^{(k)},\, f_1^{(k)},\, f_0^{(k)} \right]^{\mathsf{T}} \tag{2.67}$$

and where $\boldsymbol{\vartheta}_{i,j}$ is a 5-dimensional parameter vector defined as follows:

$$\boldsymbol{\vartheta}_{i,j} = \frac{1}{(1+\vartheta_{i,j}^2)^2} \left[\vartheta_{i,j}^4,\, \vartheta_{i,j}^3,\, \vartheta_{i,j}^2,\, \vartheta_{i,j},\, 1 \right]^{\mathsf{T}} \tag{2.68}$$

The global minimum $\vartheta_{i,j}$ can be obtained by computing the roots of the derivative of the cost function(2.65) and selecting the one yielding the smallest value of (2.65). Once $\vartheta_{i,j}$ is obtained, $\theta_{i,j}$ can be computed from the inverse tangent function $\theta_{i,j} = \arctan(\vartheta_{i,j})$. It is noteworthy that the found $\theta_{i,j}$ cannot guarantee to decrease the original cost function (2.58). If $\theta_{i,j}$ leads to an increase of (2.58), we reset $\theta_{i,j} = 0$, thus the value of the original cost function remains unchanged. Otherwise, $\widehat{\boldsymbol{B}}$ is updated as described in (2.57) and the joint diagonalizer $\widehat{\boldsymbol{A}}$ is updated by computing $(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}$. The same procedure will be repeated to compute $\theta_{i,j}$ with the next (i,j) couple. The processing of all the $N(N-1)/2$ parameters $\theta_{i,j}$ and also the other $N(N-1)/2$ parameters $u_{i,j}$ is called a QR sweep. Several QR sweeps yield the proposed iJDC$^{+}_{\text{QR-1}}$ algorithm.

Both the iJDC$^{+}_{\text{LU-1}}$ and iJDC$^{+}_{\text{QR-1}}$ algorithms can be stopped when the value of the global cost function (2.47) or its relative change between two successive sweeps fall below a fixed small positive threshold. Such a stopping criterion is guaranteed to be met since the cost function is non-increasing in each Jacobi-like sweep.

2.3.2 Practical issues

2.3.2.1 Nonnegativity relaxation

As we discussed in section 2.2.3, if each input matrix $\boldsymbol{C}^{(k)}$ is almost exactly jointly diagonalizable due to a high SNR value, the explicit nonnegativity constraint by means of a square change of variable could be unnecessary. Consequently, the nonnegativity constraint can be relaxed by directly decomposing $\boldsymbol{A}$ using the LU factorization in equation (2.43) or the QR factorization as follows:

$$\boldsymbol{A} = \prod_{i=1}^{N}\prod_{j=i+1}^{N} \boldsymbol{Q}^{(i,j)}(\theta_{i,j}) \prod_{i=1}^{N}\prod_{j=i+1}^{N} \boldsymbol{U}^{(i,j)}(u'_{i,j}) \overset{\text{def}}{=} \prod_{i<j} \boldsymbol{Q}^{(i,j)}(\theta_{i,j}) \prod_{i<j} \boldsymbol{T}^{(i,j)}(t_{i,j}) \tag{2.69}$$

respectively. By means of the JDC criterion $\Psi_{1'}(\boldsymbol{A})$ in (2.46), each $\boldsymbol{T}^{(i,j)}(t_{i,j})$ and $\boldsymbol{Q}^{(i,j)}(\theta_{i,j})$ can be estimated by solving the two following Jacobi-like optimization problems, respectively:

$$t_{i,j} = \underset{t_{i,j}}{\operatorname{argmin}}\, \Psi_{1'}(\widehat{\boldsymbol{A}}\boldsymbol{T}^{(i,j)}(t_{i,j})) \tag{2.70}$$

$$\theta_{i,j} = \underset{\theta_{i,j}}{\operatorname{argmin}}\, \Psi_{1'}(\widehat{\boldsymbol{A}}\boldsymbol{Q}^{(i,j)}(\theta_{i,j})) \tag{2.71}$$

where $\widehat{\boldsymbol{A}}$ is the estimate of $\boldsymbol{A}$ at the $(it-1)$-th iteration. It is easy to show that the right hand side of (2.70) is a second degree polynomial function in $t_{i,j}$. Then the global optimal $t_{i,j}$ can be directly expressed as follows:

$$t_{i,j} = -\frac{\sum_k \sum_{n\neq j} \widehat{C}^{(k)}_{i,n}\widehat{C}^{(k)}_{j,n}}{\sum_k \sum_{n\neq j} \widehat{C}^{(k)}_{i,n}\widehat{C}^{(k)}_{i,n}} \tag{2.72}$$

where $\widehat{\boldsymbol{C}}^{(k)} = \widehat{\boldsymbol{A}}^{\mathsf{T}}\boldsymbol{C}^{(k,-1)}\widehat{\boldsymbol{A}}$ is a $(N\times N)$ constant matrix, and where $\widehat{C}^{(k)}_{i,n}$ and $\widehat{C}^{(k)}_{j,n}$ are the (i,n)-th and (j,n)-th components of $\widehat{\boldsymbol{C}}^{(k)}$, respectively. In addition, the right hand side of (2.71) can be written as the following quadratic form:

$$\Psi_{1'}(\theta_{i,j}) = \boldsymbol{\theta}^{\mathsf{T}}_{i,j}(\boldsymbol{G}^{\mathsf{T}}\boldsymbol{G})\boldsymbol{\theta}_{i,j} \tag{2.73}$$

where $\boldsymbol{\theta}_{i,j}$ is a 2-dimensional parameter vector defined by $\boldsymbol{\theta}_{i,j} = [\cos(2\theta_{i,j}), \sin(2\theta_{i,j})]^{\mathsf{T}}$, and where $\boldsymbol{G} \in \mathbb{R}^{K\times 2}$ is a coefficient matrix, whose $(k,1)$-th and $(k,2)$-th components are equal to $\widehat{C}^{(k)}_{j,j} - \widehat{C}^{(k)}_{i,i}$ and $\widehat{C}^{(k)}_{i,j} + \widehat{C}^{(k)}_{j,i}$, respectively. The vector $\boldsymbol{\theta}_{i,j}$ can be chosen as the unit-norm eigenvector corresponding to the smaller eigenvalue of the matrix $\boldsymbol{G}^{\mathsf{T}}\boldsymbol{G}$. Then $\theta_{i,j}$ can be computed from $\boldsymbol{\theta}_{i,j}$ accordingly. The derivations of equations (2.72) and (2.73) are straightforward and therefore omitted.

In fact, the ways of estimating the two sets of parameters $\{\ell_{i,j}, u_{i,j}\}$ in equation (2.43), and $\{\theta_{i,j}, u'_{i,j}\}$ in equation (2.69) are identical to those of Afsari's LUJ1D and QRJ1D methods [Afsari, 2006] performing on the inverted input matrices $\boldsymbol{C}^{(k,-1)}$, respectively. In practice, for iJDC$^{+}_{\text{LU-1}}$ and iJDC$^{+}_{\text{QR-1}}$, in each Jacobi-like iteration for computing an elementary triangular matrix, we suggest to compute $t_{i,j}$ by equation (2.72) first. If all the elements in the j-th column vector of $\widehat{\boldsymbol{A}}\boldsymbol{T}^{(i,j)}(t_{i,j})$ have the same sign ς, the j-th column vector of $\widehat{\boldsymbol{A}}^{(\text{new})}$ is replaced by $\varsigma(\widehat{\boldsymbol{a}}_j + t_{i,j}\widehat{\boldsymbol{a}}_i)$, while the other column vectors remain

unchanged, where $\widehat{\boldsymbol{a}}_i$ and $\widehat{\boldsymbol{a}}_j$ are the i-th and j-th column vectors of $\widehat{\boldsymbol{A}}$, respectively. Otherwise, $t_{i,j}$ is computed by minimizing (2.53). Then $\widehat{\boldsymbol{A}}$ is updated by means of equation (2.29). Regarding the Givens rotation matrices $\boldsymbol{Q}^{(i,j)}(\theta_{i,j})$ in the iJDC$^{+}_{\text{QR-1}}$ algorithm, $\theta_{i,j}$ is computed in the same manner by minimizing either (2.73) or (2.65). Then $\widehat{\boldsymbol{A}}^{(\text{new})}$ is computed accordingly.

2.3.2.2 Row balancing

Afsari reported in [Afsari, 2006] that if the row vectors of matrices $\widehat{\boldsymbol{C}}^{(k)}$ ($k \in \{1, \cdots, K\}$) are not balanced in their norms, the computation of the parameters could be inaccurate. In order to cope with this effect, we apply Afsari's row balancing scheme every few sweeps. Such a scheme updates each $\widehat{\boldsymbol{C}}^{(k)}$ by $\widehat{\boldsymbol{C}}^{(k,\text{new})} = \boldsymbol{\Lambda}\widehat{\boldsymbol{C}}^{(k)}\boldsymbol{\Lambda}$ and $\widehat{\boldsymbol{A}}$ by $\widehat{\boldsymbol{A}}^{(\text{new})} = \widehat{\boldsymbol{A}}\boldsymbol{\Lambda}$ using a diagonal matrix $\boldsymbol{\Lambda} \in \mathbb{R}_{+}^{N\times N}$, whose diagonal elements are defined as follows:

$$\Lambda_{n,n} = \frac{1}{\sqrt{\sum_{k=1}^{K} \|\widehat{\boldsymbol{C}}^{(k)}_{n,:}\|^2}}, \quad n \in \{1, 2, \cdots, N\} \tag{2.74}$$

where $\widehat{\boldsymbol{C}}^{(k)}_{n,:}$ denotes the n-th row vector of matrix $\widehat{\boldsymbol{C}}^{(k)}$.

2.3.3 Discussions

The effectiveness of the iJDC$^{+}_{\text{LU-1}}$ and iJDC$^{+}_{\text{QR-1}}$ algorithms relies on the two assumptions that $\boldsymbol{A}$ is a nonsingular square matrix and that the main diagonals of $\{\boldsymbol{D}^{(k)}\}$ do not contain any zero entry. In ICA, the first nonsingularity assumption of the mixing matrix $\boldsymbol{A}$ is mostly satisfied. In the following we verify the validity of the second assumption.

Provided that all the sources are non-Gaussian, which is often the case in practice [Comon and Jutten, 2010], we can resort to use the FO cumulant array of the observation data. As we mentioned in section 1.3.1, the jointly diagonalizable matrices $\{\boldsymbol{C}^{(k)}\}$ can be built by stacking the matrix slices of the cumulant array. Then the $(N^2 \times N)$ matrix $\boldsymbol{D}$ whose k-th row vector contains the diagonal components of $\boldsymbol{D}^{(k)}$, can be expressed by $\boldsymbol{D} = (\boldsymbol{A} \odot \boldsymbol{A})\boldsymbol{C}_{\boldsymbol{s}}^{(4)}$, where $\boldsymbol{C}_{\boldsymbol{s}}^{(4)} = \text{diag}\,[\mathcal{C}_{1,1,1,1,\boldsymbol{s}}, \ldots, \mathcal{C}_{N,N,N,N,\boldsymbol{s}}]$ is a $(N \times N)$ diagonal matrix with $\mathcal{C}_{n,n,n,n,\boldsymbol{s}}$ being the FO cumulant of the n-th source, $n \in \{1, \ldots, N\}$, and where $\odot$ denotes the Khatri-Rao product. Suppose that $\boldsymbol{A}$ is a nonsingular dense mixing matrix, then $\boldsymbol{D}$ is also nonsingular and contains non-zeros components. A dense mixing matrix means that the sources are highly mixed, therefore the components of $\boldsymbol{A}$ have relative large values. A dense mixing matrix is often encountered in BSS problems, such as in hyperspectral unmixing [Miao and Qi, 2007, Chan et al., 2009] and in separating MRS signals [Moussaoui, 2005]. Consequently, the nonzero assumption is reasonable for some BSS applications.

One drawback brought by these assumptions is that the mixing matrix (and the joint transformation matrix) $\boldsymbol{A}$ is constrained to be square, since a non-square matrix $\boldsymbol{A} \in \mathbb{R}_{+}^{N\times P}$ with $N > P$ will lead to the jointly diagonalizable matrices $\boldsymbol{C}^{(k)} = \boldsymbol{A}\boldsymbol{D}^{(k)}\boldsymbol{A}^{\mathsf{T}}$ being rank deficient and non-invertible. In ICA, when such a non-square matrix $\boldsymbol{A}$ is encountered, we can compress $\boldsymbol{A}$ by means of a nonnegative matrix $\overline{\boldsymbol{W}} \in \mathbb{R}_{+}^{P\times N}$ such that the resulting matrix $\overline{\boldsymbol{A}}_{+} = \overline{\boldsymbol{W}}\boldsymbol{A}$ is a nonnegative square matrix. $\overline{\boldsymbol{W}}$ can be computed by means of the nonnegative compression algorithm that we proposed in [Wang et al., 2014d], which transforms the classical prewhitening matrix [Belouchrani et al., 1997]

into a nonnegative one by linear mappings. Such a method is described in appendix B. Then the iJDC$^+_{\text{LU-1}}$ and iJDC$^+_{\text{QR-1}}$ algorithms are used to compute the compressed mixing matrix $\overline{\boldsymbol{A}}_+$.

Finally, the pseudo-codes for the iJDC$^+_{\text{LU-1}}$ and iJDC$^+_{\text{QR-1}}$ algorithms are given in the appendix C.3 and C.4, respectively.

2.4 The JDC$^+_{\text{LU-1}}$ and JDC$^+_{\text{LU-2}}$ algorithms

2.4.1 Algorithm derivation

In some practical JDC problems, the main diagonals of matrices $\boldsymbol{D}^{(k)}$ could contain zero entries, which breaks the second assumption of the iJDC$^+_{\text{LU-1}}$ and iJDC$^+_{\text{QR-1}}$ algorithms. Therefore such an assumption could limit the applicability of iJDC$^+_{\text{LU-1}}$ and iJDC$^+_{\text{QR-1}}$. Moreover, even though this assumption is satisfied, in some cases, some jointly diagonalizable matrices $\boldsymbol{C}^{(k)}$ may be ill-conditioned. The inverse preprocessing of these matrices could generate bias in the following JDC procedure. Consequently, the purpose of this section is to develop Jacobi-like algorithms that require less assumptions. More precisely, we allow the diagonal matrices $\boldsymbol{D}^{(k)}$ contain zero components on their main diagonals and discard the preprocessing step of inverting each $\boldsymbol{C}^{(k)}$. We still assume that the nonnegative joint transformation matrix $\boldsymbol{A}$ is nonsingular. In addition, we would like to study the performance of the Jacobi-like procedure on difference cost functions. Hence, by considering both the indirect-fit JDC criteria (1.27) and (1.28) and the LU matrix factorization, we derive two algorithms, namely JDC$^+_{\text{LU-1}}$ and JDC$^+_{\text{LU-2}}$.

By inserting the square change of variable $\boldsymbol{A} = \boldsymbol{B}^{\boxdot 2}$ into the indirect-fit JDC criteria (1.27) and (1.28), we obtain the following two nonnegativity constrained JDC criteria, respectively:

$$\Psi_1^+(\boldsymbol{B}) = \sum_{k=1}^{K} \left\| \operatorname{off}\{(\boldsymbol{B}^{\boxdot 2})^{-1}\boldsymbol{C}^{(k)}(\boldsymbol{B}^{\boxdot 2})^{-\mathsf{T}}\} \right\|_F^2 \tag{2.75}$$

$$\Psi_2^+(\boldsymbol{B}) = \sum_{k=1}^{K} \left\| \boldsymbol{C}^{(k)} - \boldsymbol{B}^{\boxdot 2} \operatorname{diag}\{(\boldsymbol{B}^{\boxdot 2})^{-1}\boldsymbol{C}^{(k)}(\boldsymbol{B}^{\boxdot 2})^{-\mathsf{T}}\}(\boldsymbol{B}^{\boxdot 2})^{\mathsf{T}} \right\|_F^2 \tag{2.76}$$

By means of the LU matrix factorization of $\boldsymbol{B}$ and the notation $\boldsymbol{B} = \prod_{i\neq j} \boldsymbol{T}^{(i,j)}(t_{i,j})$ defined in equation (2.26), the minimization of (2.75) and (2.76) with respect to $\boldsymbol{B}$ is converted to the estimation of the set of $N(N-1)$ parameters $t_{i,j}$. Therefore, each parameter $t_{i,j}$ can be estimated sequentially using one of the two following Jacobi-like optimization procedures:

$$t_{i,j} = \underset{t_{i,j}}{\operatorname{argmin}}\, \Psi_1^+(\boldsymbol{B}^{(it-1)}\boldsymbol{T}^{(i,j)}(t_{i,j})) \tag{2.77}$$

$$t_{i,j} = \underset{t_{i,j}}{\operatorname{argmin}}\, \Psi_2^+(\boldsymbol{B}^{(it-1)}\boldsymbol{T}^{(i,j)}(t_{i,j})) \tag{2.78}$$

As in the previous sections, let $\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{B}}$ denote $\boldsymbol{A}^{(it-1)}$ and $\boldsymbol{B}^{(it-1)}$, respectively, before estimating the parameter $t_{i,j}$. Let $\widehat{\boldsymbol{A}}^{(\text{new})}$ and $\widehat{\boldsymbol{B}}^{(\text{new})}$ stand for $\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{B}}$ updated by $\boldsymbol{T}^{(i,j)}(t_{i,j})$, respectively. The update of $\widehat{\boldsymbol{B}}$ is defined by $\widehat{\boldsymbol{B}}^{(\text{new})} = \widehat{\boldsymbol{B}}\boldsymbol{T}^{(i,j)}(t_{i,j})$. Inserting $\widehat{\boldsymbol{B}}^{(\text{new})}$ into the Jacobi-like optimizations (2.77) and (2.78), the optimal $t_{i,j}$ can be chosen

as the global minimum of the following two local cost functions, respectively:

$$\Psi_1^+(t_{i,j}) = \sum_{k=1}^{K} \left\| \text{off} \left\{ \left[(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2} \right]^{-1} \boldsymbol{C}^{(k)} \left[(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2} \right]^{-\mathsf{T}} \right\} \right\|_F^2 \tag{2.79}$$

$$\Psi_2^+(t_{i,j}) = \sum_{k=1}^{K} \left\| \boldsymbol{C}^{(k)} - (\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2} \times \right.$$
$$\left. \text{diag} \left\{ \left[(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2} \right]^{-1} \boldsymbol{C}^{(k)} \left[(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2} \right]^{-\mathsf{T}} \right\} \left[(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2} \right]^{\mathsf{T}} \right\|_F^2 \tag{2.80}$$

We define the K matrices $\widehat{\boldsymbol{C}}^{(k,\text{new})} \in \mathbb{R}^{N \times N}$ as follows:

$$\widehat{\boldsymbol{C}}^{(k,\text{new})} = \left[(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2} \right]^{-1} \boldsymbol{C}^{(k)} \left[(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2} \right]^{-\mathsf{T}}, \quad k \in \{1, 2, \ldots, K\} \tag{2.81}$$

According to equations (2.79), (2.80) and (2.81), the computation of the cost functions $\Psi_1^+(t_{i,j})$ and $\Psi_2^+(t_{i,j})$ requires to express the components of the K matrices $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ as an explicit function of $t_{i,j}$. Therefore, it is necessary to write $[(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}]^{-1}$ as a function of $t_{i,j}$. The explicit expression of $(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}$ as a function of $t_{i,j}$ is provided in equation (2.29) in proposition 2.1. In the right hand side of (2.29), the first term of the sum is a nonsingular matrix and the second term is a rank-1 matrix. The inverse of the sum of such two matrices can be computed by the Sherman-Morrison formula [Sherman and Morrison, 1950, Bartlett, 1951]:

Theorem 2.1. *Suppose that $\boldsymbol{R} \in \mathbb{R}^{N \times N}$ is a nonsingular square matrix and $\boldsymbol{u}$, $\boldsymbol{v} \in \mathbb{R}^N$ are two column vectors satisfying $1 + \boldsymbol{v}^{\mathsf{T}} \boldsymbol{R}^{-1} \boldsymbol{u} \neq 0$, then:*

$$(\boldsymbol{R} + \boldsymbol{u}\boldsymbol{v}^{\mathsf{T}})^{-1} = \boldsymbol{R}^{-1} - \frac{\boldsymbol{R}^{-1} \boldsymbol{u}\boldsymbol{v}^{\mathsf{T}} \boldsymbol{R}^{-1}}{1 + \boldsymbol{v}^{\mathsf{T}} \boldsymbol{R}^{-1} \boldsymbol{u}} \tag{2.82}$$

Suppose that matrix $\widehat{\boldsymbol{B}}$, the two vectors $2\, t_{i,j} (\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)$ and $\boldsymbol{e}_j$ satisfy the conditions of the Sherman-Morrison formula, the expression of $[(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}]^{-1}$ as a function of the parameter $t_{i,j}$ has the following form:

$$\left[(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2} \right]^{-1} = \boldsymbol{T}^{(i,j)}(-t_{i,j}^2)\, \boldsymbol{Q} (\widehat{\boldsymbol{B}}^{\boxdot 2})^{-1} \tag{2.83}$$

where $\boldsymbol{Q} \in \mathbb{R}^{N \times N}$ is defined as follows:

$$\boldsymbol{Q} = \boldsymbol{I}_N - \frac{2\, t_{i,j}}{1 + 2\beta_j t_{i,j}} \boldsymbol{\beta} \boldsymbol{e}_j^{\mathsf{T}} \tag{2.84}$$

where $\boldsymbol{I}_N \in \mathbb{R}^{N \times N}$ is the identity matrix, where $\boldsymbol{\beta} \in \mathbb{R}^N$ is a column vector defined as follows:

$$\boldsymbol{\beta} = (\widehat{\boldsymbol{B}}^{\boxdot 2})^{-1} (\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j) \tag{2.85}$$

and where β_j is the j-th element of vector $\boldsymbol{\beta}$. Inserting equation (2.83) into (2.81), $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ can be rewritten as follows:

$$\widehat{\boldsymbol{C}}^{(k,\text{new})} = \boldsymbol{T}^{(i,j)}(-t_{i,j}^2)\, \boldsymbol{Q}\, \widehat{\boldsymbol{C}}^{(k)} \boldsymbol{Q}^{\mathsf{T}}\, \boldsymbol{T}^{(i,j)}(-t_{i,j}^2)^{\mathsf{T}} \tag{2.86}$$

where $\widehat{\boldsymbol{C}}^{(k)} = \widehat{\boldsymbol{A}}^{-1} \boldsymbol{C}^{(k)} \widehat{\boldsymbol{A}}^{-\mathsf{T}}$ is a $(N \times N)$ constant matrix. Then through a straightforward computation of (2.86), each component of $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ can be expressed as a function of the parameter $t_{i,j}$ as described in the following two propositions.

Proposition 2.6. *Each non-diagonal component of* $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ *is a rational function in* $t_{i,j}$*:*

$$\widehat{C}^{(k,\text{new})}_{\substack{m,n\\ m\neq n}} = \frac{E^{(k,3)}_{m,n} t^3_{i,j} + E^{(k,2)}_{m,n} t^2_{i,j} + E^{(k,1)}_{m,n} t_{i,j} + E^{(k,0)}_{m,n}}{(1+2\beta_j t_{i,j})^2} \tag{2.87}$$

where $E^{(k,3)}_{m,n}$*,* $E^{(k,2)}_{m,n}$*,* $E^{(k,1)}_{m,n}$ *and* $E^{(k,0)}_{m,n}$ *are the* (m,n)*-th components of the* $(N\times N)$ *symmetric coefficient matrices* $\boldsymbol{E}^{(k,3)}$*,* $\boldsymbol{E}^{(k,2)}$*,* $\boldsymbol{E}^{(k,1)}$ *and* $\boldsymbol{E}^{(k,0)}$*, respectively. These coefficients are defined as follows:*

$$E^{(k,3)}_{m,n} = \begin{cases} 2(\widehat{C}^{(k)}_{j,j}\beta_m - \widehat{C}^{(k)}_{m,j}\beta_j) & \text{if } \ n=i, 1\leq m\neq i\leq N \\ E^{(k,3)}_{n,m} & \text{if } m=i, 1\leq n\neq i\leq N \\ 0 & \text{otherwise} \end{cases} \tag{2.88}$$

$$E^{(k,2)}_{m,n} = \begin{cases} 4(\widehat{C}^{(k)}_{j,j}\beta_m\beta_n + \widehat{C}^{(k)}_{m,n}\beta_j^2 - (\widehat{C}^{(k)}_{m,j}\beta_n + \widehat{C}^{(k)}_{j,n}\beta_m)\beta_j) & \\ \qquad\qquad \text{if } 1\leq m<n\leq N, m\neq i, n\neq i & \\ 4(\widehat{C}^{(k)}_{j,j}\beta_m\beta_n + \widehat{C}^{(k)}_{m,n}\beta_j^2 - (\widehat{C}^{(k)}_{m,j}\beta_n + \widehat{C}^{(k)}_{j,n}\beta_m)\beta_j) - \widehat{C}^{(k)}_{m,j} & \\ \qquad\qquad \text{if } n=i, 1\leq m<i & \\ E^{(k,2)}_{n,m} \qquad \text{if } 1\leq n<m\leq N & \\ 0 \qquad \text{otherwise} \end{cases} \tag{2.89}$$

$$E^{(k,1)}_{m,n} = \begin{cases} 4\widehat{C}^{(k)}_{m,n}\beta_j - 2(\widehat{C}^{(k)}_{m,j}\beta_n + \widehat{C}^{(k)}_{j,n}\beta_m) & \text{if } 1\leq m<n\leq N \\ E^{(k,1)}_{n,m} & \text{if } 1\leq n<m\leq N \\ 0 & \text{otherwise} \end{cases} \tag{2.90}$$

$$E^{(k,0)}_{m,n} = \begin{cases} \widehat{C}^{(k)}_{m,n} & \text{if } 1\leq m\neq n\leq N \\ 0 & \text{otherwise} \end{cases} \tag{2.91}$$

where $\widehat{C}^{(k)}_{m,n}$ *is the* (m,n)*-th component of the matrix* $\widehat{\boldsymbol{C}}^{(k)}$*, and where* β_m *is the* m*-th element of the vector* $\boldsymbol{\beta}$ *which is defined in equation* (2.85)*.*

Proposition 2.7. *Let us define* K *column vectors* $\widehat{\boldsymbol{d}}^{(k)}\in\mathbb{R}^N$ *with* $k\in\{1,\ldots,K\}$*, each containing the diagonal elements of* $\widehat{\boldsymbol{C}}^{(k,\text{new})}$*, then the* n*-th element of* $\widehat{\boldsymbol{d}}^{(k)}$*, denoted by* $\widehat{d}^{(k)}_n$*, can be expressed as a rational function in* $t_{i,j}$ *as follows:*

$$\widehat{d}^{(k)}_n = \frac{f^{(k,4)}_n t^4_{i,j} + f^{(k,3)}_n t^3_{i,j} + f^{(k,2)}_n t^2_{i,j} + f^{(k,1)}_n t_{i,j} + f^{(k,0)}_n}{(1+2\beta_j t_{i,j})^2} \tag{2.92}$$

where $f^{(k,4)}_n$*,* $f^{(k,3)}_n$*,* $f^{(k,2)}_n$*,* $f^{(k,1)}_n$ *and* $f^{(k,0)}_n$ *are the* n*-th elements of the* N*-dimensional coefficient vectors* $\boldsymbol{f}^{(k,4)}$*,* $\boldsymbol{f}^{(k,3)}$*,* $\boldsymbol{f}^{(k,2)}$*,* $\boldsymbol{f}^{(k,1)}$ *and* $\boldsymbol{f}^{(k,0)}$*, respectively, with* $n\in\{1,\ldots,N\}$*. These coefficients are defined as follows:*

$$f^{(k,4)}_n = \begin{cases} \widehat{C}^{(k)}_{j,j} & \text{if } n=i \\ 0 & \text{otherwise} \end{cases} \tag{2.93}$$

$$f^{(k,3)}_n = \begin{cases} 4\widehat{C}^{(k)}_{j,j}\beta_i - 2(\widehat{C}^{(k)}_{i,j} + \widehat{C}^{(k)}_{j,i})\beta_j & \text{if } n=i \\ 0 & \text{otherwise} \end{cases} \tag{2.94}$$

$$f_n^{(k,2)} = \begin{cases} 4(\widehat{C}_{n,n}^{(k)}\beta_j^2 - (\widehat{C}_{n,j}^{(k)} + \widehat{C}_{j,n}^{(k)})\beta_n\beta_j + \widehat{C}_{j,j}^{(k)}\beta_n^2) \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{if } n \neq i, n \neq j \\ 4\widehat{C}_{i,i}^{(k)}\beta_j^2 - (\widehat{C}_{i,j}^{(k)} + \widehat{C}_{j,i}^{(k)})(1 + 4\beta_i\beta_j) + 4\widehat{C}_{j,j}^{(k)}\beta_i^2 \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{if } n = i \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \tag{2.95}$$

$$f_n^{(k,1)} = \begin{cases} 4\widehat{C}_{n,n}^{(k)}\beta_j - 2(\widehat{C}_{n,j}^{(k)} + \widehat{C}_{j,n}^{(k)})\beta_n & \text{if } n \neq j \\ 0 & \text{otherwise} \end{cases} \tag{2.96}$$

$$f_n^{(k,0)} = \widehat{C}_{n,n}^{(k)} \tag{2.97}$$

where $\widehat{C}_{i,j}^{(k)}$ is the (i,j)-th component of the matrix $\widehat{\boldsymbol{C}}^{(k)}$, and where β_i is the i-th element of the vector $\boldsymbol{\beta}$ which is defined in equation (2.85).

The proofs of propositions 2.6 and 2.7 are provided in the appendix A.5. Then we can derive the proposed $\text{JDC}^{+}_{\text{LU-1}}$ and $\text{JDC}^{+}_{\text{LU-2}}$ algorithms.

2.4.1.1 The $\text{JDC}^{+}_{\text{LU-1}}$ algorithm

The cost function $\Psi_1^+(t_{i,j})$ defined in (2.79) is equal to the total sum of the squares of the non-diagonal components of $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ defined in proposition 2.6, therefore it can be expressed in the following compact matrix form:

$$\Psi_1^+(t_{i,j}) = \sum_{k=1}^{K} \left\| \boldsymbol{E}^{(k)}\boldsymbol{\tau}_{i,j} \right\|_F^2 = \boldsymbol{\tau}_{i,j}^{\mathsf{T}}\, \boldsymbol{Q}_E\, \boldsymbol{\tau}_{i,j} \tag{2.98}$$

where $\boldsymbol{Q}_E \in \mathbb{R}^{4\times 4}$ is a symmetric coefficient matrix defined by:

$$\boldsymbol{Q}_E = \sum_{k=1}^{K} (\boldsymbol{E}^{(k)})^{\mathsf{T}} \boldsymbol{E}^{(k)} \tag{2.99}$$

where $\boldsymbol{E}^{(k)}$ is a $(N^2 \times 4)$ matrix defined as follows:

$$\boldsymbol{E}^{(k)} = \left[\text{vec}\,(\boldsymbol{E}^{(k,3)}), \text{vec}\,(\boldsymbol{E}^{(k,2)}), \text{vec}\,(\boldsymbol{E}^{(k,1)}), \text{vec}\,(\boldsymbol{E}^{(k,0)})\right] \tag{2.100}$$

and where $\boldsymbol{\tau}_{i,j}$ is defined as the following 4-dimensional parameter vector :

$$\boldsymbol{\tau}_{i,j} = \frac{1}{(1 + 2\beta_j t_{i,j})^2}[t_{i,j}^3,\, t_{i,j}^2,\, t_{i,j},\, 1]^{\mathsf{T}} \tag{2.101}$$

Equation (2.98) shows that $\Psi_1^+(t_{i,j})$ is a rational function, where the degrees of the numerator and the denominator are 6 and 4, respectively. The global minimum $t_{i,j}$ can be obtained by computing the roots of the numerator of its derivative,which is a polynomial of degree 9, and selecting the one yielding the smallest value of (2.98). In practice, the roots of a polynomial of degree 9 can be obtained by computing the eigenvalues of the corresponding companion matrix of dimension (9×9) [Golub and Van Loan, 1996].

2.4.1.2 The JDC$^{+}_{\text{LU-2}}$ algorithm

According to proposition 2.7, $\widehat{\boldsymbol{d}}^{(k)}$ can be expressed in the following compact matrix form:

$$\widehat{\boldsymbol{d}}^{(k)} = \boldsymbol{F}^{(k)}\boldsymbol{\zeta}_{i,j} \tag{2.102}$$

where $\boldsymbol{F}^{(k)} \in \mathbb{R}^{N\times 5}$ is a coefficient matrix:

$$\boldsymbol{F}^{(k)} = \left[\boldsymbol{f}^{(k,4)},\, \boldsymbol{f}^{(k,3)},\, \boldsymbol{f}^{(k,2)},\, \boldsymbol{f}^{(k,1)},\, \boldsymbol{f}^{(k,0)}\right] \tag{2.103}$$

and where $\boldsymbol{\zeta}_{i,j} \in \mathbb{R}^5$ is a parameter vector defined as follows:

$$\boldsymbol{\zeta}_{i,j} = \frac{1}{(1+2\beta_j t_{i,j})^2}[t_{i,j}^4,\, t_{i,j}^3,\, t_{i,j}^2,\, t_{i,j},\, 1]^{\mathsf{T}} \tag{2.104}$$

The use of vec (.) operator in criterion (2.80) leads to the following equivalent form of $\Psi_2^{+}(t_{i,j})$:

$$\Psi_2^{+}(t_{i,j}) = \sum_{k=1}^{K} \left\| \boldsymbol{c}^{(k)} - \left[(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2} \odot (\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}\right] \widehat{\boldsymbol{d}}^{(k)} \right\|_F^2 \tag{2.105}$$

where $\boldsymbol{c}^{(k)} = \text{vec}\,(\boldsymbol{C}^{(k)})$, and where $\odot$ denotes the Khatri-Rao product. Let us consider the following proposition:

Proposition 2.8. *According to equation* (2.29) *and the definition of Khatri-Rao product, the following equality holds:*

$$(\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2} \odot (\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2} = (\widehat{\boldsymbol{A}} \odot \widehat{\boldsymbol{A}}) + \boldsymbol{G}\,\boldsymbol{\zeta}_{i,j}\,\boldsymbol{e}_j^{\mathsf{T}} \tag{2.106}$$

where $\boldsymbol{G}$ *is a* $(N^2\times 5)$ *matrix, whose column vectors are defined as follows:*

$$\boldsymbol{g}_1 = \text{vec}\,(\widehat{\boldsymbol{a}}_i\widehat{\boldsymbol{a}}_i^{\mathsf{T}}) \tag{2.107}$$
$$\boldsymbol{g}_2 = \text{vec}\,(2\,\widehat{\boldsymbol{a}}_i(\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)^{\mathsf{T}} + 2\,(\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)\widehat{\boldsymbol{a}}_i^{\mathsf{T}}) \tag{2.108}$$
$$\boldsymbol{g}_3 = \text{vec}\,(\widehat{\boldsymbol{a}}_i\widehat{\boldsymbol{a}}_j^{\mathsf{T}} + \widehat{\boldsymbol{a}}_j\widehat{\boldsymbol{a}}_i^{\mathsf{T}} + 4\,(\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)(\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)^{\mathsf{T}}) \tag{2.109}$$
$$\boldsymbol{g}_4 = \text{vec}\,(2\,\widehat{\boldsymbol{a}}_j(\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)^{\mathsf{T}} + 2\,(\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)\widehat{\boldsymbol{a}}_j^{\mathsf{T}}) \tag{2.110}$$
$$\boldsymbol{g}_5 = \boldsymbol{0} \tag{2.111}$$

and where $\widehat{\boldsymbol{a}}_i$ *and* $\widehat{\boldsymbol{a}}_j$ *are the* i*-th and* j*-th column vectors of* $\widehat{\boldsymbol{A}}$*, respectively.*

The proof of proposition 2.8 is given in appendix A.6. By inserting equations (2.102) and (2.106) into (2.105), the cost function $\Psi_2^{+}(t_{i,j})$ can be rewritten as follows:

$$\Psi_2^{+}(t_{i,j}) = \sum_{k=1}^{K} \left\| \boldsymbol{c}^{(k)} - \boldsymbol{G}_A^{(k)}\,\boldsymbol{\zeta}_{i,j} \right\|_F^2 \tag{2.112}$$

where $\boldsymbol{G}_A^{(k)}$ is a $(N^2\times 5)$ matrix defined as follows:

$$\boldsymbol{G}_A^{(k)} = (\widehat{\boldsymbol{A}} \odot \widehat{\boldsymbol{A}})\boldsymbol{F}^{(k)} + f_j^{(k,0)}\boldsymbol{G} \tag{2.113}$$

Now, let us define a $(N^2\times 5)$ matrix $\boldsymbol{H}^{(k)}$ as follows:

$$\boldsymbol{H}^{(k)} = \boldsymbol{c}^{(k)}(4\,\beta_j^2\,\boldsymbol{e}_3^{\mathsf{T}} + 4\,\beta_j\,\boldsymbol{e}_4^{\mathsf{T}} + \boldsymbol{e}_5^{\mathsf{T}}) - \boldsymbol{G}_A^{(k)} \tag{2.114}$$

where $\boldsymbol{e}_n$, $n \in \{3,4,5\}$ is the n-th column vector of the (5×5) identity matrix. Then the criterion $\Psi_2^+(t_{i,j})$ can be expressed in a compact matrix form as follows:

$$\Psi_2^+(t_{i,j}) = \sum_{k=1}^{K} \left\| \boldsymbol{H}^{(k)} \boldsymbol{\zeta}_{i,j} \right\|_F^2 = \boldsymbol{\zeta}_{i,j}^{\mathsf{T}} \boldsymbol{Q}_H \, \boldsymbol{\zeta}_{i,j} \tag{2.115}$$

where:

$$\boldsymbol{Q}_H = \sum_{k=1}^{K} (\boldsymbol{H}^{(k)})^{\mathsf{T}} \boldsymbol{H}^{(k)} \tag{2.116}$$

is a (5×5) symmetric coefficient matrix.

Equation (2.115) shows that the cost function $\Psi_2^+(t_{i,j})$ is a rational function, where the degrees of the numerator and the denominator are 8 and 4, respectively. The global minimum $t_{i,j}$ can be obtained by computing the roots of its derivative and selecting the one yielding the smallest value of (2.115).

Once the optimal $t_{i,j}$ is computed by either JDC$^{+}_{\text{LU-1}}$ or JDC$^{+}_{\text{LU-2}}$, $\widehat{\boldsymbol{B}}^{(\text{new})}$ and $\widehat{\boldsymbol{A}}^{(\text{new})}$ are computed as described in proposition 2.1. Then the Jacobi-like procedure continues to compute $t_{i,j}$ with the next (i,j) index couple. The processing of all the $N(N-1)$ parameters is called a sweep. In practice, it is necessary to perform several sweeps in order to ensure the convergence of this sequential optimization procedure. Such procedures can be terminated when the values of (2.98) and (2.115), or their relative decreases between two successive sweeps fall below a specified small positive threshold. Such a stopping criterion is guaranteed to be met since the criterions (2.98) and (2.115) are guaranteed to decrease (or at least to not increase) after each Jacobi-like sweep.

2.4.2 Generalization to the non-square case

In practice a JDC problem often involves a non-square joint transformation matrix $\boldsymbol{A} \in \mathbb{R}_+^{N \times P}$ and K diagonal matrices $\boldsymbol{D}^{(k)} \in \mathbb{R}^{P \times P}$ with $N > P$. In order to compute a non-square full column rank matrix $\boldsymbol{A}$ by minimizing the cost functions (1.27) and (1.28), the inverse operator in (1.27) and (1.28) should be replaced by the pseudo-inverse operator. However, the pseudo-inverse is more difficult to handle than the inverse operation. In order to avoid computing the pseudo-inverse, let us firstly define the following two $(N \times N)$ matrices:

$$\mathring{\boldsymbol{A}} = [\boldsymbol{A}, \boldsymbol{e}_{P+1}, \ldots, \boldsymbol{e}_N] = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_P, \boldsymbol{e}_{P+1}, \ldots, \boldsymbol{e}_N] \tag{2.117}$$

$$\mathring{\boldsymbol{D}}^{(k)} = \left[\begin{array}{c:c} \boldsymbol{D}^{(k)} & \boldsymbol{0}_{P,(N-P)} \\ \hdashline \boldsymbol{0}_{(N-P),P} & \boldsymbol{0}_{(N-P),(N-P)} \end{array}\right] \tag{2.118}$$

where $\boldsymbol{a}_p$ is the p-th column vector of $\boldsymbol{A}$, where $\boldsymbol{e}_p$ is the p-th column vector of the identity matrix $\boldsymbol{I}_N$, and where $\boldsymbol{0}_{Y,Z}$ denotes a $(Y \times Z)$ zero matrix. Then we obtain the following equality:

$$\boldsymbol{C}^{(k)} = \boldsymbol{A} \, \boldsymbol{D}^{(k)} \boldsymbol{A}^{\mathsf{T}} = \mathring{\boldsymbol{A}} \, \mathring{\boldsymbol{D}}^{(k)} \mathring{\boldsymbol{A}}^{\mathsf{T}} \tag{2.119}$$

Equation (2.119) suggests that we can transform the non-square matrix $\boldsymbol{A}$ into a square matrix $\mathring{\boldsymbol{A}}$ by combining $\boldsymbol{A}$ with additional columns $\{\boldsymbol{e}_p\}$ $p \in \{P+1, P+2, \ldots, N\}$, without changing the structure of $\boldsymbol{C}^{(k)}$. In general, each column vector $\boldsymbol{a}_p$ is linearly

independent of the $N-P$ vectors $\boldsymbol{e}_p$, therefore $\mathring{\boldsymbol{A}}$ is nonsingular. Similarly, we can define the following three $(N \times N)$ matrices:

$$\mathring{\boldsymbol{B}} = [\boldsymbol{B}, \boldsymbol{e}_{P+1}, \ldots, \boldsymbol{e}_N] \tag{2.120}$$

$$\mathring{\boldsymbol{L}} = [\boldsymbol{L}, \boldsymbol{e}_{P+1}, \ldots, \boldsymbol{e}_N] \tag{2.121}$$

$$\mathring{\boldsymbol{U}} = \left[\begin{array}{c:c} \boldsymbol{U} & \boldsymbol{0}^{P\times(N-P)} \\ \hdashline \boldsymbol{0}^{(N-P)\times P} & \boldsymbol{I}_{N-P} \end{array}\right] \tag{2.122}$$

where $\boldsymbol{B} \in \mathbb{R}^{N\times P}$, where $\boldsymbol{L} \in \mathbb{R}^{N\times P}$ is a unit lower triangular matrix, where $\boldsymbol{U} \in \mathbb{R}^{P\times P}$ is a unit upper triangular matrix, and where $\boldsymbol{I}_{N-P} \in \mathbb{R}^{(N-P)\times(N-P)}$ is an identity matrix. In addition $\boldsymbol{B}$, $\boldsymbol{L}$ and $\boldsymbol{U}$ satisfy $\boldsymbol{B} = \boldsymbol{L}\boldsymbol{U}$. If $\boldsymbol{A} = \boldsymbol{B}^{\boxdot 2}$, it is easy to prove that $\mathring{\boldsymbol{A}} = \mathring{\boldsymbol{B}}^{\boxdot 2} = (\mathring{\boldsymbol{L}}\mathring{\boldsymbol{U}})^{\boxdot 2}$. Furthermore, $\mathring{\boldsymbol{L}}$ and $\mathring{\boldsymbol{U}}$ admit the following elementary triangular factorization, respectively:

$$\mathring{\boldsymbol{L}} = \prod_{j=1}^{P}\prod_{i=j+1}^{N} \mathring{\boldsymbol{L}}^{(i,j)}(\ell_{i,j}),\ \mathring{\boldsymbol{U}} = \prod_{i=1}^{P}\prod_{j=i+1}^{P} \mathring{\boldsymbol{U}}^{(i,j)}(u_{i,j}) \tag{2.123}$$

where $\mathring{\boldsymbol{L}}^{(i,j)}(\ell_{i,j}) \in \mathbb{R}^{N\times N}$ and $\mathring{\boldsymbol{U}}^{(i,j)}(u_{i,j}) \in \mathbb{R}^{N\times N}$ are elementary lower and upper triangular matrices, respectively. Therefore $\mathring{\boldsymbol{B}}$ can be directly estimated by the proposed JDC$^{+}_{\text{LU-1}}$ and JDC$^{+}_{\text{LU-2}}$methods. The total number of free parameters of $\mathring{\boldsymbol{B}}$ is equal to $P(N-1)$. Then the final output matrix $\boldsymbol{B} \in \mathbb{R}^{N\times P}$ can be obtained be choosing the first P column vectors of $\mathring{\boldsymbol{B}} \in \mathbb{R}^{N\times N}$.

2.4.3 Practical issues

As we mentioned in the previous sections, when the model error of each jointly diagonalizable matrix $\boldsymbol{C}^{(k)}$ is small, the explicit nonnegativity constraint on the matrix $\boldsymbol{A}$ could be needless. We can directly decompose $\boldsymbol{A}$ using the LU factorization as presented in equation (2.43). By means of the JDC criteria $\Psi_1(\boldsymbol{A})$ (1.27) and $\Psi_2(\boldsymbol{A})$ (1.28), each $\boldsymbol{T}^{(i,j)}(t_{i,j})$ in decomposition (2.43) can now be estimated by solving one of the two following Jacobi-like optimization problems, respectively:

$$t_{i,j} = \underset{t_{i,j}}{\operatorname{argmin}}\, \Psi_1(\widehat{\boldsymbol{A}}\boldsymbol{T}^{(i,j)}(t_{i,j})) \tag{2.124}$$

$$t_{i,j} = \underset{t_{i,j}}{\operatorname{argmin}}\, \Psi_2(\widehat{\boldsymbol{A}}\boldsymbol{T}^{(i,j)}(t_{i,j})) \tag{2.125}$$

where $\widehat{\boldsymbol{A}}$ is the estimate of $\boldsymbol{A}$ at the $(it-1)$-th iteration. By a straightforward computation of (2.124), its global optimal $t_{i,j}$ is the minimum of a second degree polynomial, and can be expressed as follows:

$$t_{i,j} = \frac{\sum_k \sum_{n\neq i} \widehat{C}^{(k)}_{i,n}\widehat{C}^{(k)}_{j,n}}{\sum_k \sum_{n\neq i} \widehat{C}^{(k)}_{j,n}\widehat{C}^{(k)}_{j,n}} \tag{2.126}$$

where $\widehat{\boldsymbol{C}}^{(k)} = \widehat{\boldsymbol{A}}^{-1}\boldsymbol{C}^{(k)}\widehat{\boldsymbol{A}}^{-\mathsf{T}}$ is a $(N \times N)$ constant matrix, and where $\widehat{C}^{(k)}_{i,n}$ and $\widehat{C}^{(k)}_{j,n}$ are the (i,n)-th and (j,n)-th components of $\widehat{\boldsymbol{C}}^{(k)}$, respectively. Regarding (2.125), the function to be minimized can be approximated by a fourth degree polynomial function

[Afsari, 2006]:

$$\begin{aligned}\Psi_2(\widehat{\boldsymbol{A}}\boldsymbol{T}^{(i,j)}(t_{i,j})) \approx 4\sum_{k=1}^{K}(\widehat{\boldsymbol{C}}_{j,j}^{(k)})^2\, t_{i,j}^4 - 8\sum_{k=1}^{K}\widehat{\boldsymbol{C}}_{j,j}^{(k)}\widehat{\boldsymbol{C}}_{i,j}^{(k)}\, t_{i,j}^3 \\ + \sum_{k=1}^{K}[4(\widehat{\boldsymbol{C}}_{i,j}^{(k)})^2 + 2(\widehat{\boldsymbol{C}}_{j,j}^{(k)})^2]\, t_{i,j}^2 - 4\sum_{k=1}^{K}\widehat{\boldsymbol{C}}_{j,j}^{(k)}\widehat{\boldsymbol{C}}_{i,j}^{(k)}\, t_{i,j} + \sum_{k=1}^{K}(\widehat{\boldsymbol{C}}_{i,j}^{(k)})^2\end{aligned} \tag{2.127}$$

where $\widehat{C}_{i,j}^{(k)}$ and $\widehat{C}_{j,j}^{(k)}$ are the (i,j)-th and (j,j)-th components of the constant matrix $\widehat{\boldsymbol{C}}^{(k)} = \widehat{\boldsymbol{A}}^{-1}\boldsymbol{C}^{(k)}\widehat{\boldsymbol{A}}^{-\mathsf{T}}$, respectively. The global minimum $t_{i,j}$ of equation (2.127) can be obtained by computing the roots of its derivative. The derivations of equations (2.126) and (2.127) are straightforward therefore omitted. As a matter of fact, the estimations of $t_{i,j}$ by using equations (2.126) and (2.127) are identical to the Afsari's LUJ1D and LUJ2D algorithms [Afsari, 2006], respectively, with a change of sign in $t_{i,j}$. Practically, in each Jacobi-like iteration of $\text{JDC}^{+}_{\text{LU-1}}$, we suggest to compute $t_{i,j}$ by equation (2.126) first. If all the elements in the j-th column vector of $\widehat{\boldsymbol{A}}\boldsymbol{T}^{(i,j)}(t_{i,j})$ have the same sign ς, the j-th column vector of $\widehat{\boldsymbol{A}}^{(\text{new})}$ is replaced by $\varsigma(\widehat{\boldsymbol{a}}_j + t_{i,j}\widehat{\boldsymbol{a}}_i)$, while the other column vectors remain unchanged, where $\widehat{\boldsymbol{a}}_i$ and $\widehat{\boldsymbol{a}}_j$ are the i-th and j-th column vectors of $\widehat{\boldsymbol{A}}$, respectively. Otherwise, $t_{i,j}$ is computed by minimizing (2.98). Then $\widehat{\boldsymbol{A}}$ is updated by means of equation (2.29). Regarding the $\text{JDC}^{+}_{\text{LU-2}}$ algorithm, $t_{i,j}$ is computed in the same manner by minimizing either (2.127) or (2.115). Then $\widehat{\boldsymbol{A}}$ is computed correspondingly.

We also utilize the row balancing scheme for the proposed $\text{JDC}^{+}_{\text{LU-1}}$ and $\text{JDC}^{+}_{\text{LU-2}}$ algorithms, as described in equation (2.74), where the matrix $\widehat{\boldsymbol{C}}^{(k)}$ is replaced by $\widehat{\boldsymbol{C}}^{(k)} = \widehat{\boldsymbol{A}}^{-1}\boldsymbol{C}^{(k)}\widehat{\boldsymbol{A}}^{-\mathsf{T}}$. For conclusion, the pseudo-codes for the $\text{JDC}^{+}_{\text{LU-1}}$ and $\text{JDC}^{+}_{\text{LU-2}}$ algorithms are presented in the appendix C.5 and C.6, respectively.

2.5 The $\text{JDC}^{+}_{\text{ADMM-3}}$ algorithm

The previous three sections presented five Jacobi-like NJDC algorithms. The purpose of this section is to introduce another algorithm based on the Alternating Direction Method of Multipliers (ADMM) approach for solving the NJDC model.

2.5.1 Algorithm derivation

We propose to impose the nonnegativity constraint in the direct-fit JDC criterion (1.29) as follows:

$$\begin{aligned}\Psi_3^{+}(\boldsymbol{A}, \{\boldsymbol{D}^{(k)}\}) = \frac{1}{2}\sum_{k=1}^{K}\left\| \boldsymbol{C}^{(k)} - \boldsymbol{A}\boldsymbol{D}^{(k)}\boldsymbol{A}^{\mathsf{T}} \right\|_F^2 \\ \text{subject to } \boldsymbol{A} \geq \boldsymbol{0}\end{aligned} \tag{2.128}$$

In order to facilitate an efficient use of alternating minimization, we introduce three auxiliary variable $\boldsymbol{A}_1 \in \mathbb{R}^{N\times P}$, $\boldsymbol{A}_2 \in \mathbb{R}^{N\times P}$ and $\boldsymbol{U} \in \mathbb{R}^{N\times P}$. Then the NJDC problem can be solved by minimizing the following cost function:

$$\begin{aligned}\Psi_3^{+}(\boldsymbol{A}_1, \boldsymbol{A}_2, \{\boldsymbol{D}^{(k)}\}) = \frac{1}{2}\sum_{k=1}^{K}\left\| \boldsymbol{C}^{(k)} - \boldsymbol{A}_1\boldsymbol{D}^{(k)}\boldsymbol{A}_2^{\mathsf{T}} \right\|_F^2 \\ \text{subject to } \boldsymbol{A}_1 - \boldsymbol{U} = \boldsymbol{0},\ \boldsymbol{A}_2 - \boldsymbol{U} = \boldsymbol{0},\ \boldsymbol{U} \geq \boldsymbol{0}\end{aligned} \tag{2.129}$$

where $\{\boldsymbol{D}^{(k)}\} \subset \mathbb{R}^{P\times P}$ are a set of K diagonal matrices. The augmented Lagrangian function of (2.129) is expressed as follows:

$$\begin{aligned}\mathrm{L_A}(\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{U}, \{\boldsymbol{D}^{(k)}\}, \boldsymbol{\Pi}_1, \boldsymbol{\Pi}_2) &= \Psi_3^{+}(\boldsymbol{A}_1, \boldsymbol{A}_2, \{\boldsymbol{D}^{(k)}\}) + f(\boldsymbol{U}) \\ &+ \boldsymbol{\Pi}_1 \bullet (\boldsymbol{A}_1 - \boldsymbol{U}) + \boldsymbol{\Pi}_2 \bullet (\boldsymbol{A}_2 - \boldsymbol{U}) + \frac{\alpha}{2}\left\|\boldsymbol{A}_1 - \boldsymbol{U}\right\|_F^2 + \frac{\beta}{2}\left\|\boldsymbol{A}_2 - \boldsymbol{U}\right\|_F^2\end{aligned} \tag{2.130}$$

where $f(\cdot)$ is the indicator function on the set of $(N \times P)$ nonnegative matrices, where $\boldsymbol{\Pi}_1 \in \mathbb{R}^{N\times P}$ and $\boldsymbol{\Pi}_2 \in \mathbb{R}^{N\times P}$ are the Lagrangian multipliers, and where $\alpha > 0$ and $\beta > 0$ are penalty parameters for the constraints $\boldsymbol{A}_1 - \boldsymbol{U} = \boldsymbol{0}$ and $\boldsymbol{A}_2 - \boldsymbol{U} = \boldsymbol{0}$, respectively. For the matrices $\boldsymbol{Y}$ and $\boldsymbol{Z}$ of the same dimension, the scalar product $\bullet$ is defined as the sum of all the element-wise products:

$$\boldsymbol{Y} \bullet \boldsymbol{Z} = \sum_i \sum_j y_{i,j} z_{i,j} \tag{2.131}$$

where $y_{i,j}$ and $z_{i,j}$ are the (i,j)-th components of $\boldsymbol{Y}$ and $\boldsymbol{Z}$, respectively.

The ADMM for (2.129) is derived by successively minimizing the augmented Lagrangian function (2.130) with respect to $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\boldsymbol{U}$ and $\{\boldsymbol{D}^{(k,it)}\}$, one at a time while fixing others at their most recent values, for example:

$$\boldsymbol{A}_1^{(it+1)} = \underset{\boldsymbol{A}_1}{\operatorname{argmin}}\, \mathrm{L_A}(\boldsymbol{A}_1, \boldsymbol{A}_2^{(it)}, \boldsymbol{U}^{(it)}, \{\boldsymbol{D}^{(k,it)}\}, \boldsymbol{\Pi}_1^{(it)}, \boldsymbol{\Pi}_2^{(it)}) \tag{2.132}$$

$$\boldsymbol{A}_2^{(it+1)} = \underset{\boldsymbol{A}_2}{\operatorname{argmin}}\, \mathrm{L_A}(\boldsymbol{A}_1^{(it+1)}, \boldsymbol{A}_2, \boldsymbol{U}^{(it)}, \{\boldsymbol{D}^{(k,it)}\}, \boldsymbol{\Pi}_1^{(it)}, \boldsymbol{\Pi}_2^{(it)}) \tag{2.133}$$

$$\boldsymbol{U}^{(it+1)} = \underset{\boldsymbol{U}}{\operatorname{argmin}}\, \mathrm{L_A}(\boldsymbol{A}_1^{(it+1)}, \boldsymbol{A}_2^{(it+1)}, \boldsymbol{U}, \{\boldsymbol{D}^{(k,it)}\}, \boldsymbol{\Pi}_1^{(it)}, \boldsymbol{\Pi}_2^{(it)}) \tag{2.134}$$

$$\{\boldsymbol{D}^{(k,it+1)}\} = \underset{\boldsymbol{D}^{(k)}}{\operatorname{argmin}}\, \mathrm{L_A}(\boldsymbol{A}_1^{(it+1)}, \boldsymbol{A}_2^{(it+1)}, \boldsymbol{U}^{(it+1)}, \{\boldsymbol{D}^{(k)}\}, \boldsymbol{\Pi}_1^{(it)}, \boldsymbol{\Pi}_2^{(it)}) \tag{2.135}$$

and then updating the multipliers $\boldsymbol{\Pi}_1$ and $\boldsymbol{\Pi}_2$, where it denotes the iteration number. Specifically, these steps can be written in closed form as follows:

$$\begin{aligned}\boldsymbol{A}_1^{(it+1)} = &\left(\sum_{k=1}^{K} \boldsymbol{C}^{(k)} \boldsymbol{A}_2^{(it)} \boldsymbol{D}^{(k,it)} + \alpha \boldsymbol{U}^{(it)} - \boldsymbol{\Pi}_1^{(it)}\right) \times \\ &\left(\sum_{k=1}^{K} \boldsymbol{D}^{(k,it)} (\boldsymbol{A}_2^{(it)})^{\mathsf{T}} \boldsymbol{A}_2^{(it)} \boldsymbol{D}^{(k,it)} + \alpha \boldsymbol{I}_P\right)^{-1}\end{aligned} \tag{2.136}$$

$$\begin{aligned}\boldsymbol{A}_2^{(it+1)} = &\left(\sum_{k=1}^{K} \boldsymbol{C}^{(k)} \boldsymbol{A}_1^{(it+1)} \boldsymbol{D}^{(k,it)} + \beta \boldsymbol{U}^{(it)} - \boldsymbol{\Pi}_2^{(it)}\right) \times \\ &\left(\sum_{k=1}^{K} \boldsymbol{D}^{(k,it)} (\boldsymbol{A}_1^{(it+1)})^{\mathsf{T}} \boldsymbol{A}_1^{(it+1)} \boldsymbol{D}^{(k,it)} + \beta \boldsymbol{I}_P\right)^{-1}\end{aligned} \tag{2.137}$$

$$\boldsymbol{U}^{(it+1)} = \mathrm{P}_{+}\left\{\left(\alpha \boldsymbol{A}_1^{(it+1)} + \beta \boldsymbol{A}_2^{(it+1)} + \boldsymbol{\Pi}_1^{(it)} + \boldsymbol{\Pi}_2^{(it)}\right)/(\alpha + \beta)\right\} \tag{2.138}$$

$$\boldsymbol{D}^{(k,it+1)} = \mathrm{Diag}\left\{\left(\boldsymbol{A}_1^{(it+1)} \odot \boldsymbol{A}_2^{(it+1)}\right)^{\sharp} \mathrm{vec}\,(\boldsymbol{C}^{(k)})\right\} \tag{2.139}$$

$$\boldsymbol{\Pi}_1^{(it+1)} = \boldsymbol{\Pi}_1^{(it)} + \gamma\alpha(\boldsymbol{A}_1^{(it+1)} - \boldsymbol{U}^{(it+1)}) \tag{2.140}$$

$$\boldsymbol{\Pi}_2^{(it+1)} = \boldsymbol{\Pi}_2^{(it)} + \gamma\beta(\boldsymbol{A}_2^{(it+1)} - \boldsymbol{U}^{(it+1)}) \tag{2.141}$$

where $\boldsymbol{I}_P$ is a $(P \times P)$ identity matrix, where $\text{P}_+\{\cdot\}$ sets all negative values of the input matrix to zero, where $\text{Diag}(\cdot)$ returns a diagonal matrix whose diagonal elements are given by the input vector, where $\sharp$ denotes the pseudo-inverse operator, where $\boxdot$ denotes the Hadamard product and $\odot$ denotes the Khatri-Rao product, where $\text{vec}(\cdot)$ reshapes a matrix into a column vector by stacking its columns vertically, and where $\gamma \in [0, 1.618]$ [Xu et al., 2012]. The pseudo-code of the $\text{JDC}^{+}_{\text{ADMM-3}}$ algorithm is summarized in appendix C.7.

2.5.2 Convergence to KKT Points

In this section, following the analysis of the convergence of the ADMM based NMF algorithm [Xu et al., 2012], we provide a preliminary study on the convergence of the proposed ADMM algorithm by studying the Karush–Kuhn–Tucker (KKT) conditions of problem (2.129). The KKT conditions of a nonlinear problem are first order necessary conditions for a solution to be optimal, provided that some regularity conditions are satisfied. Consider the following nonlinear optimization problem:

$$\begin{aligned} &\text{minimize}_x\ f(x) \\ &\text{subject to} \qquad h_i(x) \leq 0,\ i = 1, 2, \ldots, m \\ &\qquad\qquad\quad\ \ g_j(x) = 0,\ j = 1, 2, \ldots, r \end{aligned} \tag{2.142}$$

where x is the optimization variable, where f is the objective function, where h_i $(i = 1, \ldots, m)$ are the inequality constraint functions, and where g_j $(j = 1, \ldots, r)$ are the equality constraint functions. Suppose that the objective function f and the constraint functions h_i and g_j are continuously differentiable at a point x^*. We say x^* is a KKT point of problem (2.142) if it satisfies the following KKT conditions [Boyd and Vandenberghe, 2004]:

$$\frac{\partial f(x^*)}{\partial x^*} + \sum_{i=1}^{m} u_i \frac{\partial h_i(x^*)}{\partial x^*} + \sum_{j=1}^{r} v_j \frac{\partial g_j(x^*)}{\partial x^*} = 0 \tag{2.143}$$

$$h_i(x^*) \leq 0,\ \forall\ i \in \{1, \ldots, m\} \tag{2.144}$$

$$g_j(x^*) = 0,\ \forall\ j \in \{1, \ldots, r\} \tag{2.145}$$

$$u_i \geq 0,\ \forall\ i \in \{1, \ldots, m\} \tag{2.146}$$

$$u_i h_i(x^*) = 0,\ \forall\ i \in \{1, \ldots, m\} \tag{2.147}$$

where u_i $(i = 1, \ldots, m)$ and v_j $(j = 1, \ldots, r)$ are constants. Then the KKT point x^* is a stationary point with respect to x of the Lagrangian function of problem (2.142).

Regarding the proposed problem (2.129), let us define the following sextuple point:

$$\boldsymbol{Z} \stackrel{\text{def}}{=} (\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{U}, \{\boldsymbol{D}^{(k)}\}, \boldsymbol{\Pi}_1, \boldsymbol{\Pi}_2) \tag{2.148}$$

A point $\boldsymbol{Z}$ is a KKT point of problem (2.129) if it satisfies the KKT conditions for

problem (2.129) as follows:

$$\sum_{k=1}^{K}(\boldsymbol{C}^{(k)} - \boldsymbol{A}_1\boldsymbol{D}^{(k)}\boldsymbol{A}_2^{\mathsf{T}})(\boldsymbol{A}_2\boldsymbol{D}^{(k)}) - \boldsymbol{\Pi}_1 = \boldsymbol{0} \tag{2.149}$$

$$\sum_{k=1}^{K}(\boldsymbol{C}^{(k)} - \boldsymbol{A}_2\boldsymbol{D}^{(k)}\boldsymbol{A}_1^{\mathsf{T}})(\boldsymbol{A}_1\boldsymbol{D}^{(k)}) - \boldsymbol{\Pi}_2 = \boldsymbol{0} \tag{2.150}$$

$$(\boldsymbol{A}_2 \odot \boldsymbol{A}_1)^{\mathsf{T}}\left(\boldsymbol{c}(k) - (\boldsymbol{A}_2 \odot \boldsymbol{A}_1)\boldsymbol{d}^{(k)}\right) = \boldsymbol{0} \tag{2.151}$$

$$\boldsymbol{A}_1 - \boldsymbol{U} = \boldsymbol{0} \tag{2.152}$$

$$\boldsymbol{A}_2 - \boldsymbol{U} = \boldsymbol{0} \tag{2.153}$$

$$(\boldsymbol{\Pi}_1 + \boldsymbol{\Pi}_2) \leq \boldsymbol{0} \leq \boldsymbol{U} \tag{2.154}$$

$$(\boldsymbol{\Pi}_1 + \boldsymbol{\Pi}_2) \boxdot \boldsymbol{U} = \boldsymbol{0} \tag{2.155}$$

where $\boxdot$ denotes the Hadamard product and $\odot$ denotes the Khatri-Rao product, where $\boldsymbol{c}^{(k)} = \text{vec}(\boldsymbol{C}^{(k)})$, and where $\boldsymbol{d}^{(k)} \in \mathbb{R}^P$ is a column vector containing the diagonal elements of the diagonal matrix $\boldsymbol{D}^{(k)}$. Then we obtain the following proposition:

Proposition 2.9. *Let* $\{\boldsymbol{Z}^{(it)}\}_{it=1}^{\infty}$ *be a sequence generated by the proposed ADMM algorithm* (2.136) - (2.141) *that satisfies the following condition:*

$$\lim_{it\to\infty}(\boldsymbol{Z}^{(it+1)} - \boldsymbol{Z}^{(it)}) = \boldsymbol{0} \tag{2.156}$$

Then any accumulation point of $\{\boldsymbol{Z}^{(it)}\}_{it=1}^{\infty}$ *is a KKT point of problem* (2.129).

The proof of proposition 2.9 is provided in appendix A.7. The following corollary can be derived immediately.

Corollary 2.1. *Whenever* $\{\boldsymbol{Z}^{(it)}\}_{it=1}^{\infty}$ *converges, it converges to a KKT point.*

The above preliminary result provides some assurance on the convergence behavior of the ADMM algorithm applied to the non-convex NJDC problem. Further theoretical studies will be considered in future work.

2.6 Numerical complexity analysis

This section is devoted to compute the numerical complexities of the proposed methods. The numerical complexity is analyzed in terms of the number of floating point operations (flops). A flop is defined as a multiplication followed by an addition. In practice, only the number of multiplications, required to identify the joint transformation matrix $\boldsymbol{A} \in \mathbb{R}_+^{N\times P}$ from K symmetric matrices $\boldsymbol{C}^{(k)} \in \mathbb{R}^{N\times N}$, $k \in \{1,\ldots,K\}$, is considered; which does not affect the order of magnitude of the numerical complexity [Kachenoura et al., 2008, Albera et al., 2010, Albera et al., 2012]. The computational complexities of the proposed methods are given in table 2.1 and explained with more details afterwards.

In the section of numerical simulations, the proposed methods are compared with five classical JDC methods: the Alternating Columns and Diagonal Center (ACDC) algorithm [Yeredor, 2002], the Fast Frobenius DIAGonalization (FFDIAG) algorithm [Ziehe et al., 2004], the Coupled LU factorization based algorithm (CLU) [Maurandi et al.,

2013], and two algorithms based on the general LU factorization and the minimization of two indirect criteria, namely LUJ1D and LUJ2D [Afsari, 2006]. Regarding the CLU method [Maurandi et al., 2013], the JDC criterion (1.27) is approximated by a sequence of (2×2) sub-criteria. Therefore, in each Jacobi-like iteration, one elementary lower triangular matrix and one elementary upper triangular matrix can be computed together by EVD, leading to a lower numerical complexity. Furthermore, a NJDC method based on a square change of variable and Levenberg Marquardt optimization [Coloigner et al., 2014c], namely $\text{LM}^{+}_{\text{sym}}$, is also included in the comparison. Their complexities are also given in table 2.1.

2.6.1 The $\text{JDC}^{+}_{\text{LU-3}}$ algorithm

The computation of the coefficient matrix $\boldsymbol{G}$ in the cost function (2.36) requires $(33N^2 + 7N)K$ flops. Computing the parameter $t_{i,j}$ by calculating the roots of the 8-th degree polynomial function (2.36) needs $8^3 = 512$ flops. Nevertheless, if $t_{i,j}$ is obtained by minimizing the unconstrained cost function (2.44), which is a 4-th degree polynomial function, it costs $(15N^2+4N)K+64$ flops. Hence, the complexity of the AC phase, which includes $P(N-1)$ parameters, is between $((15N^2+4N)K+64)P(N-1)$ and $((33N^2+7N)K+512)P(N-1)$ flops, depending on the number of unconstrained optimizations is performed. The DC phase consists of estimating K diagonal matrices by means of equation (2.41), and it involves $N^2PK + N^2P^2 + N^2P + NP^2 + P^3 + P^2$ flops. Hence, the magnitude of the total numerical complexities of one AC phase and one DC phase is $\mathcal{O}(N^3PK)$. The full expression of the complexity of the $\text{JDC}^{+}_{\text{LU-3}}$ algorithm is given in table 2.1.

2.6.2 The $\text{iJDC}^{+}_{\text{LU-1}}$ and $\text{iJDC}^{+}_{\text{QR-1}}$ algorithms

The $\text{iJDC}^{+}_{\text{LU-1}}$ and $\text{iJDC}^{+}_{\text{QR-1}}$ algorithms only work on a square matrix $\boldsymbol{A}$. Therefore, here we assume that $N = P$. For both algorithms, the inverses of the K matrices $\boldsymbol{C}^{(k)}$ cost N^3K flops, the initialization of $\widehat{\boldsymbol{C}}^{(k)}$ in equations (2.51) and (2.61) requires $2N^3K$ flops. Regarding the estimation of the elementary triangular matrices, in each Jacobi-like iteration, the calculation of the parameter $t_{i,j}$ by minimizing the cost function (2.53) needs $(5N^2+12N-8)K+64$ flops. The calculation cost of $\widehat{\boldsymbol{A}}^{(\text{new})}$, $\widehat{\boldsymbol{B}}^{(\text{new})}$ and $\widehat{\boldsymbol{C}}^{(k,\text{new})}$, with $k \in \{1, \cdots, K\}$, is $4N+(4N+1)K$ flops. Otherwise, if the parameter $t_{i,j}$ is computed without the nonnegativity constraint by means of equation (2.72), it requires $2(N-1)K$ flops. In this situation, the computation load of $\widehat{\boldsymbol{A}}^{(\text{new})}$ and $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ is $2NK + N$ flops. Regarding the estimation of the Given rotation matrices, in each Jacobi-like iteration, the complexity of calculating the parameters $\theta_{i,j}$ by minimizing the cost function (2.65) is equal to $(5N^2+3N+29)K+3375$ flops, and the estimation of $\widehat{\boldsymbol{A}}^{(\text{new})}$, $\widehat{\boldsymbol{B}}^{(\text{new})}$ and $\widehat{\boldsymbol{C}}^{(k,\text{new})}$, with $k \in \{1, \cdots, K\}$, costs $10N + (12N + 20)K$ flops. Otherwise, if the parameter $\theta_{i,j}$ is obtained by minimizing the unconstrained cost function (2.73), it costs $2K + 8$ flops. Then the update of $\widehat{\boldsymbol{A}}^{(\text{new})}$ and $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ needs $4NK + 2N$ flops. In addition, the row balancing scheme defined in equation (2.74) requires $3N^2K + N^2$ flops. Therefore, the magnitude of global numerical complexities of $\text{iJDC}^{+}_{\text{LU-1}}$ and $\text{iJDC}^{+}_{\text{QR-1}}$ for estimating all the $N(N-1)$ parameters is between $\mathcal{O}(N^3K)$ and $\mathcal{O}(N^4K)$. Their detailed expressions are given in table 2.1.

2.6.3 The $\text{JDC}^+_{\text{LU-1}}$ and $\text{JDC}^+_{\text{LU-2}}$ algorithms

For both algorithms, the initialization of $\widehat{\boldsymbol{C}}^{(k)}$ in equations (2.86) requires $2N^3K$ flops. Regarding the $\text{JDC}^+_{\text{LU-1}}$ algorithm, in order to compute one parameter $t_{i,j}$ in one Jacobi-like iteration, computing the coefficient matrix $\boldsymbol{Q}_E$ in the cost function (2.98) involves $(23N^2-4N-3)K+N^3+N^2+N$ flops. Calculating the roots of the rational function (2.98) costs 729 flops. The computation load of $\widehat{\boldsymbol{A}}^{(\text{new})}$, $\widehat{\boldsymbol{B}}^{(\text{new})}$ and $\widehat{\boldsymbol{C}}^{(k,\text{new})}$, with $k \in \{1,\cdots,K\}$, is $4N+(2N^2+5N+2)K$ flops. Otherwise, if the parameter $t_{i,j}$ is computed without the nonnegativity constraint by means of equation (2.126), its calculation load is $2(N-1)K$ flops. Accordingly, the computation cost of $\widehat{\boldsymbol{A}}^{(\text{new})}$ and $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ is $2NK+N$ flops. The row balancing scheme needs $3N^2K+NP$ flops. Thus, the magnitude of global numerical complexity of $\text{JDC}^+_{\text{LU-1}}$ consisting of estimating all the $P(N-1)$ parameters is between $\mathcal{O}(N^2PK)$ and $\mathcal{O}(N^4P+N^3PK)$, depending on the number of unconstrained optimizations is performed. Regarding the $\text{JDC}^+_{\text{LU-2}}$ algorithm, in each Jacobi-like iteration, the estimation of $t_{i,j}$ by minimizing the cost function (2.115) requires $(5N^3+32N^2+11N-5)K+2N^3+5N^2+4N+1331$ flops. Computing $\widehat{\boldsymbol{A}}^{(\text{new})}$, $\widehat{\boldsymbol{B}}^{(\text{new})}$ and $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ involves $4N+(2N^2+5N+2)K$ flops. Nevertheless, if $t_{i,j}$ is estimated by minimizing the unconstrained cost function (2.127), it costs $8K+64$ flops. In this case, updating $\widehat{\boldsymbol{A}}^{(\text{new})}$ and $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ costs $2NK+N$ flops. Consequently, considering all the $P(N-1)$ parameters, the magnitude of the total numerical complexities of $\text{JDC}^+_{\text{LU-2}}$ is between $\mathcal{O}(N^2PK)$ and $\mathcal{O}(N^4PK)$. The full expressions of the complexities of the $\text{JDC}^+_{\text{LU-1}}$ and $\text{JDC}^+_{\text{LU-2}}$ algorithms are given in table 2.1.

2.6.4 The $\text{JDC}^+_{\text{ADMM-3}}$ algorithm

Computing $\boldsymbol{A}_1^{(it+1)}$ according to equation (2.136) requires $(N^2P+NP+2P^2)K+NP^2+P^3+NP+P$ flops. The complexity of $\boldsymbol{A}_2^{(it+1)}$ is the same as that of $\boldsymbol{A}_1^{(it+1)}$. Updating $\boldsymbol{U}^{(it+1)}$ by means of equation (2.138) involves $3NP$ flops. According to equation (2.139), the calculation of K diagonal matrices $\boldsymbol{D}^{(k,it+1)}$ needs $N^2PK+N^2P^2+N^2P+NP^2+P^3+P^2$ flops. Finally, the computation of the multipliers $\boldsymbol{\Pi}_1^{(it+1)}$ and $\boldsymbol{\Pi}_2^{(it+1)}$ both require NP flops, following equations (2.140) and (2.141), respectively. Therefore the magnitude of the total numerical complexities of one iteration of $\text{JDC}^+_{\text{ADMM-3}}$ is $\mathcal{O}(N^2PK)$, while its full expression is given in table 2.1.

From table 2.1, we can observe that in order to estimate a square joint transformation matrix $\boldsymbol{A} \in \mathbb{R}_+^{N\times N}$, the classical JDC methods ACDC, CLU, FFDIAG, LUJ1D and LUJ2D achieve lower numerical complexities per sweep/iteration than most of the nonnegativity constrained methods. The magnitude of the complexities of these classical JDC methods is $\mathcal{O}(N^3K)$. Among the proposed methods, the $\text{JDC}^+_{\text{ADMM-3}}$ algorithm gives the lowest complexity, which is also of magnitude $\mathcal{O}(N^3K)$. The complexities of the $\text{JDC}^+_{\text{LU-3}}$, $\text{iJDC}^+_{\text{LU-1}}$ and $\text{iJDC}^+_{\text{QR-1}}$ algorithms are at most one magnitude higher than that of $\text{JDC}^+_{\text{ADMM-3}}$, which are of magnitude $\mathcal{O}(N^4K)$ in the worst case when all the parameters are computed by the nonnegativity constrained optimizations. The $\text{JDC}^+_{\text{LU-1}}$ and $\text{JDC}^+_{\text{LU-2}}$ algorithms require a large amount of computation, whose magnitudes of complexities are $\mathcal{O}(N^5+N^4K)$ and $\mathcal{O}(N^5K)$ in the worst case, respectively. Nevertheless, their complexities are still much lower that of the Levenberg Marquardt method, namely LM^+_{sym}, when K is much larger than N. The complexity of LM^+_{sym} is of magni-

tude $\mathcal{O}(N^3K^3+N^6)$ due to the heavy computation of the modification of the Hessian. A similar observation can be made from table 2.1 when a non-square joint transformation matrix $\boldsymbol{A} \in \mathbb{R}_+^{N\times P}$ with $N > P$ is encountered in a JDC problem.

	Computational complexity (flops) (N, P): the dimension of the joint transformation matrix $\boldsymbol{A}$; K: the number of the $(N \times N)$ symmetric matrices $\boldsymbol{C}^{(k)}$; J_i, $i \in \{1, 2, 3, 4, 5, 6, 12\}$: the number of sweeps (or iterations) used in the i-th method; J_i^{rb}, $i \in \{4, 5, 7, 8, 9, 10\}$: the number of row balancing schemes used in the i-th method; J_i^+ and J_i^*, $i \in \{7, \ldots, 11\}$: the number of sweeps with and without the nonnegativity constraint used in the i-th method, respectively; CLU, FFDIAG, LUJ1D, LUJ2D, iJDC$^+_{\text{LU-1}}$ and iJDC$^+_{\text{QR-1}}$ only work on the $N = P$ case.	
	$N = P$	$N > P$
ACDC	$(13/3N^3K + 3N^4 + 2N^2K + N^3 + N^2)J_1$	$(13/3N^2PK + 3N^3P + 2NPK + NP^2 + P^2)J_1$
CLU	$(2N^3K + 2.5N^2K + 1.5N^3 + 18.5N^2 - 4.5NK - 20N)J_2$	
FFDIAG	$(2N^3K + N^3 + 2N^2K + 4N^2 - 4N)J_3$	
LUJ1D	$(4N^3K - 6N^2K + N^3 - N^2 + 2NK)J_4 + (3N^2K + N^2)J_4^{rb}$	
LUJ2D	$(2N^3K + 6N^2K + N^3 + 63N^2 - 8NK - 64N)J_5 + (3N^2K + N^2)J_5^{rb}$	
LM$^+_{\text{sym}}$	$(\ell_h(N+K)^3N^3 + \ell_h(N+K)^2N^2 + 4N^6 + N^5 + N^5K + 11N^4 + 3K^2N^2 + 3N^3K + 8N^3 + 3N^2K + N^2)J_6$, where ℓ_h is the number of iterations computing the modification of the Hessian	$(\ell_h(N+K)^3P^3 + \ell_h(N+K)^2P^2 + N^4P^2 + 2N^2P^4 + N^3P^3 + N^3P^2 + N^2P^3K + 8N^2P^2 + 3NP^3 + 3K^2P^2 + NKP^2 + 2N^2KP + 6N^2P + 2NP^2 + 2KP^2 + NKP + P^2)J_6$
iJDC$^+_{\text{LU-1}}$	$3N^3K + (4N^3K - 6N^2K + N^3 - N^2 + 2NK)J_7^* + (5N^4K + 11N^3K - 23N^2K + 4N^3 + 60N^2 + 7NK - 64N)J_7^+ + (3N^2K + N^2)J_7^{rb}$	
iJDC$^+_{\text{QR-1}}$	$3N^3K + (4N^3K - 4N^2K + 1.5N^3 + 2.5N^2 - 4N)J_8^* + (5N^4K + 15.5N^3K + 5.5N^2K + 7N^3 + 1712.5N^2 - 21NK - 1719.5N)J_8^+ + (3N^2K + N^2)J_8^{rb}$	
JDC$^+_{\text{LU-1}}$	$(4N^3K - 6N^2K + N^3 - N^2 + 2NK)J_9^* + (N^5 + 25N^4K - 22N^3K - 2N^2K + 4N^3 + 724N^2 + NK - 729N)J_9^+ + (3N^2K + N^2)J_9^{rb}$	$(4N^2PK - 6NPK + N^2P - NP + 2PK)J_9^* + (N^4P + 25N^3PK + 2N^3K - 24N^2PK - 2NPK + 4N^2P + 724NP + PK - 729P)J_9^+ + (3N^2K + NP)J_9^{rb}$
JDC$^+_{\text{LU-2}}$	$(2N^3K + 6N^2K + N^3 + 63N^2 - 8NK - 64N)J_{10}^* + (5N^5K + 29N^4K + 2N^5 - 16N^3K + 3N^4 - 19N^2K + 3N^3 + 1323N^2 + 3NK - 1331N)J_{10}^+ + (3N^2K + N^2)J_{10}^{rb}$	$(2N^2PK + 6NPK + N^2P + 63NP - 8PK - 64P)J_{10}^* + (5N^4PK + 29N^3PK + 2N^4P + 2N^3K - 18N^2PK + 3N^3P - 19NPK + 3N^2P + 1323NP + 3PK - 1331P)J_{10}^+ + (3N^2K + NP)J_{10}^{rb}$
JDC$^+_{\text{LU-3}}$	$(15N^4K - 10N^3K + N^4 - 4N^2K + 3N^3 + 65N^2 - 64N)J_{11}^* + (33N^4K - 25N^3K + N^4 - 7N^2K + 3N^3 + 513N^2 - 512N)J_{11}^+$	$(15N^3PK - 10N^2PK + N^2P^2 - 4NPK + N^2P + NP^2 + P^3 + P^2 + 64NP - 64P)J_{11}^* + (33N^3PK - 25N^2PK + N^2P^2 - 7NPK + N^2P + NP^2 + P^3 + 512NP + P^2 - 512P)J_{11}^+$
JDC$^+_{\text{ADMM-3}}$	$(3N^3K + N^4 + 6N^2K + 7N^3 + 8N^2 + 2N)J_{12}$	$(3N^2PK + N^2P^2 + 2NPK + 4P^2K + N^2P + 3NP^2 + 3P^3 + 7NP + P^2 + 2P)J_{12}$

Table 2.1: Numerical complexities of the proposed methods and of ACDC [Yeredor, 2002], CLU [Maurandi et al., 2013], FFDIAG [Ziehe et al., 2004], LUJ1D, LUJ2D [Afsari, 2006] and LM$^+_{\text{sym}}$ [Coloigner et al., 2014c].

Chapter 3

Numerical simulations

In this chapter, the performance of the proposed algorithms is evaluated through simulated data fitting the NJDC model. Several experiments are designed to study the convergence property, the influence of SNR, the impact of the number K of the input symmetric matrices $\boldsymbol{C}^{(k)}$, the effect of the coherence of the column vectors of the joint mixing matrix $\boldsymbol{A}$ and the influence of the modulus of uniqueness of JDC. In each experiment, the performance is analyzed for both estimating a square joint transformation matrix $\boldsymbol{A}$ and a non-square matrix $\boldsymbol{A}$. The proposed algorithms are compared with five classical nonorthogonal JDC methods, which do not exploit the nonnegativity property of $\boldsymbol{A}$, namely the Alternating Columns and Diagonal Center (ACDC) algorithm [Yeredor, 2002], the Fast Frobenius DIAGonalization (FFDIAG) algorithm [Ziehe et al., 2004], the Coupled LU factorization based algorithm (CLU) [Maurandi et al., 2013], and two algorithms based on the general LU factorization and the minimization of two indirect criteria, namely LUJ1D and LUJ2D [Afsari, 2006]. In addition, a NJDC method based on a square change of variable and Levenberg Marquardt optimization [Coloigner et al., 2014c], namely $\mathrm{LM}^{+}_{\mathrm{sym}}$, is also included in the comparison.

3.1 Experimental setup

K synthetic symmetric matrices $\boldsymbol{C}^{(k)} \in \mathbb{R}^{N\times N}$ with $k \in \{1,\ldots,K\}$ are generated randomly according to the NJDC model (1.51). When used without further specification, all the algorithms are manipulated under the following conditions:

1. Model generation: the joint transformation matrix $\boldsymbol{A} \in \mathbb{R}_{+}^{N\times P}$ is randomly drawn from a uniform distribution on the interval $[0,1]$. The diagonal elements of the K diagonal matrices $\boldsymbol{D}^{(k)}$ with $k \in \{1,\ldots,K\}$ are drawn from the zero-mean unit-variance Gaussian distribution. Each pure matrix $\boldsymbol{C}^{(k)}$ is perturbed by a random symmetric noise matrix as follows:

$$\boldsymbol{C}_N^{(k)} = \frac{\boldsymbol{C}^{(k)}}{\|\boldsymbol{C}^{(k)}\|_F} + \sigma^{(k)} \frac{\boldsymbol{R}^{(k)}(\boldsymbol{R}^{(k)})^{\mathsf{T}}}{\|\boldsymbol{R}^{(k)}(\boldsymbol{R}^{(k)})^{\mathsf{T}}\|_F} \tag{3.1}$$

   where $\sigma^{(k)}$ is a scalar controlling the noise level, and where the components of the noise matrix $\boldsymbol{R}^{(k)} \in \mathbb{R}^{N\times N}$ are drawn from the zero-mean unit-variance Gaussian distribution. Then the SNR is defined by $\mathrm{SNR} = -20\log_{10}(\sigma^{(k)})$.

2. Initialization: in each Monte Carlo trial, all the algorithms are initialized by the absolute value of the inverse of the output of the Uniformly Weighted Exhaustive Diagonalization with Gauss itErations (UWEDGE) algorithm [Tichavský and Yeredor, 2009] due to its fast speed.

3. Afsari's row balancing scheme: the LUJ1D, LUJ2D, iJDC$^{+}_{\text{LU-1}}$, iJDC$^{+}_{\text{QR-1}}$, JDC$^{+}_{\text{LU-1}}$ and JDC$^{+}_{\text{LU-2}}$ algorithms perform the row balancing scheme once per each run of five sweeps.

4. Stopping criterion: all the algorithms stop either when the relative error of the corresponding criterion between two successive sweeps of a Jacobi-like method (or iterations of an alternating minimization method) is less than 10^{-4} or when the number of sweeps (iterations) exceeds 500.

5. Performance measurement: the performance is measured by means of the error between the true joint transformation matrix $\boldsymbol{A}$ and the estimate $\widehat{\boldsymbol{A}}$, and the numerical complexity. Note that the numerical complexity can be approximately measured by computing the CPU time. However, the CPU time is computer-dependent and also varies according to the operating system. In this report, the simulations are carried out in Matlab v7.14 on Mac OS X and run on Intel Quad-Core CPU 2.8 GHz with 32 GigaByte (GB) memory.

   In order to assess the estimation accuracy of $\boldsymbol{A}$, we use the following scale-invariant and permutation-invariant distance [Coloigner et al., 2014a,Coloigner et al., 2014c]:

$$\alpha(\boldsymbol{A}, \widehat{\boldsymbol{A}}) = \frac{1}{N} \sum_{n=1}^{N} \min_{(n,n') \in I_n^2} d(\boldsymbol{a}_n, \widehat{\boldsymbol{a}}_{n'}) \tag{3.2}$$

   where $\boldsymbol{a}_n$ and $\widehat{\boldsymbol{a}}_{n'}$ are the n-th column vector of $\boldsymbol{A}$ and the n'-th column vector of $\widehat{\boldsymbol{A}}$, respectively. I_n^2 is a set of indices defined recursively by $I_1^2 = \{1, \cdots, N\} \times \{1, \cdots, N\}$, and $I_{n+1}^2 = I_n^2 - J_n^2$ where $J_n^2 = \text{argmin}_{(n,n') \in I_n^2} d(\boldsymbol{a}_n, \widehat{\boldsymbol{a}}_{n'})$. In addition, $d(\boldsymbol{a}_n, \widehat{\boldsymbol{a}}_{n'})$ is the pseudo-distance between two column vectors defined by [Albera et al., 2004]:

$$d(\boldsymbol{a}_n, \widehat{\boldsymbol{a}}_{n'}) = 1 - \frac{\|\boldsymbol{a}_n^{\mathsf{T}} \widehat{\boldsymbol{a}}_{n'}\|^2}{\|\boldsymbol{a}_n\|^2 \|\widehat{\boldsymbol{a}}_{n'}\|^2} \tag{3.3}$$

   The criterion (3.2) is an upper bound of the optimal permutation-invariant criterion. It avoids the burdensome computation of all the permutations. A small value of (3.2) means a good performance in the sense that $\widehat{\boldsymbol{A}}$ is close to $\boldsymbol{A}$. Moreover, we repeat all the experiments with 500 Monte Carlo trials.

3.2 Convergence property

In this section, we aim at studying the convergences of the proposed algorithms, as well as those of the ACDC, CLU, FFDIAG, LUJ1D, LUJ2D and LM$^{+}_{\text{sym}}$ algorithms.

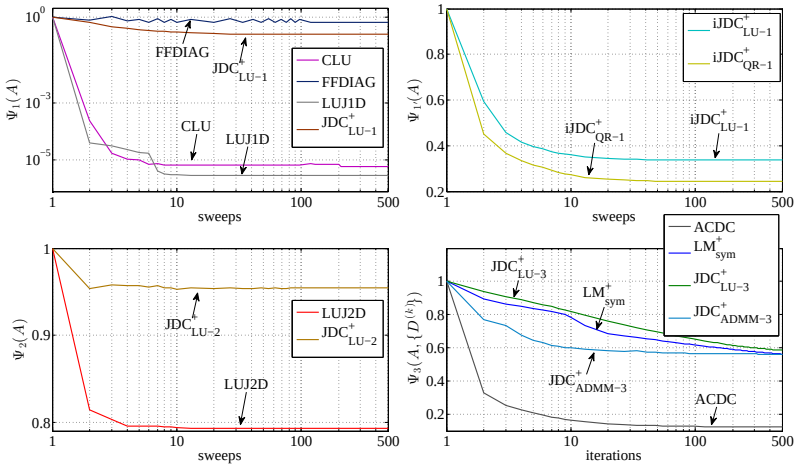
3.2.1 Test on square matrices

In this experiment, the dimensions of the square joint transformation matrix $\boldsymbol{A} \in \mathbb{R}_{+}^{N \times P}$ are set to $N = 3$ and $P = 3$. The number of the input matrices $\boldsymbol{C}^{(k)}$ is fixed to $K = 5$.

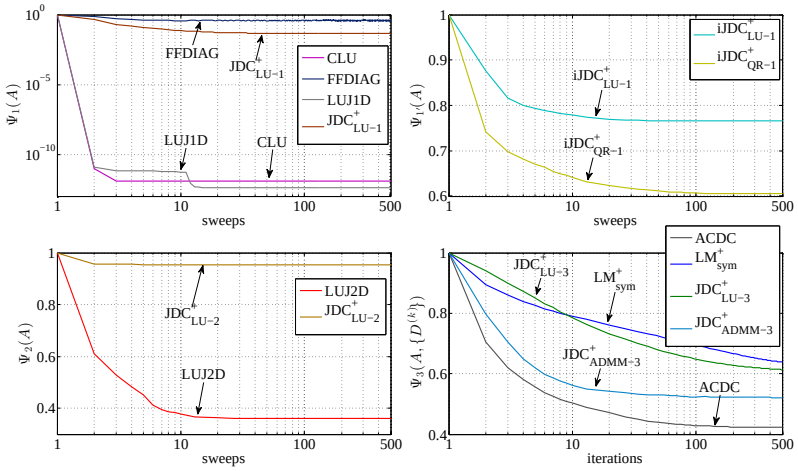
The convergence is assessed under three SNR conditions: SNR = 30 dB, 10 dB and 0 dB, respectively. Figure 3.1 shows the average convergence curves measured in terms of the cost functions as a function of sweeps/iterations. The twelve algorithms are separated into four groups according to their cost functions. In general, each algorithm exhibits a relative stable convergence pattern under different SNR levels. Among the classical approaches, CLU shows the fastest convergence property. Generally, it converges in less than 10 sweeps. FFDIAG also converges fast when SNR is equal to 10 dB and 0 dB. LUJ1D and LUJ2D require less than 20 sweeps to converge. The alternating optimization based method ACDC, as well as $\text{LM}^{+}_{\text{sym}}$, converge slower than those Jacobi-like algorithms. Generally ACDC converges after 200 iterations, while $\text{LM}^{+}_{\text{sym}}$ requires more than 500 iterations to converge. Regarding the proposed methods, $\text{JDC}^{+}_{\text{LU-1}}$ converges in 30 sweeps. $\text{JDC}^{+}_{\text{LU-2}}$ converges in 20 sweeps. $\text{iJDC}^{+}_{\text{LU-1}}$ and $\text{iJDC}^{+}_{\text{QR-1}}$ converge in less than 40 sweeps and 100 sweeps, respectively. Generally, $\text{JDC}^{+}_{\text{ADMM-3}}$ converges in 50 iterations. $\text{JDC}^{+}_{\text{LU-3}}$ converges the slowest among the proposed methods and requires more than 500 iterations to converge. In addition, it can be seen that there exist some fluctuations in the convergence curves of CLU, FFDIAG, $\text{JDC}^{+}_{\text{LU-2}}$ and $\text{JDC}^{+}_{\text{ADMM-3}}$, which are caused by the increase of the cost functions in a few Mont Carlo realizations. For CLU and FFDIAG, they minimize the approximations of the cost function (1.27). Therefore, in a few cases, the inaccurate estimation of the parameters leads to an increase of the original cost function. Similarly, the fluctuation of the convergence curve of $\text{JDC}^{+}_{\text{LU-2}}$ is caused by the step of unconstrained minimization (2.127), which is also an approximation of the original JDC criterion. Regarding $\text{JDC}^{+}_{\text{ADMM-3}}$, the limited fluctuation is due to the projection of $\boldsymbol{U}$ into the nonnegative space by equation (2.138), which can lead to an increase of the cost function in rare cases [Kim et al., 2007]. In most cases, these algorithms show good convergence properties. The numerical complexities of all the methods in this experiment are listed in table 3.1. It shows that the classical Jacobi-like methods, CLU, FFDIAG, LUJ1D and LUJ2D achieve the lowest numerical complexities. The proposed methods require more flops to converge, but they are still much less costly than $\text{LM}^{+}_{\text{sym}}$. Among the proposed methods, $\text{JDC}^{+}_{\text{ADMM-3}}$ is the cheapest method in terms of complexity, and it needs just a bit more flops than ACDC.

3.2.2 Test on non-square matrices

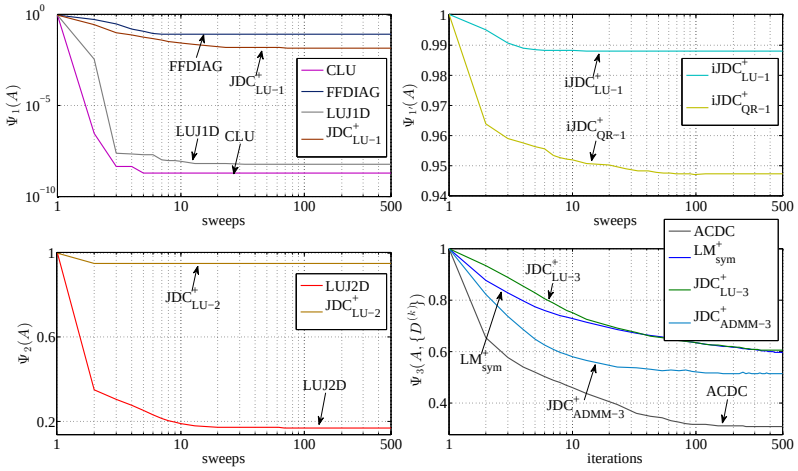
In this experiment, we compare four of the proposed algorithms, namely $\text{JDC}^{+}_{\text{LU-1}}$, $\text{JDC}^{+}_{\text{LU-2}}$, $\text{JDC}^{+}_{\text{LU-3}}$ and $\text{JDC}^{+}_{\text{ADMM-3}}$, with ACDC and $\text{LM}^{+}_{\text{sym}}$ for estimating a non-square matrix $\boldsymbol{A}$. These methods can be directly applied to solve a non-square JDC problem. The dimensions of the joint transformation matrix $\boldsymbol{A} \in \mathbb{R}_{+}^{N \times P}$ are set to $N = 5$ and $P = 3$, and the number of the input matrices $\boldsymbol{C}^{(k)}$ is fixed to $K = 5$. The convergence is also assessed under three SNR conditions: SNR = 30 dB, 10 dB and 0 dB, respectively. The convergence curves of those six algorithms measured in terms of the cost functions as a function of sweeps/iterations are shown in figure 3.2. It shows that $\text{JDC}^{+}_{\text{LU-1}}$ and $\text{JDC}^{+}_{\text{LU-2}}$ converge in less than 50 sweeps and 10 sweeps, respectively, whatever the SNR is. ACDC and $\text{JDC}^{+}_{\text{ADMM-3}}$ show similar convergence behaviors, and they require 50 iterations to converge. $\text{LM}^{+}_{\text{sym}}$ and $\text{JDC}^{+}_{\text{ADMM-3}}$ converge in 200 iterations when SNR is equal to 0 dB, and in 500 iterations when SNR is equal to 10 dB. Nevertheless, it seems that they do not converge in 500 iterations when SNR = 30 dB. The numerical complexities of those six methods are listed in table 3.2. We can observe that ACDC and $\text{JDC}^{+}_{\text{ADMM-3}}$



(a) Convergence as a function of sweeps/iterations with SNR = 30 dB



(b) Convergence as a function of sweeps/iterations with SNR = 10 dB



(c) Convergence as a function of sweeps/iterations with SNR = 0 dB

Figure 3.1: The convergence in terms of minimizing the cost functions for estimating a (3×3) square nonnegative joint transformation matrix $\boldsymbol{A}$ from 5 input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (3×3) with three different SNR values.

	Computational complexity (flops)		
	SNR = 30 dB	SNR = 10 dB	SNR = 0 dB
ACDC	1.0237×10^5	5.3056×10^4	5.0350×10^4
CLU	7.7080×10^3	1.5691×10^4	7.4604×10^3
FFDIAG	8.9549×10^3	1.2108×10^4	5.9718×10^3
LUJ1D	3.2279×10^3	9.1461×10^3	2.2162×10^4
LUJ2D	7.8555×10^3	1.9839×10^4	4.5952×10^4
$\text{LM}^{+}_{\text{sym}}$	2.1544×10^6	5.4142×10^6	3.7247×10^6
$\text{iJDC}^{+}_{\text{LU-1}}$	1.8195×10^5	5.7355×10^4	1.2168×10^4
$\text{iJDC}^{+}_{\text{QR-1}}$	1.7893×10^6	1.1665×10^6	1.2487×10^5
$\text{JDC}^{+}_{\text{LU-1}}$	4.5719×10^5	1.2280×10^6	7.6316×10^5
$\text{JDC}^{+}_{\text{LU-2}}$	1.9668×10^5	1.3533×10^5	6.8720×10^4
$\text{JDC}^{+}_{\text{LU-3}}$	1.6055×10^6	1.1272×10^6	7.9567×10^5
$\text{JDC}^{+}_{\text{ADMM-3}}$	1.0582×10^5	6.1208×10^4	6.9601×10^4

Table 3.1: Average numerical complexities of twelve methods in the convergence test for estimating a (3×3) square nonnegative joint transformation matrix $\boldsymbol{A}$ from 5 input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (3×3) with various SNR values.

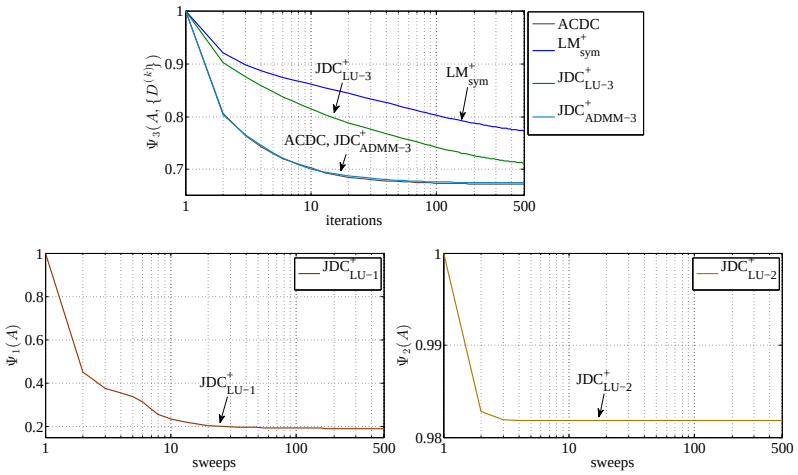
	Computational complexity (flops)		
	SNR = 30 dB	SNR = 10 dB	SNR = 0 dB
ACDC	1.2473×10^5	9.2324×10^4	5.5712×10^4
$\text{LM}^{+}_{\text{sym}}$	2.0315×10^6	9.2810×10^6	6.6496×10^6
$\text{JDC}^{+}_{\text{LU-1}}$	1.0388×10^6	4.0737×10^6	4.3547×10^6
$\text{JDC}^{+}_{\text{LU-2}}$	4.8495×10^5	6.8402×10^5	5.1805×10^5
$\text{JDC}^{+}_{\text{LU-3}}$	3.2191×10^6	4.1906×10^6	3.7078×10^6
$\text{JDC}^{+}_{\text{ADMM-3}}$	8.2344×10^4	7.8530×10^4	9.1460×10^4

Table 3.2: Average numerical complexities of six methods in the convergence test for estimating a (5×3) non-square nonnegative joint transformation matrix $\boldsymbol{A}$ from 5 input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (5×5) with various SNR values.

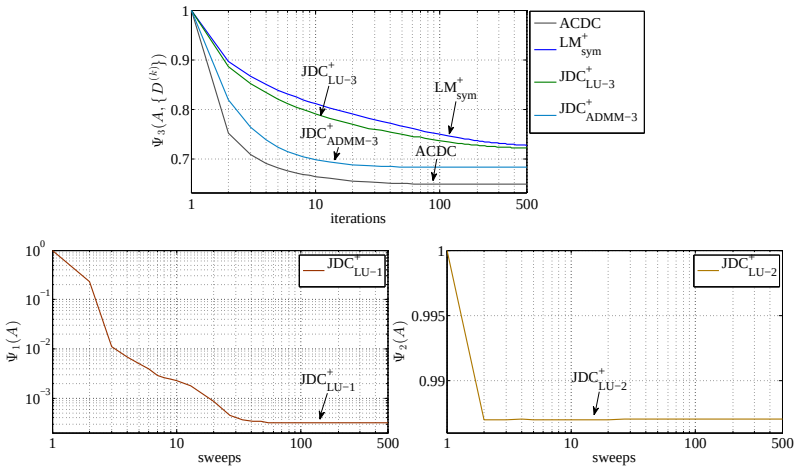
maintain the lowest numerical complexities. $\text{LM}^{+}_{\text{sym}}$ requires the heaviest computational load when SNR is equal to 10 dB and 0 dB. When SNR = 30 dB, $\text{JDC}^{+}_{\text{LU-3}}$ is the most computationally expensive method. The two Jacobi-like methods, $\text{JDC}^{+}_{\text{LU-1}}$ and $\text{JDC}^{+}_{\text{LU-2}}$ need relatively moderate complexities, while the latter is more efficient than the former for estimating a non-square matrix.

3.3 Influence of SNR

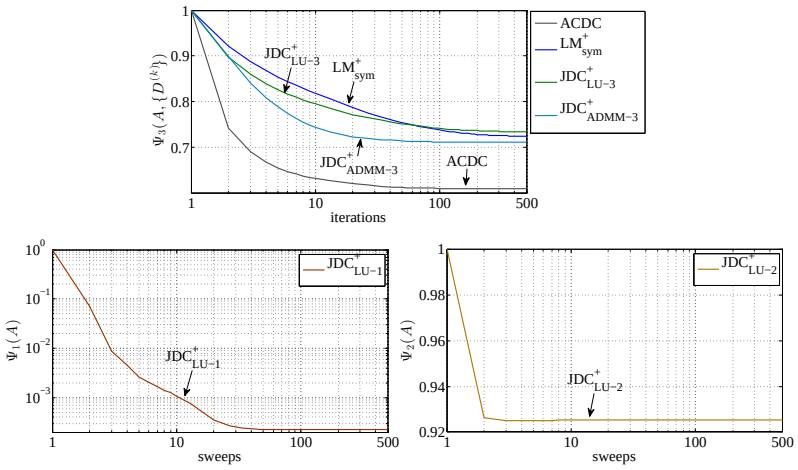
In this section, we study the behaviors of the considered algorithms as a function of SNR.



(a) Convergence as a function of sweeps/iterations with SNR = 30 dB



(b) Convergence as a function of sweeps/iterations with SNR = 10 dB



(c) Convergence as a function of sweeps/iterations with SNR = 0 dB

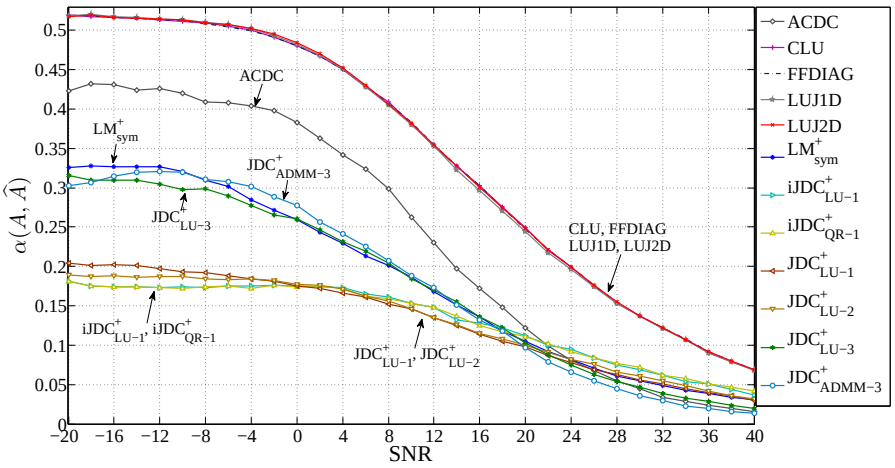
Figure 3.2: The convergence in terms of minimizing the cost functions for estimating a (5×3) non-square nonnegative joint transformation matrix $\boldsymbol{A}$ from 5 input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (5×5) with three different SNR values.

3.3.1 Test on square matrices

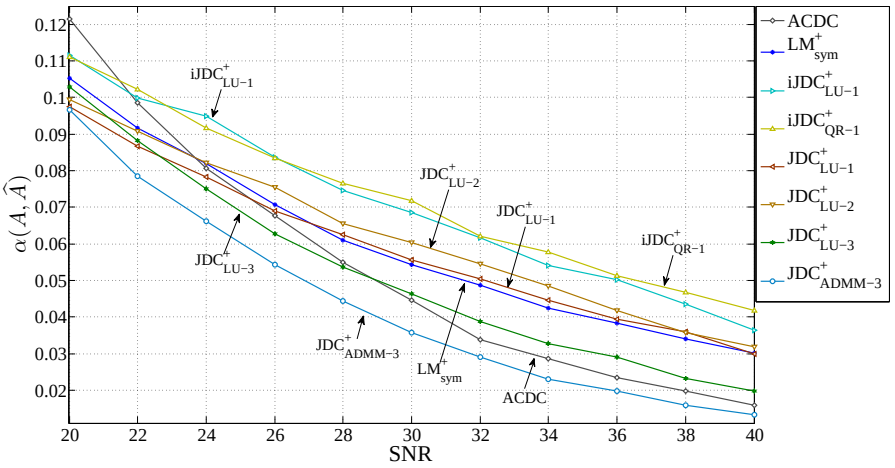
In this experiment, the dimensions of the square joint transformation matrix $\boldsymbol{A} \in \mathbb{R}_{+}^{N \times P}$ are fixed to $N = 3$ and $P = 3$. The number of the input matrices $\boldsymbol{C}^{(k)}$ is set to $K = 5$. We repeat the experiment with SNR ranging from -20 dB to 40 dB with a step of 2 dB. Figure 3.3(a) depicts the average curves of the estimation error $\alpha(\boldsymbol{A}, \widehat{\boldsymbol{A}})$ of the twelve algorithms as a function of SNR. The obtained results show that the performance of all the methods increases quasi-quadratically as SNR grows. For the classical unconstrained methods, ACDC performs better than CLU, FFDIAG, LUJ1D and LUJ2D, while the latter four Jacobi-like methods show almost the same performance. The nonnegativity constraint obviously helps $\text{LM}_{\text{sym}}^{+}$ and the proposed six methods to improve the estimation accuracy especially for lower SNR values. When SNR is less than 20 dB, the proposed four Jacobi-like algorithms, $\text{iJDC}_{\text{LU-1}}^{+}$, $\text{iJDC}_{\text{QR-1}}^{+}$, $\text{JDC}_{\text{LU-1}}^{+}$ and $\text{JDC}_{\text{LU-2}}^{+}$ outperform the others, while the two alternating minimization based NJDC methods $\text{JDC}_{\text{LU-3}}^{+}$ and $\text{JDC}_{\text{ADMM-3}}^{+}$, as well as $\text{LM}_{\text{sym}}^{+}$, also product better results than the classical unconstrained algorithms. Figure 3.3(b) displays the detail information for some selected methods with SNR ranging from 20 dB to 40 dB. We can seen that $\text{JDC}_{\text{ADMM-3}}^{+}$ maintains a competitive advantage over the other methods under those higher SNR conditions. $\text{JDC}_{\text{LU-3}}^{+}$ also achieves good results, and it outperforms ACDC when SNR ≤ 28 dB. $\text{LM}_{\text{sym}}^{+}$, $\text{JDC}_{\text{LU-1}}^{+}$ and $\text{JDC}_{\text{LU-2}}^{+}$ give similar performance, and they produce more accurate estimation results than $\text{iJDC}_{\text{LU-1}}^{+}$ and $\text{iJDC}_{\text{QR-1}}^{+}$. The average numerical complexities at each SNR level of all the methods in this experiment are shown in figure 3.3(c). It shows that the classical Jacobi-like methods CLU, FFDIAG and LUJ1D obtain the lowest numerical complexities. Generally, $\text{LM}_{\text{sym}}^{+}$ is the most computationally expensive algorithm. Among the proposed methods, the complexity of $\text{iJDC}_{\text{LU-1}}^{+}$ is the lowest, and increase as SNR grows. The complexity of $\text{JDC}_{\text{ADMM-3}}^{+}$ is almost identical to that of ACDC. $\text{iJDC}_{\text{QR-1}}^{+}$, $\text{JDC}_{\text{LU-1}}^{+}$, $\text{JDC}_{\text{LU-2}}^{+}$ and $\text{JDC}_{\text{LU-3}}^{+}$ cost more flops to achieve better estimations of $\boldsymbol{A}$, but they are still much more efficient than $\text{LM}_{\text{sym}}^{+}$. Therefore, the $\text{JDC}_{\text{LU-2}}^{+}$ and $\text{JDC}_{\text{ADMM-3}}^{+}$ give the best performance/complexity trade-off for all the considered SNR values.

3.3.2 Test on non-square matrices

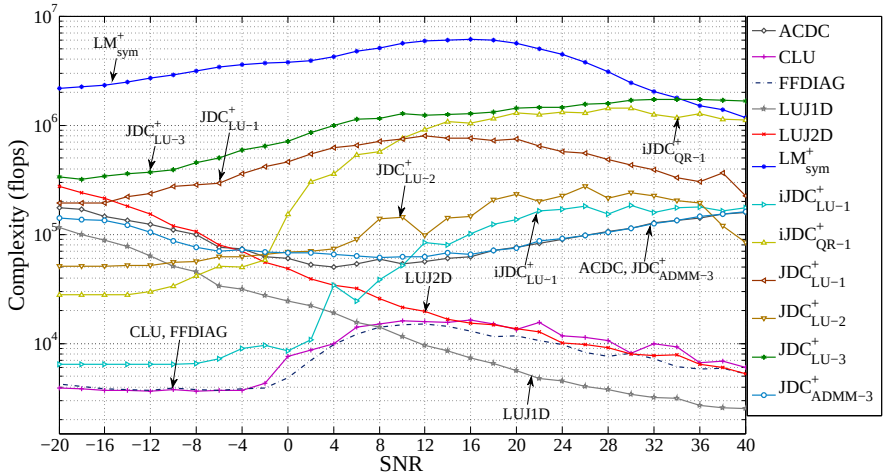
In this experiment, a non-square joint transformation matrix $\boldsymbol{A} \in \mathbb{R}_{+}^{N \times P}$ is considered, where the dimensions are set to $N = 5$ and $P = 3$, and where the number of the input matrices $\boldsymbol{C}^{(k)}$ is fixed to $K = 5$. We repeat the experiment with SNR ranging from -20 dB to 40 dB with a step of 2 dB. Figure 3.4(a) shows the average curves of the estimation error $\alpha(\boldsymbol{A}, \widehat{\boldsymbol{A}})$ as a function of SNR for ACDC, $\text{LM}_{\text{sym}}^{+}$, $\text{JDC}_{\text{LU-1}}^{+}$, $\text{JDC}_{\text{LU-2}}^{+}$, $\text{JDC}_{\text{LU-3}}^{+}$ and $\text{JDC}_{\text{ADMM-3}}^{+}$. As illustrated in the picture, the performance of all these methods increases as SNR grows. The nonnegativity constrained algorithms, $\text{LM}_{\text{sym}}^{+}$, $\text{JDC}_{\text{LU-1}}^{+}$, $\text{JDC}_{\text{LU-2}}^{+}$, $\text{JDC}_{\text{LU-3}}^{+}$ and $\text{JDC}_{\text{ADMM-3}}^{+}$, outperform ACDC when SNR is less than 14 dB, and perform similarly to ACDC when SNR exceeds 20 dB. The proposed $\text{JDC}_{\text{LU-2}}^{+}$ algorithm is more superior than the other methods when SNR is less than 12 dB. More detailed information of all the methods with SNR ranging from 12 dB to 40 dB is displayed in figure 3.4(b). Under these higher SNR levels, $\text{JDC}_{\text{ADMM-3}}^{+}$ gives the most accurate estimation results. $\text{JDC}_{\text{LU-3}}^{+}$ outperforms ACDC when SNR < 26 dB. $\text{JDC}_{\text{LU-1}}^{+}$ and $\text{JDC}_{\text{LU-2}}^{+}$ also achieve comparable results. The average numerical complexity over all the SNR values for each



(a) Estimation accuracy of $\boldsymbol{A}$ as a function of SNR



(b) Zoom in of figure (a) for a few methods with SNR ranging from 20 dB to 40 dB



(c) Numerical complexity as a function of SNR

Figure 3.3: Influence of SNR for estimating a (3×3) square nonnegative joint transformation matrix $\boldsymbol{A}$ from 5 input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (3×3).

algorithm is illustrated in figure 3.4(c). One can see that generally $\text{LM}^{+}_{\text{sym}}$ has the heaviest computation load. $\text{JDC}^{+}_{\text{LU-1}}$ and $\text{JDC}^{+}_{\text{LU-3}}$ are more efficient than $\text{LM}^{+}_{\text{sym}}$ under most SNR levels. ACDC requires the lowest complexity when SNR is less than 8 dB, while $\text{JDC}^{+}_{\text{ADMM-3}}$ remains the most computationally efficient when SNR exceeds 8 dB. The numerical complexity of $\text{JDC}^{+}_{\text{LU-2}}$ is between those of ACDC and $\text{JDC}^{+}_{\text{LU-1}}$. Thus, $\text{JDC}^{+}_{\text{LU-2}}$ and $\text{JDC}^{+}_{\text{ADMM-3}}$ provide the best accuracy/speed compromise.

3.4 Influence of number of input matrices

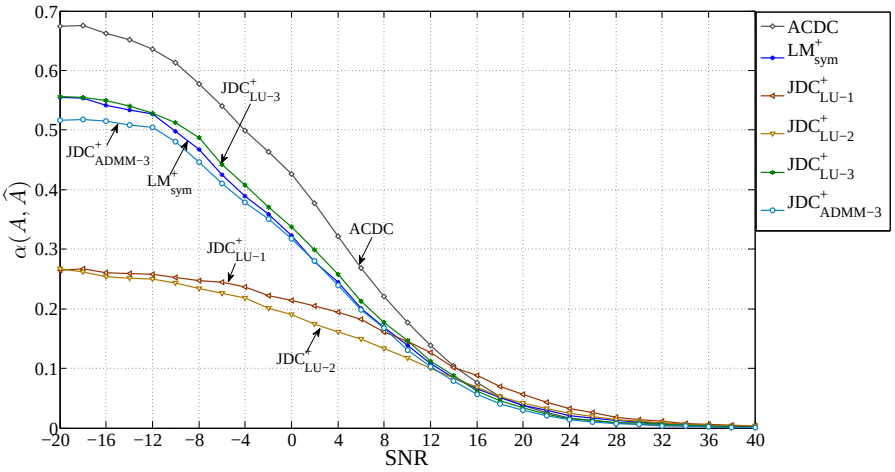
In this section, we study the influence of the number K of the input matrices $\boldsymbol{C}^{(k)}$ on the performance of the considered algorithms. In ICA, the value of K corresponds for instance to the number of matrix slices derived from a FO cumulant array.

3.4.1 Test on square matrices

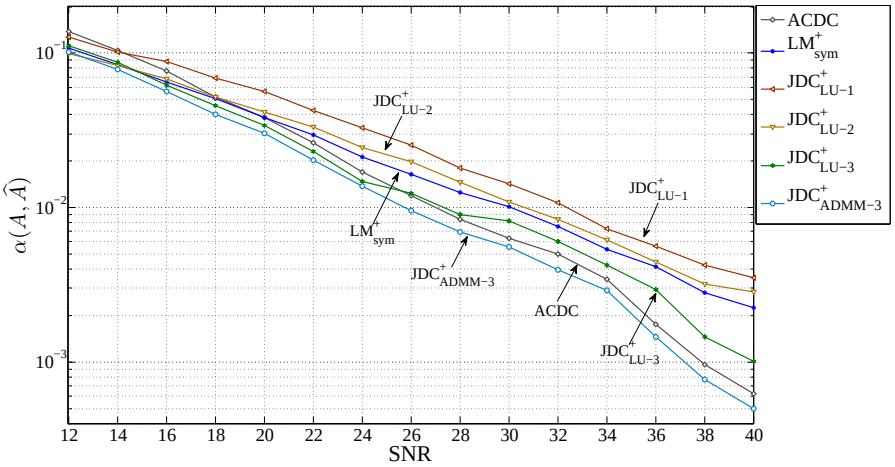
In this experiment, the dimensions of the square joint transformation matrix $\boldsymbol{A} \in \mathbb{R}_{+}^{N\times P}$ are set to $N = 3$ and $P = 3$. The SNR value is fixed to 10 dB. We repeat the experiment with K ranging from 3 to 63 with a step of 2. Figure 3.5(a) shows the average curves of the estimation error $\alpha(\boldsymbol{A}, \widehat{\boldsymbol{A}})$ of all the algorithms as a function of K. We can observe that the performance of all the methods progresses as K increases, and then practically stabilizes for high values of K. It indicates that after some point (e.g. $K \geq 30$), the additional information brought by an increase of K does not further improve the results. Regarding the four classical Jacobi-like methods, CLU, FFDIAG, LUJ1D and LUJ2D, their performance is quite similar. ACDC outperforms the aforementioned four algorithms. All the nonnegativity constrained methods give better results than the five classical unconstrained algorithms. The proposed $\text{JDC}^{+}_{\text{LU-1}}$ and $\text{JDC}^{+}_{\text{LU-2}}$ algorithms maintain competitive advantages over the other methods through all the K values. $\text{iJDC}^{+}_{\text{QR-1}}$ and $\text{JDC}^{+}_{\text{ADMM-3}}$ are also among the most competitive methods, and they achieve better results than $\text{LM}^{+}_{\text{sym}}$ under most K values. $\text{iJDC}^{+}_{\text{LU-1}}$ shows similar performance to $\text{LM}^{+}_{\text{sym}}$. The performance of $\text{JDC}^{+}_{\text{LU-3}}$ is between that of ACDC and $\text{LM}^{+}_{\text{sym}}$, and it is also comparable. The average numerical complexities of all the algorithms in this experiment are presented in figure 3.5(b). It shows that the numerical complexities of all the methods progress quasi-linearly as K increases except that of $\text{LM}^{+}_{\text{sym}}$, which grows quadratically in K. Now let's give a detailed description. The complexities of the classical Jacobi-like methods, CLU, FFDIAG, LUJ1D and LUJ2D, remain the lowest, especially FFDIAG. The proposed $\text{iJDC}^{+}_{\text{LU-1}}$, $\text{JDC}^{+}_{\text{LU-2}}$ and $\text{JDC}^{+}_{\text{ADMM-3}}$ algorithms cost a bit more flops than ACDC. The complexities of $\text{iJDC}^{+}_{\text{QR-1}}$, $\text{JDC}^{+}_{\text{LU-1}}$ and $\text{JDC}^{+}_{\text{LU-3}}$ are almost one order of magnitude higher than that of ACDC. Nevertheless, they are still much more efficient than $\text{LM}^{+}_{\text{sym}}$, especially for a large K value. Therefore, the $\text{JDC}^{+}_{\text{LU-2}}$ and $\text{JDC}^{+}_{\text{ADMM-3}}$ methods seem to be the most effective algorithms compared to the other methods in terms of both estimation accuracy and numerical complexity.

3.4.2 Test on non-square matrices

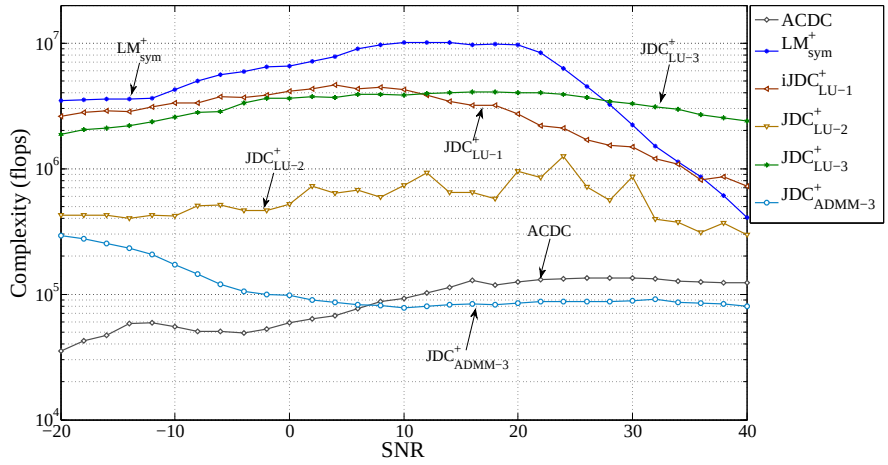
In this experiment, the dimensions of the non-square joint transformation matrix $\boldsymbol{A} \in \mathbb{R}_{+}^{N\times P}$ are set to $N = 5$ and $P = 3$, and the SNR value is fixed to 10 dB. Again, we repeat the experiment with K ranging from 3 to 63 with a step of 2. The average curves



(a) Estimation accuracy of $\boldsymbol{A}$ as a function of SNR

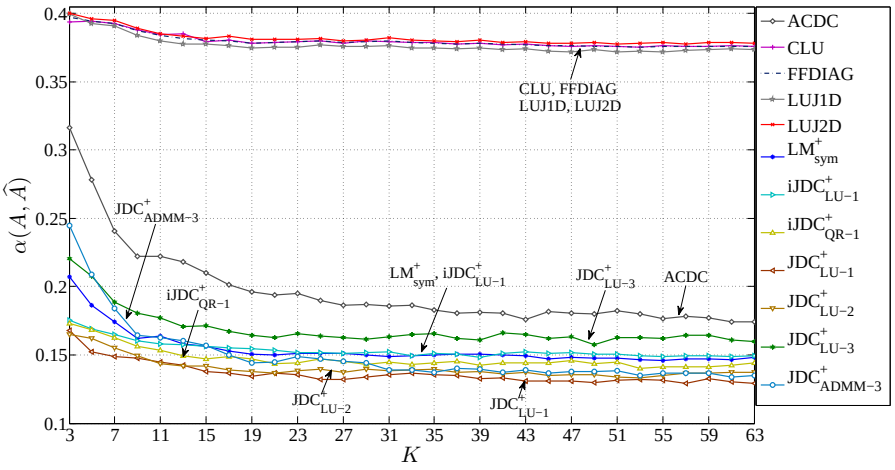


(b) Zoom in of figure (a) for a few methods with SNR ranging from 12 dB to 40 dB

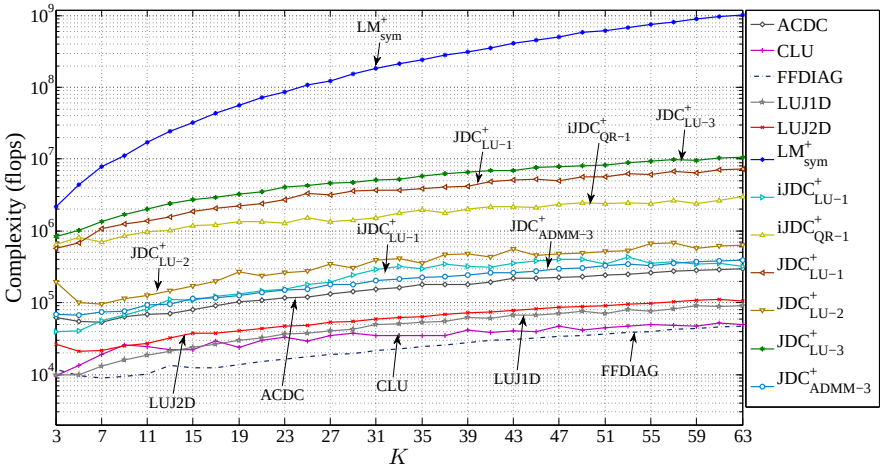


(c) Numerical complexity as a function of SNR

Figure 3.4: Influence of SNR for estimating a (5×3) non-square nonnegative joint transformation matrix $\boldsymbol{A}$ from 5 input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (5×5).



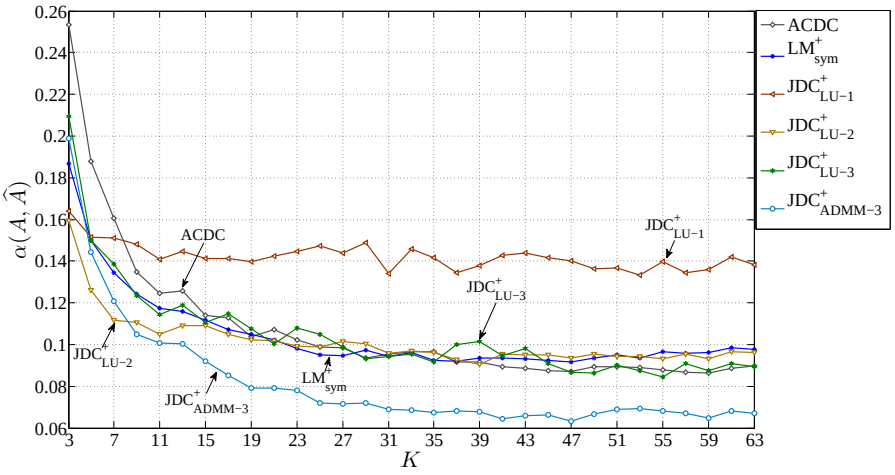
(a) Estimation accuracy of $\boldsymbol{A}$ as a function of number K of matrices



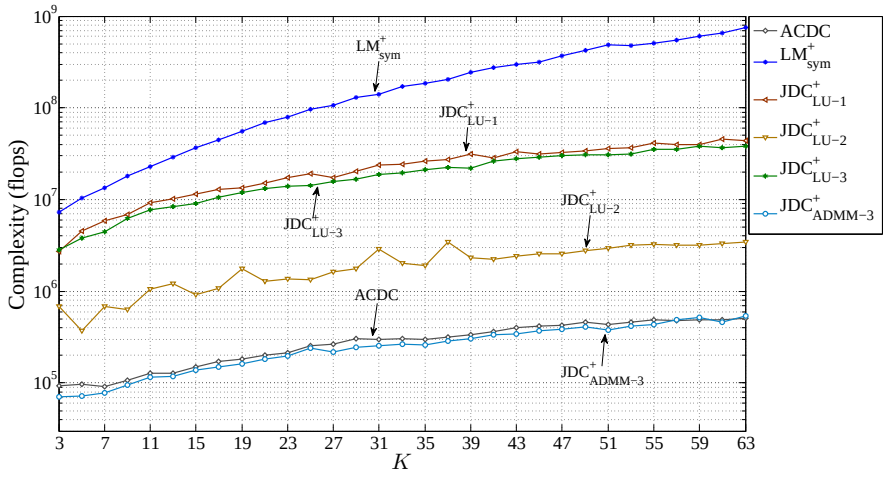
(b) Numerical complexity as a function of number K of matrices

Figure 3.5: Influence of number K of the input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (3×3) for estimating a (3×3) square nonnegative joint transformation matrix $\boldsymbol{A}$ with SNR $= 10$ dB.

of the estimation error $\alpha(\boldsymbol{A}, \widehat{\boldsymbol{A}})$ of the ACDC, $\text{LM}^{+}_{\text{sym}}$, $\text{JDC}^{+}_{\text{LU-1}}$, $\text{JDC}^{+}_{\text{LU-2}}$, $\text{JDC}^{+}_{\text{LU-3}}$ and $\text{JDC}^{+}_{\text{ADMM-3}}$ algorithms as a function of K are displayed in figure 3.6(a). The result shows that the increase of K benefits most JDC algorithms. $\text{JDC}^{+}_{\text{LU-2}}$ gives the most accurate estimation results when only a small number of input matrices is available (e.g. $K \leq 7$). When we have a large number of input matrices, that is to say K is larger than 11, $\text{JDC}^{+}_{\text{ADMM-3}}$ maintains competitive advantages over the other methods with large performance gains. $\text{LM}^{+}_{\text{sym}}$, $\text{JDC}^{+}_{\text{LU-2}}$ and $\text{JDC}^{+}_{\text{LU-3}}$ outperform ACDC when K is less than 15, and show similar performance to ACDC when $K > 15$. The result of $\text{JDC}^{+}_{\text{LU-1}}$ is less desirable for estimating a non-square matrix $\boldsymbol{A}$ when K is large. Figure 3.5(b) depicts the average numerical complexities of all the algorithms in this experiment. The complexities of all the methods in terms of flops grows as K increases. Obviously, $\text{LM}^{+}_{\text{sym}}$ requires the highest numerical complexity, while ACDC and $\text{JDC}^{+}_{\text{ADMM-3}}$ are the most computationally efficient methods. The complexity of $\text{JDC}^{+}_{\text{LU-2}}$ is nearly one order



(a) Estimation accuracy of $\boldsymbol{A}$ as a function of number K of matrices



(b) Numerical complexity as a function of number K of matrices

Figure 3.6: Influence of number K of the input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (5×5) for estimating a (5×3) square nonnegative joint transformation matrix $\boldsymbol{A}$ with SNR = 10 dB.

of magnitude higher than that of ACDC. The complexities of $\text{JDC}^{+}_{\text{LU-1}}$ and $\text{JDC}^{+}_{\text{LU-3}}$ are between those of $\text{JDC}^{+}_{\text{LU-2}}$ and $\text{LM}^{+}_{\text{sym}}$. Consequently, the $\text{JDC}^{+}_{\text{ADMM-3}}$ algorithm provides the best performance/complexity compromise, while the $\text{JDC}^{+}_{\text{LU-2}}$ and $\text{JDC}^{+}_{\text{LU-3}}$ algorithms are also competitive with regard to $\text{LM}^{+}_{\text{sym}}$.

3.5 Influence of coherence of matrix $\boldsymbol{A}$

In this section, the impact of the coherence between two column vectors of the joint transformation matrix $\boldsymbol{A}$ on the performance of the considered algorithms is evaluated. A nonnegative matrix $\boldsymbol{A}$ of dimension $(N \times P)$ is generated with a coherence parameter β as follows:

$$\boldsymbol{A} = [\boldsymbol{a}_1, (1-\beta)\boldsymbol{a}_1 + \beta\boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_P] \tag{3.4}$$

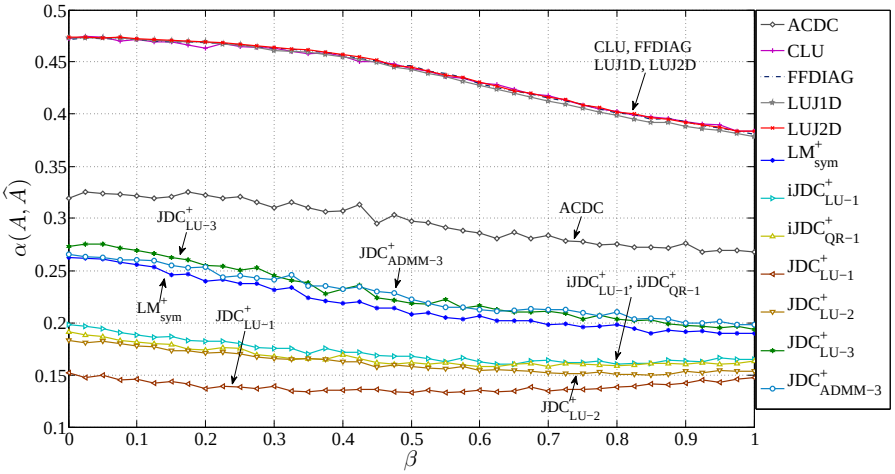
where each column vector $\boldsymbol{a}_p \in \mathbb{R}_+^N$ is randomly drawn from a uniform distribution over $[0, 1]$, and where β is in the range of $[0, 1]$. A small value of β indicates that the matrix $\boldsymbol{A}$ is close to be singular.

3.5.1 Test on square matrices

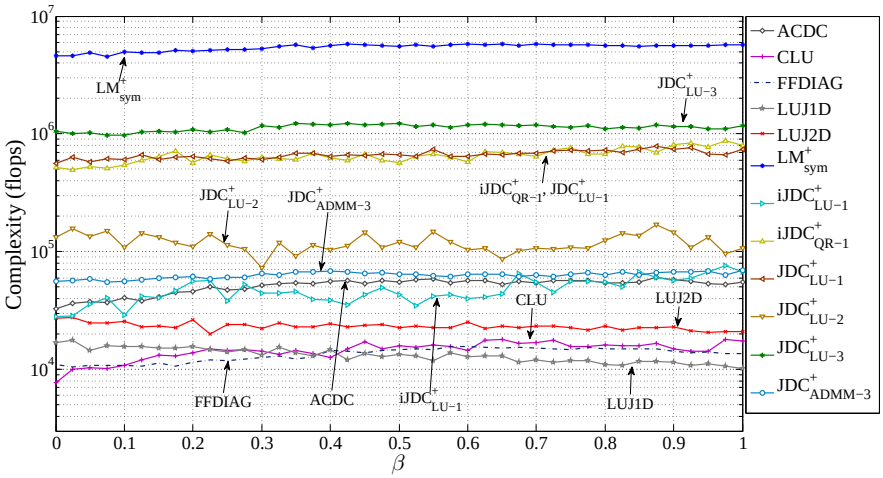
In this simulation, the dimensions of the square joint transformation matrix $\boldsymbol{A} \in \mathbb{R}_+^{N\times P}$ are set to $N = 3$ and $P = 3$, the number of the input matrices $\boldsymbol{C}^{(k)}$ is fixed to $K = 5$, and the SNR value is set to 10 dB. We repeat the experiment with β ranging from 0 to 1 with a step of 0.025. The average curves of the estimation error $\alpha(\boldsymbol{A}, \widehat{\boldsymbol{A}})$ of the twelve algorithms are depicted in figure 3.7(a). One can notice that in general all the JDC algorithms are less effective in terms of the estimation accuracy when two column vectors of $\boldsymbol{A}$ are close to collinear. The performance of most JDC methods increases quasi-linearly as β increases. The four classical Jacobi-like methods CLU, FFDIAG, LUJ1D and LUJ2D yield less accurate estimation results. ACDC exhibits an improved performance over the aforementioned methods. The nonnegativity constrained algorithms give much better results than the classical unconstrained methods. Three methods based on the direct-fit JDC criterion (1.29), namely $\text{LM}_{\text{sym}}^{+}$, $\text{JDC}_{\text{LU-3}}^{+}$ and $\text{JDC}_{\text{ADMM-3}}^{+}$, show similar performance. The three proposed Jacobi-like NJDC algorithms, namely $\text{iJDC}_{\text{LU-1}}^{+}$, $\text{iJDC}_{\text{QR-1}}^{+}$ and $\text{JDC}_{\text{LU-2}}^{+}$, outperform the previous three methods. The performance of $\text{JDC}_{\text{LU-1}}^{+}$ is quite stable and remains the best through all the considered β values. Figure 3.7(b) depicts the average numerical complexities of all the algorithms in this experiment. We can observe that the complexity evolution is quite stable as β changes. $\text{LM}_{\text{sym}}^{+}$ is the most computationally expensive, followed by $\text{JDC}_{\text{LU-3}}^{+}$, $\text{JDC}_{\text{LU-1}}^{+}$ and $\text{iJDC}_{\text{QR-1}}^{+}$, consecutively. $\text{iJDC}_{\text{LU-1}}^{+}$ and $\text{JDC}_{\text{ADMM-3}}^{+}$ cost a bit more or less flops than ACDC, while $\text{JDC}_{\text{LU-2}}^{+}$ is somewhat more expensive than ACDC. The classical Jacobi-like methods maintain the lowest numerical complexities. Hence, $\text{JDC}_{\text{LU-2}}^{+}$ gives the best accuracy/complexity compromise in this experiment, and $\text{JDC}_{\text{LU-1}}^{+}$ is also more efficient than $\text{LM}_{\text{sym}}^{+}$.

3.5.2 Test on non-square matrices

In this experiment, the dimensions of the non-square joint transformation matrix $\boldsymbol{A} \in \mathbb{R}_+^{N\times P}$ are set to $N = 5$ and $P = 3$, and the number of the input matrices $\boldsymbol{C}^{(k)}$ is fixed to $K = 5$. The SNR value is set to 10 dB. We repeat the experiment with β ranging from 0 to 1 with a step of 0.025. The average curves of the estimation error $\alpha(\boldsymbol{A}, \widehat{\boldsymbol{A}})$ of the six considered algorithms as a function of SNR are presented in figure 3.8(a). The obtained results show that the nonnegativity constrained methods outperform the classical ACDC algorithm. The direct-fit JDC criterion (1.29) based methods, $\text{LM}_{\text{sym}}^{+}$, $\text{JDC}_{\text{LU-3}}^{+}$ and $\text{JDC}_{\text{ADMM-3}}^{+}$ give similar results, where the proposed $\text{JDC}_{\text{ADMM-3}}^{+}$ algorithm is slightly better than the other two algorithms. $\text{JDC}_{\text{LU-1}}^{+}$ gives quite stable performance as β increase, and remains advantage when β is less than 0.4. $\text{JDC}_{\text{LU-2}}^{+}$ generates the most accurate results when β exceeds 0.5. The average numerical complexities evolutions of all the six methods as a function of β are illustrated in figure 3.8(b). We can see that ACDC and $\text{JDC}_{\text{ADMM-3}}^{+}$ are the most computationally efficient, followed by $\text{JDC}_{\text{LU-2}}^{+}$, $\text{JDC}_{\text{LU-1}}^{+}$ and $\text{JDC}_{\text{LU-3}}^{+}$, consecutively. $\text{LM}_{\text{sym}}^{+}$ requires the heaviest workload. Therefore, the proposed $\text{JDC}_{\text{LU-2}}^{+}$ and $\text{JDC}_{\text{ADMM-3}}^{+}$ algorithms offer the best performance/complexity compromise in this experiment.



(a) Estimation accuracy of $\boldsymbol{A}$ as a function of the coherence parameter β

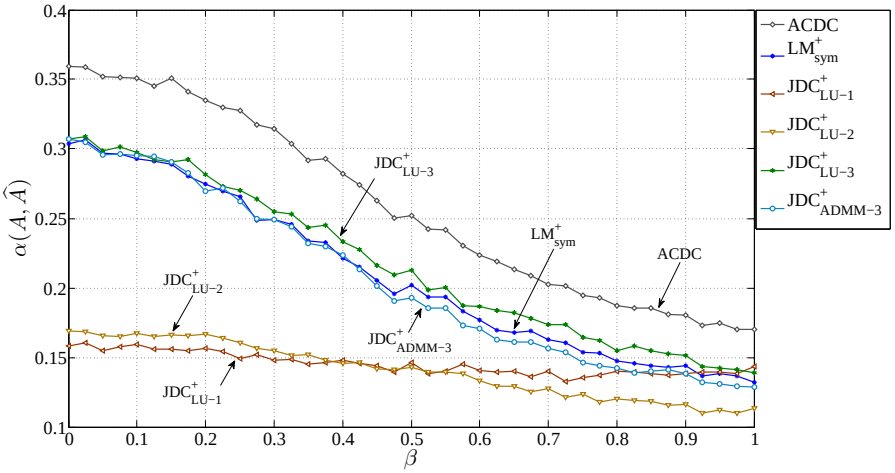


(b) Numerical complexity as a function of the coherence parameter β

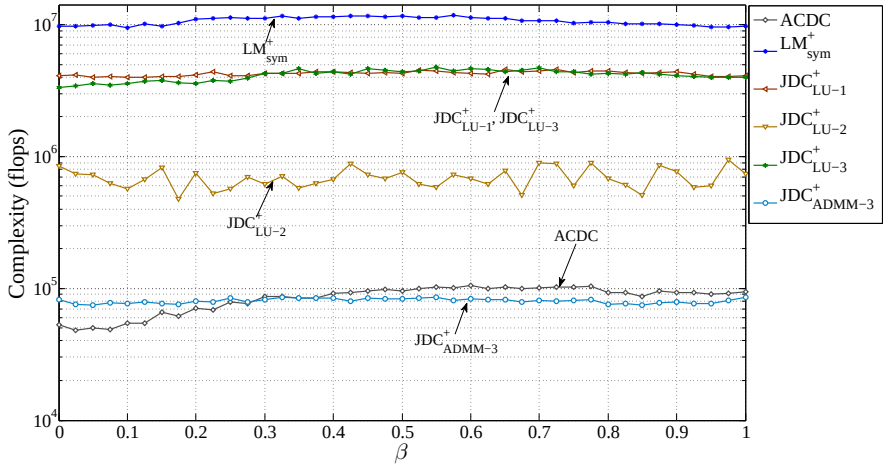
Figure 3.7: Influence of coherence of the column vectors of $\boldsymbol{A}$ for estimating a (3×3) square nonnegative joint transformation matrix $\boldsymbol{A}$ from 5 input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (3×3) with SNR = 10 dB.

3.6 Influence of modulus of uniqueness

In this section, the effect of the modulus of uniqueness ρ of a JDC problem is evaluated. The modulus of uniqueness ρ is defined in equation (1.26) in section 1.3.2. By its definition, ρ falls in the range of $[0, 1]$. The JDC problem is considered to be ill-conditioned when ρ is close to 1. In order to control ρ, firstly we randomly generate an orthogonal matrix $\boldsymbol{D} \in \mathbb{R}^{K \times P}$ so that $\rho = 0$ by orthogonalizing a $(K \times P)$ random matrix whose components are drawn from the zero-mean unit-variance Gaussian distribution. Secondly we rotate the P column vectors of $\boldsymbol{D}$ such that all the internal angles between any column vectors are equal to a predefined value ψ. Therefore ρ is only controlled by the angle ψ and is equal to $|\cos(\psi)|$. Finally we can obtain K diagonal matrix $\boldsymbol{D}^{(k)} \in \mathbb{R}^{P \times P}$, where the diagonal elements of each $\boldsymbol{D}^{(k)}$ are equal to the elements of the k-th row vector of the predefined matrix $\boldsymbol{D}$.



(a) Estimation accuracy of $\boldsymbol{A}$ as a function of the coherence parameter β

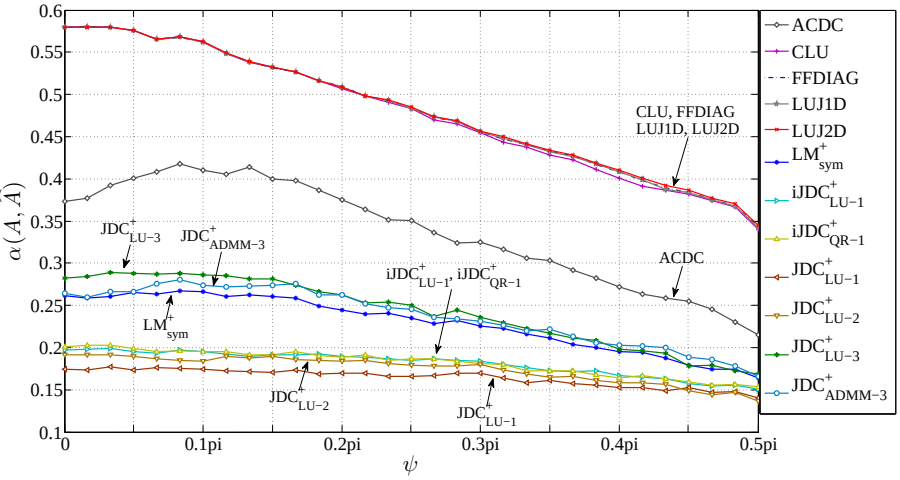


(b) Numerical complexity as a function of the coherence parameter β

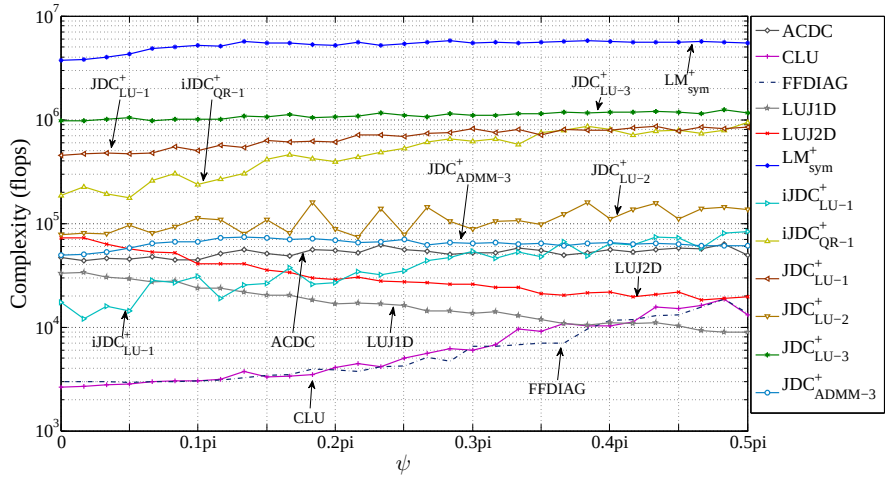
Figure 3.8: Influence of coherence of the column vectors of $\boldsymbol{A}$ for estimating a (5×3) non-square nonnegative joint transformation matrix $\boldsymbol{A}$ from 5 input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (5×5) with SNR $= 10$ dB.

3.6.1 Test on square matrices

In this simulation, the dimensions of the square joint transformation matrix $\boldsymbol{A} \in \mathbb{R}_+^{N \times P}$ are set to $N = 3$ and $P = 3$, the number of the input matrices $\boldsymbol{C}^{(k)}$ is fixed to $K = 5$, and the SNR value is set to 10 dB. We repeat the experiment with the angle ψ ranging from 0 to $\pi/2$ with a step of $\pi/60$. A small ψ value means a large ρ value. Figure 3.9(a) shows the average curves of the estimation error $\alpha(\boldsymbol{A}, \widehat{\boldsymbol{A}})$ of the twelve algorithms as a function of ψ. As we can see, in general cases, the performance of all the algorithms increases as the value of ψ grows. Four classical Jacobi-like methods CLU, FFDIAG, LUJ1D and LUJ2D give moderate results, while ACDC offers an improvement in terms of estimation accuracy. The nonnegativity constraint obviously benefits $\text{LM}_{\text{sym}}^{+}$ and the proposed six methods to further enhance the results. Generally speaking, the performance of the proposed four Jacobi-like NJDC methods, namely $\text{JDC}_{\text{LU-1}}^{+}$, $\text{JDC}_{\text{LU-2}}^{+}$, $\text{iJDC}_{\text{LU-1}}^{+}$ and $\text{iJDC}_{\text{QR-1}}^{+}$, surpasses that of $\text{LM}_{\text{sym}}^{+}$, $\text{JDC}_{\text{LU-3}}^{+}$ and $\text{JDC}_{\text{ADMM-3}}^{+}$, where the latter three



(a) Estimation accuracy of $\boldsymbol{A}$ as a function of the modulus of uniqueness indicator ψ



(b) Numerical complexity as a function of the modulus of uniqueness indicator ψ

Figure 3.9: Influence of modulus of uniqueness for estimating a (3×3) square nonnegative joint transformation matrix $\boldsymbol{A}$ from 5 input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (3×3) with SNR = 10 dB.

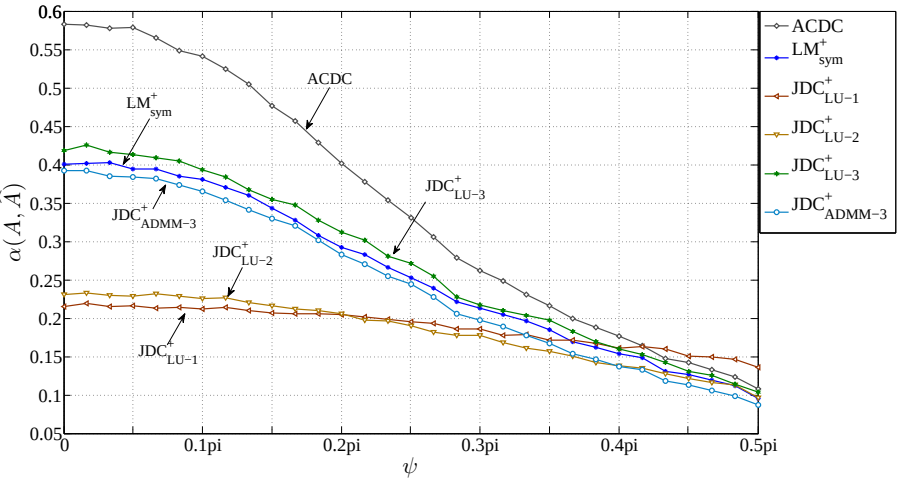
methods are based on the same direct-fit cost function (1.29). The $\text{JDC}^{+}_{\text{LU-1}}$ algorithm maintains competitive advantage through all the considered ψ values. The average numerical complexities of all the algorithms as a function of ψ in this experiment is shown in figure 3.9(b). As illustrated in the figure, generally the classical Jacobi-like methods CLU and FFDIAG preserve the lowest numerical complexities. LUJ1D and LUJ2D are still efficient. The complexity of $\text{iJDC}^{+}_{\text{LU-1}}$ is the lowest among those of the proposed six methods, and it is also lower than or equal to that of ACDC. $\text{JDC}^{+}_{\text{ADMM-3}}$ costs a bit more computation load than ACDC, followed by $\text{JDC}^{+}_{\text{LU-2}}$ and $\text{iJDC}^{+}_{\text{QR-1}}$, consecutively. The complexities of $\text{JDC}^{+}_{\text{LU-1}}$ and $\text{JDC}^{+}_{\text{LU-3}}$ are almost one order of magnitude higher than that of $\text{JDC}^{+}_{\text{ADMM-3}}$. The $\text{LM}^{+}_{\text{sym}}$ algorithm requires the heaviest workload. Consequently, the proposed $\text{iJDC}^{+}_{\text{LU-1}}$ and $\text{JDC}^{+}_{\text{LU-2}}$ algorithms offer the best performance/complexity trade-off for all the considered ψ values.

3.6.2 Test on non-square matrices

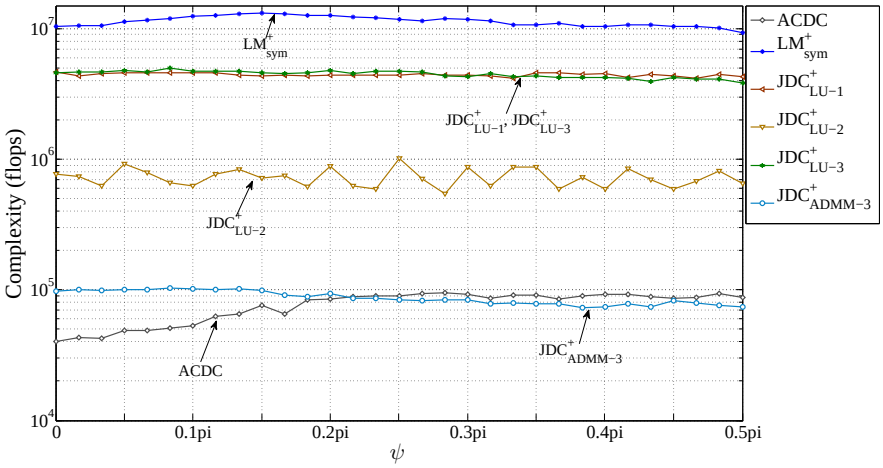
In this experiment, the dimensions of the non-square joint transformation matrix $\boldsymbol{A} \in \mathbb{R}_+^{N\times P}$ are set to $N = 5$ and $P = 3$, and the number of the input matrices $\boldsymbol{C}^{(k)}$ is fixed to $K = 5$. The SNR value is fixed to 10 dB. We repeat the experiment with the angle ψ ranging from 0 to $\pi/2$ with a step of $\pi/60$. Figure 3.10(a) displays the average curves of the estimation error $\alpha(\boldsymbol{A}, \widehat{\boldsymbol{A}})$ of the six selected algorithms as a function of ψ. One can observe that the nonnegativity constrained algorithms surpass the classical ACDC algorithm, except $\text{JDC}^+_{\text{LU-1}}$ when $\psi > 5\pi/12$. Three methods based on the direct-fit JDC criterion (1.29), called LM^+_{sym}, $\text{JDC}^+_{\text{LU-3}}$ and $\text{JDC}^+_{\text{ADMM-3}}$ behave similarly, where the proposed $\text{JDC}^+_{\text{ADMM-3}}$ algorithm offers better performance and gives the best estimation result when $\psi > 2\pi/5$. The proposed two Jacobi-like algorithms $\text{JDC}^+_{\text{LU-1}}$ and $\text{JDC}^+_{\text{LU-2}}$, particularly $\text{JDC}^+_{\text{LU-1}}$, achieve large performance gains when ψ is less than $3\pi/10$. The evolutions of the average numerical complexities of the six methods as a function of ψ are depicted in figure 3.10(b). It shows that ACDC and $\text{JDC}^+_{\text{ADMM-3}}$ keep the lowest computational complexities, while ACDC is the most efficient when $\psi < 2\pi/10$. The complexity of $\text{JDC}^+_{\text{LU-2}}$ is almost one order of magnitude higher than that of $\text{JDC}^+_{\text{ADMM-3}}$. $\text{JDC}^+_{\text{LU-1}}$ and $\text{JDC}^+_{\text{LU-3}}$ need the heaviest workload among the proposed methods. Nevertheless, they are still more efficient than LM^+_{sym}, since the complexities of the former are half of the latter. Therefore, the proposed $\text{JDC}^+_{\text{LU-2}}$ algorithm gives the best performance/complexity compromise in this experiment.

3.7 Concluding remark

In this chapter, the performance of the proposed six NJDC algorithms was evaluated with simulated set of jointly diagonalizable matrices. Five classical nonorthogonal JDC methods without using the nonnegativity constraint, including ACDC [Yeredor, 2002], CLU [Maurandi et al., 2013], FFDIAG [Ziehe et al., 2004], LUJ1D [Afsari, 2006], LUJ2D [Afsari, 2006], and one NJDC method LM^+_{sym} [Coloigner et al., 2014c], were tested as reference methods. The performance was assessed in terms of the matrix estimation accuracy and the numerical complexity. Then, various scenarios were considered, aiming at testing: i) the convergence property, ii) the influence of additive noise, which can stand for model errors, iii) the impact of the number of input matrices, iv) the effect of coherence of the column vectors of the joint transformation matrix, and v) the influence of the modulus of uniqueness. In each scenario, the performance was extensively studied through Monte Carlo experiments for both estimating a square joint transformation matrix and a non-square one. The obtained results showed that the proposed algorithms offer better estimation accuracy in difficult contexts by means of exploiting the nonnegativity *a priori*. Comparing to an existing nonnegative method based on the Levenberg Marquardt optimization [Coloigner et al., 2014c], the proposed methods reduce the total numerical complexity by up to three orders of magnitude. For a difficult problem with either a lower value of SNR, a small number of available input matrices, a high level of coherence presented in the joint transformation matrix or a large value of modulus of uniqueness, generally the proposed $\text{JDC}^+_{\text{LU-2}}$ algorithm provides the best performance/-complexity compromise. $\text{JDC}^+_{\text{LU-1}}$ also gives the best estimation results in many difficult contexts, where a highly coherent joint transformation matrix is encountered or a large



(a) Estimation accuracy of $\boldsymbol{A}$ as a function of the modulus of uniqueness indicator ψ



(b) Numerical complexity as a function of the modulus of uniqueness indicator ψ

Figure 3.10: Influence of modulus of uniqueness for estimating a (5×3) non-square nonnegative joint transformation matrix $\boldsymbol{A}$ from 5 input symmetric matrices $\boldsymbol{C}^{(k)}$ of size (5×5) with SNR $= 10$ dB.

value of modulus of uniqueness is involved, however, at the cost of a higher computational complexity. When a well-conditioned NJDC problem is considered, that is to say, when a high value of SNR can be expected or a large number of input matrices is available, the $\text{JDC}^{+}_{\text{ADMM-3}}$ algorithm is the most effective in terms of estimation accuracy and also has a low numerical complexity.

Chapter 4

Blind source separation applications

We have shown the good performance of the proposed algorithms for solving simulated NJDC model in the last chapter. The purpose of this chapter is to illustrate the practical usefulness of our methods by solving three BSS problems from very different application domains. The first one which is rooted in the biomedical diagnostics, consists of separating synthetic realistic Magnetic Resonance Spectroscopy (MRS) signal. The second one is an automatic music transcription problem. It aims at estimating the musical notes and their attributes from a single-channel music record by decomposing its magnitude spectrogram. The third one is devoted to separating digital images which are degraded by the so-called show-through effect. Such a problem is usually caused by the seeping of ink from the reverse side or scanning a semi-transparent paper. In the following paragraphs, the SeNICA methods based on the proposed NJDC algorithms, namely, for example, $\text{JDC}^{+}_{\text{ADMM-3}}$-ICA, consist of the following four steps:

1. Compressing the M realizations of the observation vector $\boldsymbol{x} \in \mathbb{R}^N$ by means of a nonnegative matrix $\overline{\boldsymbol{W}} \in \mathbb{R}_{+}^{P\times N}$ computed using the NonNegative COMPression (NN-COMP) method, which is addressed in appendix B, leading to a compressed observation vector $\bar{\boldsymbol{x}} \in \mathbb{R}^P$, where N, P and M are the numbers of the observations, of the sources and of the samples points, respectively. The numerical complexity of NN-COMP is analyzed in equation (B.20) of appendix B.

2. Estimating the FO cumulant array $\boldsymbol{\mathcal{C}}_{\boldsymbol{x}}^{(4)} \in \mathbb{R}^{N\times N\times N\times N}$ from the original observation $\boldsymbol{x}$, or estimating the FO cumulant array $\boldsymbol{\mathcal{C}}_{\bar{\boldsymbol{x}}}^{(4)} \in \mathbb{R}^{P\times P\times P\times P}$ from the compressed observation $\bar{\boldsymbol{x}}$; then extracting all the cumulant matrix slices to form a matrices set. The complexity of computing $\boldsymbol{\mathcal{C}}_{\boldsymbol{x}}^{(4)}$ is given as follows [Albera et al., 2010]:
$$f_{\text{cum}}^{(4)}(N, M) = \frac{3}{8}N(N+1)(N^2+N+2)M \tag{4.1}$$
while the complexity of calculating $\boldsymbol{\mathcal{C}}_{\bar{\boldsymbol{x}}}^{(4)}$ is $f_{\text{cum}}^{(4)}(P, M)$.

3. Estimating the mixing matrix $\boldsymbol{A} \in \mathbb{R}_{+}^{N\times P}$ or the compressed mixing matrix $\overline{\boldsymbol{A}}_{+} = \overline{\boldsymbol{W}}\boldsymbol{A} \in \mathbb{R}_{+}^{P\times P}$ by applying the proposed NJDC methods on the FO cumulant matrix slices of $\boldsymbol{\mathcal{C}}_{\boldsymbol{x}}^{(4)}$ or $\boldsymbol{\mathcal{C}}_{\bar{\boldsymbol{x}}}^{(4)}$, respectively. The numerical complexities of the proposed NJDC methods are listed in table 2.1.

4. Reconstructing the M realizations of the source vector by means of $\boldsymbol{s} = \boldsymbol{A}^{\sharp}\boldsymbol{x}$ or $\boldsymbol{s} = \overline{\boldsymbol{A}}_{+}^{-1}\bar{\boldsymbol{x}}$, where $\sharp$ denotes the pseudo-inverse operator. The complexity of this step is $2NP^2 + NPM + P^3$ or $P^2M + P^3$ by using the above two equations, respectively.

The first step is indispensable for the iJDC$^{+}_{\text{LU-1}}$-ICA and iJDC$^{+}_{\text{QR-1}}$-ICA algorithms when a non-square mixing matrix is encountered, since they only work on a square matrix. Such a compression step is optional for the JDC$^{+}_{\text{LU-1}}$-ICA, JDC$^{+}_{\text{LU-2}}$-ICA, JDC$^{+}_{\text{LU-3}}$-ICA and JDC$^{+}_{\text{ADMM-3}}$-ICA methods. The numerical complexities of the proposed six SeNICA methods include the complexities of the four steps. Their full expressions are straightforward, thus omitted here for brevity. The behaviors of the proposed SeNICA methods are compared with five state-of-the-art BSS algorithms, namely two efficient ICA methods CoM2 [Comon, 1994] and SOBI [Belouchrani et al., 1997], the NICA method with a line search along the geodesic [Plumbley, 2003], the NMF method based on alternating nonnegativity least squares [Kim and Park, 2008], and the semi-NMF method based on one-sided multiplicative update [Ding et al., 2010]. A comprehensive analysis of the numerical complexities of the classical BSS algorithms is provided in [Kachenoura et al., 2008, Albera et al., 2010, Albera et al., 2012]. Their results are not repeated here for the sake of conciseness.

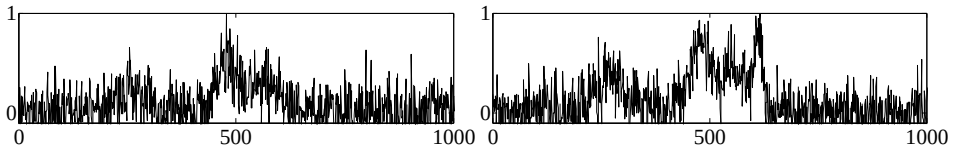
4.1 Separation of simulated MRS signal

MRS is a powerful non-invasive analytical technique for analyzing the chemical content of Magnetic Resonance (MR)-visible nuclei and therefore enjoys particular advantages for assessing metabolism. The chemical property of each nucleus determines the frequency at which it appears in the MR spectrum, giving rise to peaks corresponding to specific metabolites [Befroy and Shulman, 2011]. Therefore, the MRS observation spectra can be modeled as the mixture of the spectrum of each constitutional source metabolite. More specifically, it can be written as the noisy linear instantaneous mixing model described in equation (1.50) as in the SeNICA model 1.6, where $\boldsymbol{x} \in \mathbb{R}^N$ is the MRS observation vector, $\boldsymbol{s} \in \mathbb{R}^P$ is the source vector representing the statistically quasi-independent source metabolites, $\boldsymbol{\varepsilon} \in \mathbb{R}^N$ is the instrumental noise vector, and $\boldsymbol{A} \in \mathbb{R}_{+}^{N\times P}$ is the nonnegative mixing matrix containing the concentrations of the source metabolites. In this context, SNR is defined as follows:

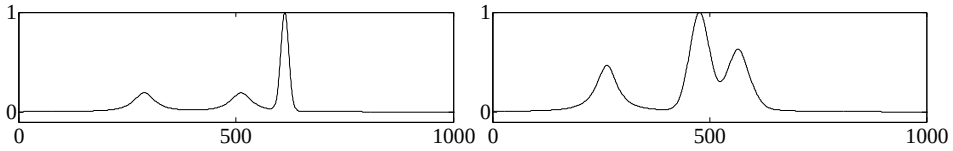
$$\text{SNR} = 20\log_{10}\frac{\|\boldsymbol{A}\boldsymbol{s}\|_F}{\|\boldsymbol{\varepsilon}\|_F} \tag{4.2}$$

Many non-parametric BSS methods were developed in order to solve the MRS separation problem, such as the ICA approach [Nakai et al., 2004, Szabo de Edelenyi et al., 2005] and the NMF approach [Sajda et al., 2004, Sun and Xin, 2012].

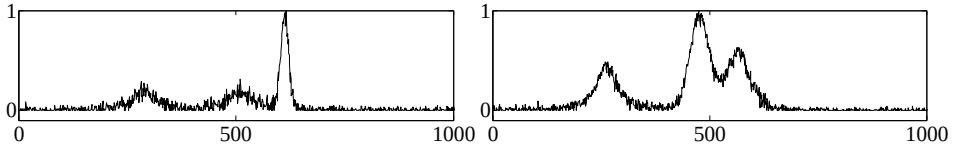
In this experiment, two simulated MRS source metabolites s_1 and s_2, namely the Choline and Myo-inositol (see figure 4.1(b)), are generated by the sum of Lorentzian and Gaussian functions, where the location and scale parameters are fixed for the purposed of realistic representation [Moussaoui, 2005]. Each of the sources contains 10^3 samples. The observation vector $\boldsymbol{x}$ is generated according to equation (1.50) (see figure 4.1(a) for illustration of two observations with a SNR of 10 dB). The components of the $(N \times 2)$ mixing matrix $\boldsymbol{A}$ are randomly drawn from a uniform distribution. The additive noise $\boldsymbol{\varepsilon}$ is modeled as a zero-mean unit-variance Gaussian vector. The performance is assessed



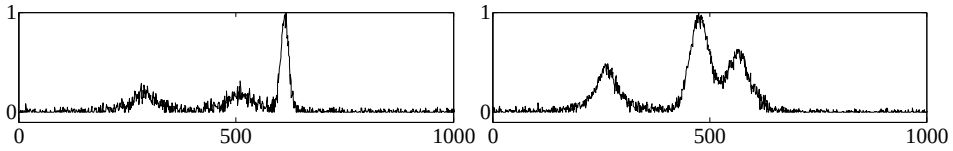
(a) Two of the 20 observations



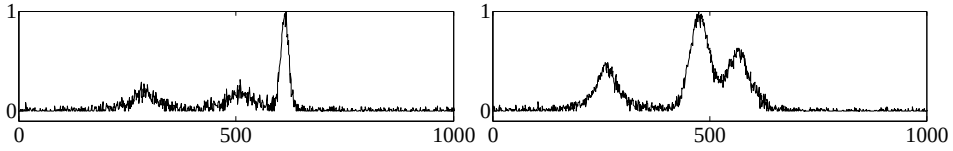
(b) Choline and Myo-inositol source spectra



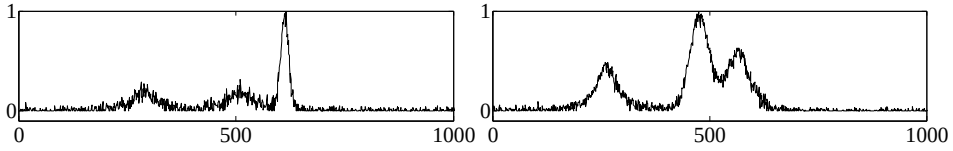
(c) Separated spectra by $\text{JDC}^{+}_{\text{LU-1}}$-ICA



(d) Separated spectra by $\text{JDC}^{+}_{\text{LU-2}}$-ICA



(e) Separated spectra by $\text{JDC}^{+}_{\text{LU-3}}$-ICA

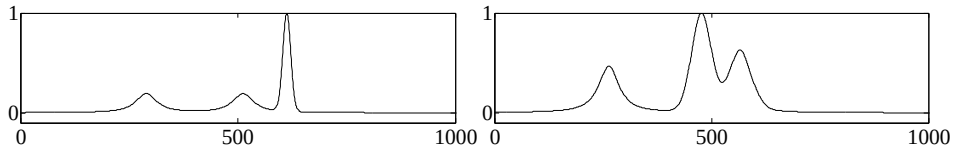


(f) Separated spectra by $\text{JDC}^{+}_{\text{ADMM-3}}$-ICA

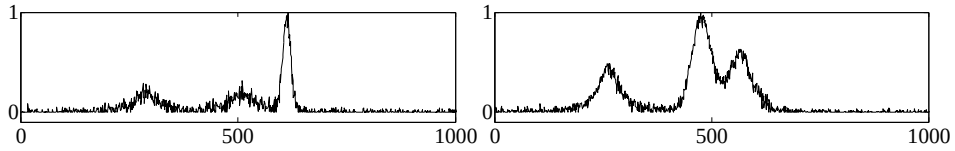
Figure 4.1: An example of the BSS results of 2 simulated MRS metabolites by the proposed four methods without using the nonnegative compression step. The number of observations is set to $N = 20$ and the SNR value is fixed to 10 dB.

by means of the estimation error $\alpha(\boldsymbol{s}^{\mathsf{T}}, \widehat{\boldsymbol{s}}^{\mathsf{T}})$ between the true source $\boldsymbol{s}$ and its estimate $\widehat{\boldsymbol{s}}$, and the numerical complexity, where the distance measure α is defined in equation (3.2). Furthermore, all the results reported in the section are averaged over 200 independent Monte Carlo trials.

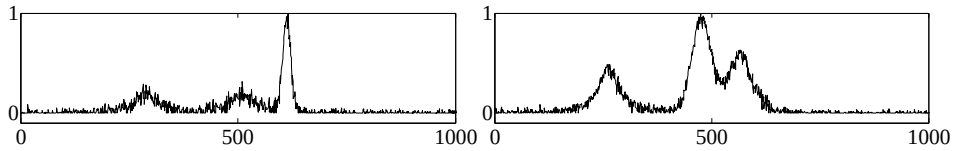
Figure 4.1 shows an example of the separation results of four of the proposed methods without using the NN-COMP method, namely $\text{JDC}^{+}_{\text{LU-1}}$-ICA, $\text{JDC}^{+}_{\text{LU-2}}$-ICA, $\text{JDC}^{+}_{\text{LU-3}}$-ICA and $\text{JDC}^{+}_{\text{ADMM-3}}$-ICA, with $N = 20$ observations and a SNR of 10 dB. Figure 4.2 is the continuation of figure 4.1. It gives the separation results of all the proposed



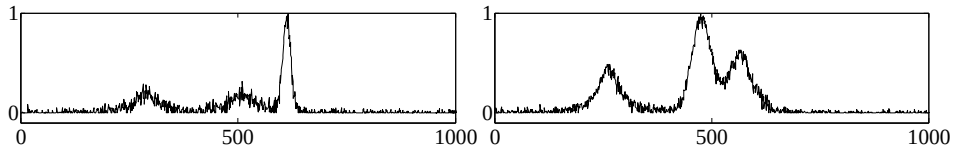
(a) Choline and Myo-inositol source spectra



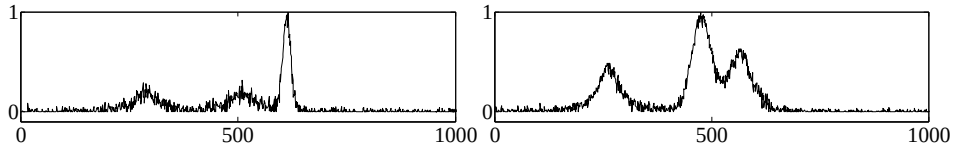
(b) Separated spectra by NN-COMP + $\text{JDC}^{+}_{\text{LU-1}}$-ICA



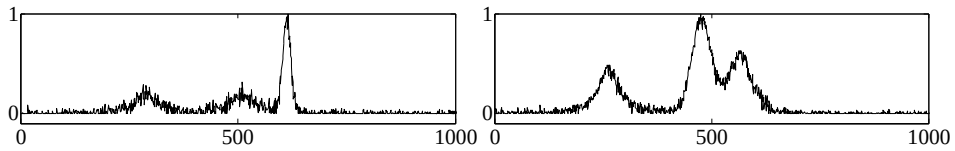
(c) Separated spectra by NN-COMP + $\text{JDC}^{+}_{\text{LU-2}}$-ICA



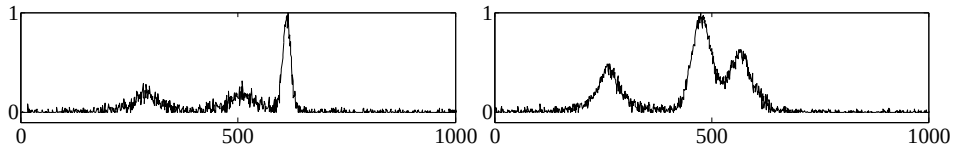
(d) Separated spectra by NN-COMP + $\text{JDC}^{+}_{\text{LU-3}}$-ICA



(e) Separated spectra by NN-COMP + $\text{JDC}^{+}_{\text{ADMM-3}}$-IC



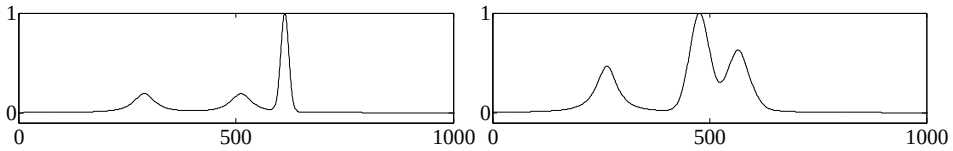
(f) Separated spectra by NN-COMP + $\text{iJDC}^{+}_{\text{LU-1}}$-ICA



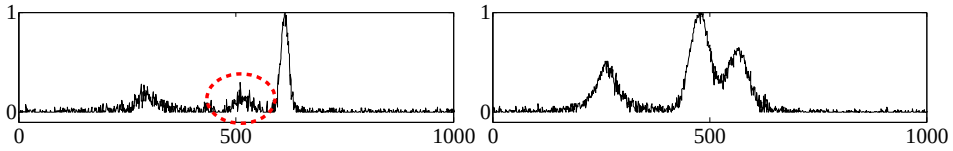
(g) Separated spectra by NN-COMP + $\text{iJDC}^{+}_{\text{QR-1}}$-ICA

Figure 4.2: Continuation of the first example of the BSS results of 2 simulated MRS metabolites by the proposed six methods with the nonnegative compression step. The number of observations is set to $N = 20$ and the SNR value is fixed to 10 dB.

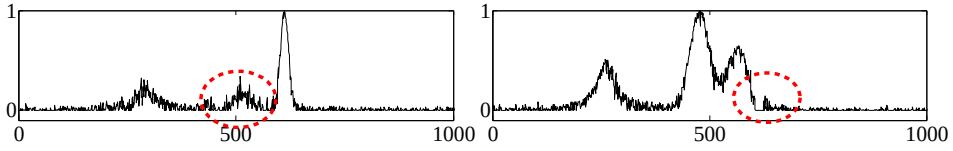
six SeNICA methods based on the usage of the NN-COMP algorithm. In addition, the separation results obtained by five classical BSS methods, namely CoM2 [Comon,



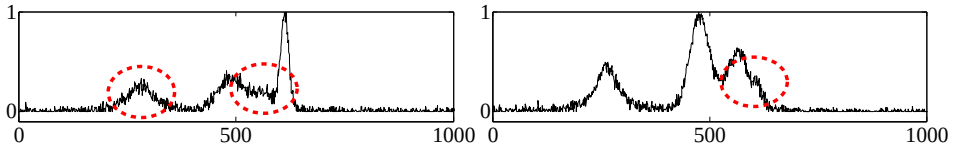
(a) Choline and Myo-inositol source spectra



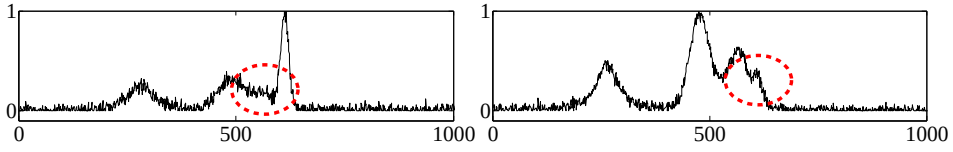
(b) Separated spectra by CoM2



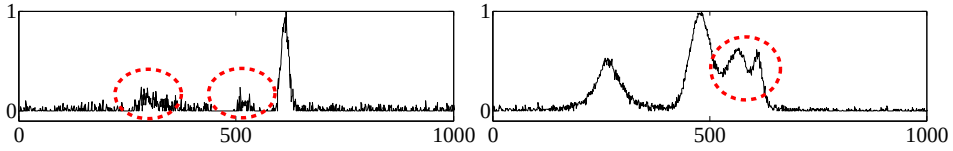
(c) Separated spectra by NICA



(d) Separated spectra by NMF



(e) Separated spectra by Semi-NMF



(f) Separated spectra by SOBI

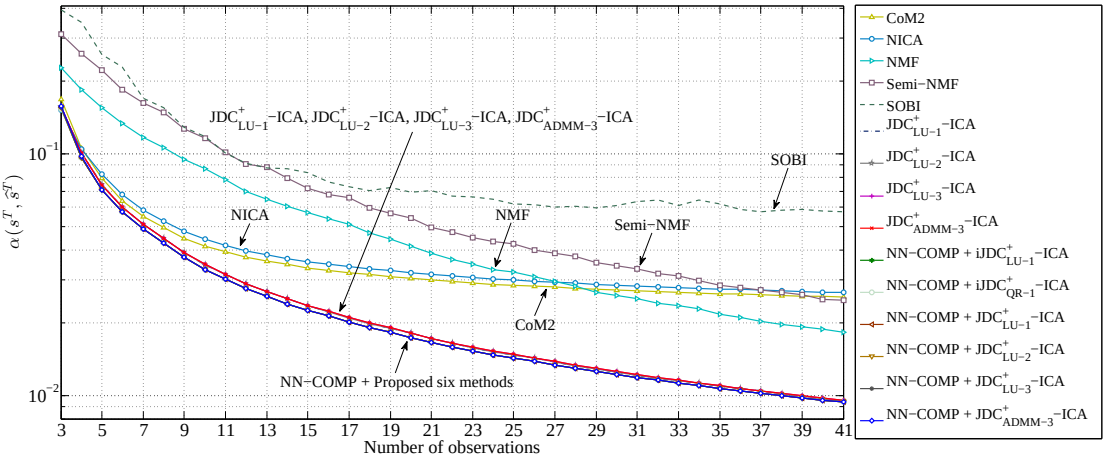
Figure 4.3: Continuation of the first example of the BSS results of 2 simulated MRS metabolites by five classical BSS methods. The number of observations is set to $N = 20$ and the SNR value is fixed to 10 dB. The red dot circles indicate the significant estimation errors in figures (b) to (f).

1994], NICA [Plumbley, 2003], NMF [Kim and Park, 2008], semi-NMF [Ding et al., 2010] and SOBI [Belouchrani et al., 1997] are shown in figure 4.3. We can observe that the proposed six SeNICA algorithms, incorporating or not the NN-COMP method, provide 10 solutions to the MRS separation problem. In these 10 solutions, the separated spectra obtained by the proposed different methods are visually indistinguishable, whatever the NN-COMP is applied or not. Furthermore, we can say that all the proposed methods yield quasi-perfect solutions. Regarding the classical BSS methods, CoM2 and NICA

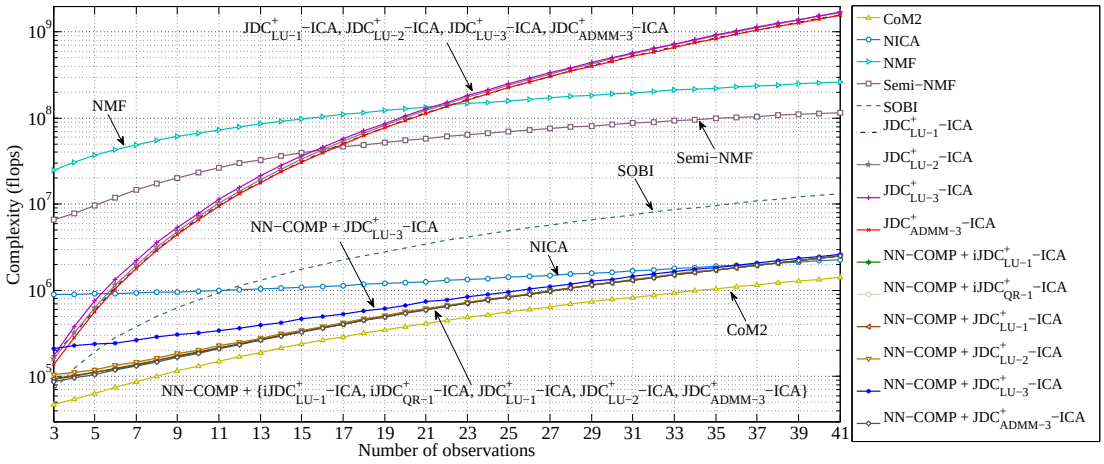
give good results, but their separated spectra are still disturbed. Regarding NMF and semi-NMF, the Choline peak still slightly exists in the estimated Myo-inositol spectrum, and the estimated Choline spectrum is also disturbed. SOBI cannot separate both source spectra. These experimental results show that by fully exploiting the prior information of data, such as the nonnegativity of the mixing matrix and the statistical independence of the sources, the proposed methods achieve visually better estimation results. In the following sections, the performances of all the considered methods are quantitatively analyzed by means of two experiments. More precisely, we study the influences of the number of observations and of the SNR.

4.1.1 Influence of the number of observations

In this experiment, the effect of the number of observations N is evaluated. The SNR is fixed to 10 dB. The fifteen methods are compared with N ranging from 3 to 41. The average curves of the estimation error $\alpha(\boldsymbol{s}^{\mathsf{T}}, \widehat{\boldsymbol{s}}^{\mathsf{T}})$ of all the methods as a function of N are shown in figure 4.4(a). It can be seen that the estimation errors of all the methods improves as N increases. It suggests that in noisy BSS contexts, using more sensors often yields better results [Joho et al., 2000]. Obviously, the proposed ten methods maintain the competitive advantages when N is greater than 5, where the six methods using the NN-COMP preprocessing slightly outperform the ones without a compression step. We can infer that NN-COMP not only compresses the dimension but also reduces the effect of additive noise to a certain extent. Moreover, in a relatively low SNR context, the merit of the NN-COMP step overweights its disadvantage, for instance, introducing additional numerical errors as in the classical prewhitening step. Regarding the classical BSS methods, CoM2 and NICA behave quite similarly and provide good results when N is less than 25. However, they restrict the search space to the group of orthogonal matrices, which may limit their BSS performance in this problem. As we can see, their performance stabilizes when N exceeds 20. NMF outperforms CoM2 and NICA when $N \geq 30$. Semi-NMF gives less accurate estimation results than NMF. It seems that only the nonnegativity constraint cannot guarantee a good separation result. SOBI yields the worst results. It may be because the source processes in this experiment do not fully satisfy the non-iid and stationary assumption of SOBI. The average curves of the numerical complexities of all the considered methods in this experiment are shown in figure 4.4(b). We can notice that the numerical complexities of all the methods increase as a function of N. CoM2 is the most computationally efficient method. The proposed six methods incorporating NN-COMP cost a bit more flops than CoM2, where NN-COMP + $\text{JDC}^{+}_{\text{LU-3}}$-ICA is somewhat more expensive than the other five algorithms. NICA and SOBI are still economical in terms of flops. The aforementioned nine methods keep relatively low numerical complexities since all of them use a dimension reduction step by means of either classical prewhitening or NN-COMP. The methods without using a dimension reduction step, namely NMF, semi-NMF, $\text{JDC}^{+}_{\text{LU-1}}$-ICA, $\text{JDC}^{+}_{\text{LU-2}}$-ICA, $\text{JDC}^{+}_{\text{LU-3}}$-ICA and $\text{JDC}^{+}_{\text{ADMM-3}}$-ICA, are more computationally intensive. NMF requires the heaviest workload when N is less than 20. The complexity of semi-NMF is almost half of that of NMF. The proposed four methods $\text{JDC}^{+}_{\text{LU-i}}$-ICA, $i \in \{1, 2, 3\}$ and $\text{JDC}^{+}_{\text{ADMM-3}}$-ICA preserve moderate complexities when a small value of N is considered, say $N < 15$. However, their complexities show a higher growth rate as N increases, and surpass that of NMF when N is above 22. The large amount of computation load of these



(a) Estimation accuracy of the source vector $\boldsymbol{s}$ as a function of the number of observations



(b) Numerical complexity as a function of the number of observations

Figure 4.4: Influence of the number of observations for estimating 2 MRS source metabolites with SNR = 10 dB.

four methods partially comes from the estimation of the FO cumulant arrays. Among these four algorithms, JDC$^{+}_{\text{LU-3}}$-ICA seems the most costly, while JDC$^{+}_{\text{ADMM-3}}$-ICA is less computationally expensive. Therefore, the proposed SeNICA methods incorporating the NN-COMP offer the best performance/complexity compromise in this experimental context.

4.1.2 Influence of SNR

In this simulation, we study the influence of SNR on the performance of all the considered fifteen methods. The number of observations N is set to 20. SNR is varied from 0 dB to 60 dB with a step of 2 dB. The average curves of the estimation error $\alpha(\boldsymbol{s}^{\mathsf{T}}, \widehat{\boldsymbol{s}}^{\mathsf{T}})$ of all methods as function of SNR are presented in figure 4.5(a). It show that the performance of all methods increases as SNR grows, and then practically stabilizes for high values of SNR. Obviously, the proposed SeNICA algorithms offer the most accurate separation solutions, especially when SNR is above 5 dB. The performance of the proposed ten methods is

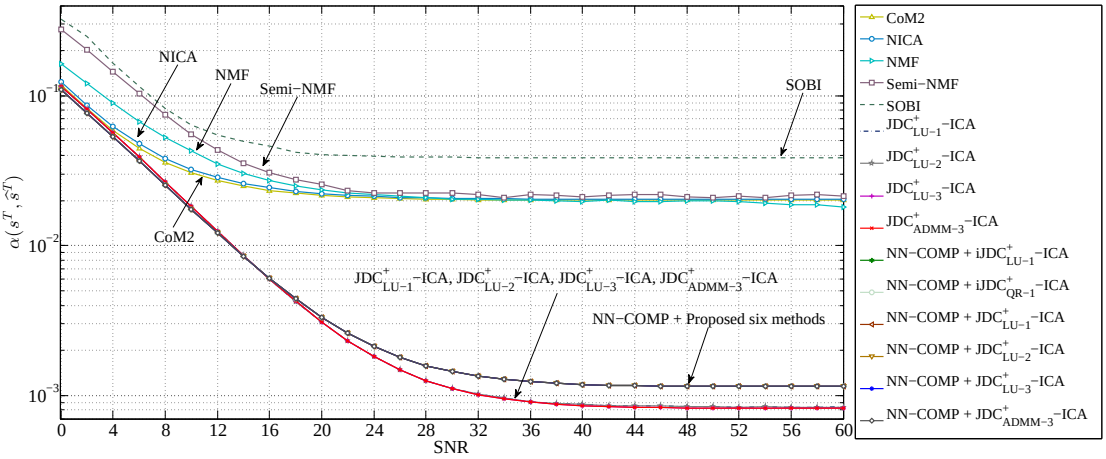
almost identical when the value of SNR is between 0 dB and 18 dB, where the six methods using NN-COMP are slightly more preferable when SNR $<$ 12 dB. As far as higher values of SNR are considered (e.g. SNR $>$ 20 dB), the proposed four methods, namely JDC$^{+}_{\text{LU-i}}$-ICA, $i \in \{1, 2, 3\}$ and JDC$^{+}_{\text{ADMM-3}}$-ICA, perform better than the six methods incorporating NN-COMP. This phenomenon suggests that in a high SNR context, the additional numerical errors introduced by NN-COMP may bring some limitations for the ensuing SeNICA algorithms. Regarding the classical BSS methods, when SNR is lower than 20 dB, CoM2 and NICA offer good results, followed by NMF, semi-NMF and SOBI, consecutively. The performance of CoM2, NICA, NMF and semi-NMF is almost the same when SNR is above 24 dB. SOBI achieves the worst solutions. Figure 4.5(b) illustrates the average curves of the numerical complexities of all methods in this experiment. We can observed that the complexity evolution is quite stable with the change in SNR. CoM2 maintains the lowest complexity, followed by the proposed six methods combined with NN-COMP, where NN-COMP + JDC$^{+}_{\text{LU-3}}$-ICA is more expensive. NICA and SOBI are also efficient. NMF is the most computationally intensive. Semi-NMF costs nearly half workload than NMF. The complexities of the proposed four methods without using NN-COMP are between those of semi-NMF and NMF. Consequently, the proposed SeNICA methods incorporating NN-COMP provide the best performance/complexity tradeoff in this experiment. Without using NN-COMP, the proposed four methods JDC$^{+}_{\text{LU-i}}$-ICA, $i \in \{1, 2, 3\}$ and JDC$^{+}_{\text{ADMM-3}}$-ICA give the most accurate estimation results at a cost of higher but acceptable numerical complexities, when a high SNR value is provided.

The results of the above two experiments demonstrate the improvement of the proposed methods in terms of the source estimation accuracy, and also show that exploiting the two *a priori* of the data, namely the nonnegativity of the mixing matrix and the statistical independence of the sources, allows us to achieve better estimation results.

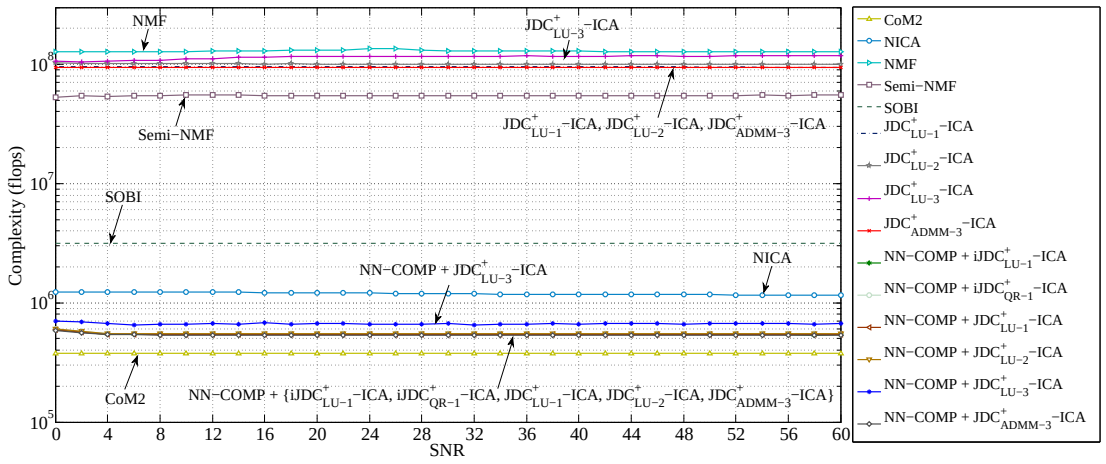
4.2 Automatic music transcription

Estimating the musical notes and their attributes (pitch, duration and onset time) from a music record is an objective of automatic music transcription. To this end, many unsupervised algorithms propose to decompose the music magnitude spectrogram $\boldsymbol{X}$ as $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S}$. Such a linear decomposition consists in modeling a time-frequency representation of music signal as a sum of basic atoms $\boldsymbol{S}$. Each row vector of matrix $\boldsymbol{S}$ represents a basic atom, which corresponds to the harmonically fixed spectrum of a single note [Smaragdis and Brown, 2003, Fuentes et al., 2013]. An atom will be active whenever the corresponding note is played. Furthermore, each column vector of matrix $\boldsymbol{A}$ is modeled as the nonnegative temporal weights of a basic note spectrum, that describes the onset time and duration of an active note. Many algorithms based on ICA [Brown et al., 2002, Plumbley and Abdallah, 2003, Brown and Smaragdis, 2004], NICA [Plumbley, 2003, Dittmar and Uhle, 2004] and NMF [Smaragdis and Brown, 2003, Holzapfel and Stylianou, 2008, Févotte et al., 2009, Gao et al., 2011, Smaragdis et al., 2014] were proposed to solve such a problem. Nevertheless, the temporal weights containing negative values obtained by classical ICA methods bring some difficulties for the physical interpretation.

In this experiment, a single-channel polyphonic piano record played from the music
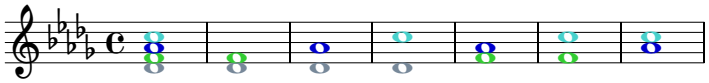


(a) Estimation accuracy of the source vector $\boldsymbol{s}$ as a function of SNR



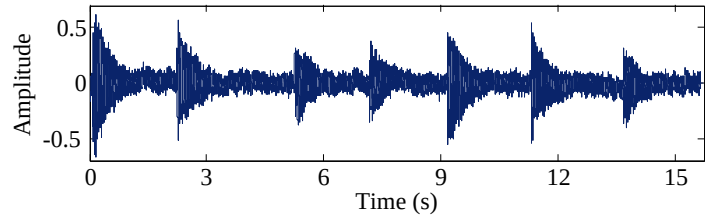
(b) Numerical complexity as a function SNR

Figure 4.5: Influence of SNR for estimating 2 MRS source metabolites from 20 observations.

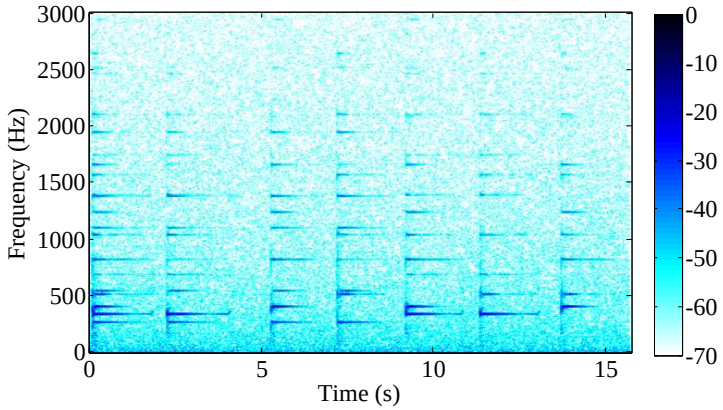
sheet in figure 4.6(a) is adopted [Févotte et al., 2009]. Such a 15.6 seconds record sampled at 22.05 kHz is made up from four different notes: C_5, A_4^b, F_4 and D_4^b (from top to bottom in figure 4.6(a)). Theoretically a note consists of several related sine waves called the fundamental and the harmonics. The sound of a note has a harmonic frequency spectrum [Burg, 2008]. The lowest frequency present is the fundamental frequency, and is the frequency at which the entire wave vibrates. A harmonic is a component frequency of a signal that is an integer multiple of the fundamental frequency. For example, if the fundamental frequency is f, the harmonics have frequencies $2f$, $3f$, $4f$, ... , etc. The theoretical fundamental frequencies of four notes in this experiment are given in table 4.1. The piano record is contaminated by an additive pink noise with a SNR value of 5 dB. A pink noise occurs in many digital recording devices, whose power spectral density is inversely proportional to the frequency of the signal. The spectrogram of the noisy record is computed using a hamming window of 125 milliseconds length with 50% overlap between two frames, leading to 251 frames and 1379 frequency bins. Moreover, the time domain waveform of the record and its magnitude spectrogram are presented in figures



(a) Music sheet



(b) Time domain waveform



(c) Magnitude spectrogram where the values are shown in logarithm scale

Figure 4.6: Three representations of the music data. The music waveform is contaminated by pink noises with a SNR value of 5 dB.

musical note	C_5	A_4^b	F_4	D_4^b
fundamental frequency (Hz)	523.251	415.305	349.228	277.183

Table 4.1: The theoretical fundamental frequencies of four musical notes, namely C_5, A_4^b, F_4 and D_4^b, on an ideal piano [Jorgensen, 1991].

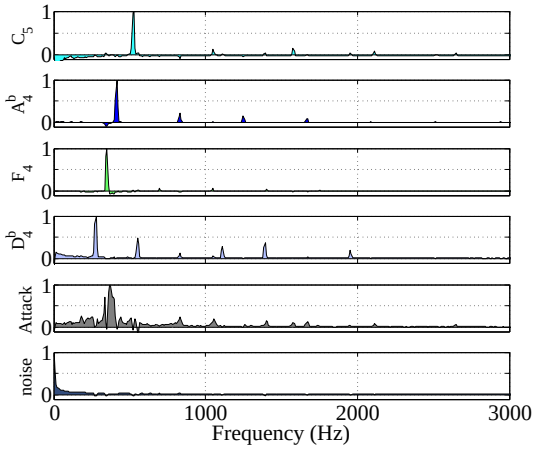
4.6(b) and 4.6(c), respectively.

Two of the proposed methods are chosen to solve this problem, namely $\text{JDC}^+_{\text{LU-2}}$-ICA and $\text{JDC}^+_{\text{ADMM-3}}$-ICA, since they provide a good compromise between estimation accuracy and numerical complexity. It is noteworthy that computing the FO cumulant array of the magnitude spectrogram $\boldsymbol{X} \in \mathbb{R}_+^{251\times 1379}$ requires at least $(251^4 \times 8)/2^{30} \approx 29.6$ GB memories, which is beyond the capacity of ordinary computers. Therefore, for the proposed methods, we propose to use NN-COMP as a preprocessing step. The proposed $\text{JDC}^+_{\text{LU-2}}$-ICA and $\text{JDC}^+_{\text{ADMM-3}}$-ICA incorporating NN-COMP are compared with CoM2 [Comon, 1994], NICA [Plumbley, 2003], NMF [Kim and Park, 2008] and semi-NMF [Ding et al., 2010]. Figure 4.7 displays the separated spectra obtained by the considered six methods. It can be seen that the proposed NN-COMP + $\text{JDC}^+_{\text{LU-2}}$-ICA and NN-COMP + $\text{JDC}^+_{\text{ADMM-3}}$-ICA methods, as well as CoM2, result in six components. The first four components correspond to the note spectra. The fundamental frequencies

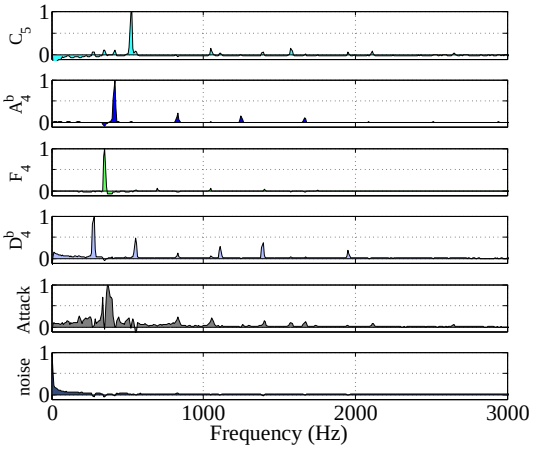
	Numerical complexity (flops)
CoM2	6.9336×10^7
NICA	2.3312×10^8
NMF	5.5786×10^9
Semi-NMF	2.1190×10^9
NN-COMP + $\text{JDC}^+_{\text{LU-2}}$-ICA	2.4355×10^9
NN-COMP + $\text{JDC}^+_{\text{ADMM-3}}$-ICA	2.4364×10^9

Table 4.2: Numerical complexities of six BSS methods in the experiment of automatic music transcription of a piano record.

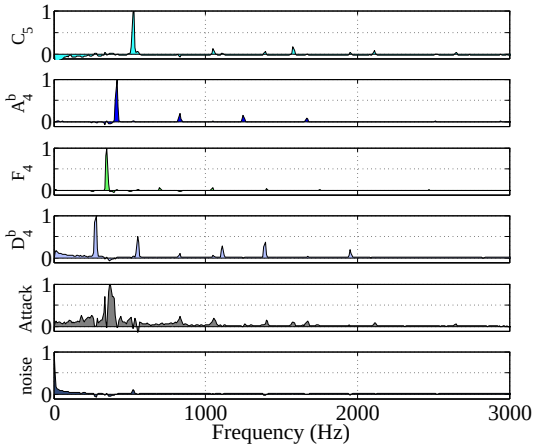
(dominant peaks) and several harmonics (small peaks) are almost correctly located in the right positions, despite some small interferences. As we can see from the first four rows in figures 4.7(a) to 4.7(c), four dominant peaks appear around 523.251 Hz, 415.305 Hz, 349.228 Hz and 277.183 Hz, respectively, that accurately describe the frequency patterns of notes C_5, A_4^b, F_4 and D_4^b, respectively. The fifth component seems to capture the transient events corresponding to the attacks and releases of the notes, more precisely, the sound produced by the hammer hitting the string and by the release of the sustain pedal. The sixth one appears to consist of the pink noise spectrum. Figures 4.7(d) and 4.7(e) show that NICA and NMF rather successfully extract notes A_4^b and F_4 into separate components, however, note C_5 is buried by the noise spectrum in the first component, while notes D_4^b and C_5 are melted into the fourth component. Regarding semi-NMF, it also fails to separate notes D_4^b and C_5, and note F_4 appears twice in the separated spectra. Figure 4.8 illustrates the temporal weight matrix $\boldsymbol{A}$ obtained by the six methods, where each row in each subfigure represents the amplitude of the corresponding column vector of $\boldsymbol{A}$. The locations and widths of the peaks appearing in each column vector of $\boldsymbol{A}$ characterize the the onset times and durations of one note. Therefore, the correctness of the result can be verified by checking if the temporal weights coincide with the original music sheet. For the proposed NN-COMP + $\text{JDC}^+_{\text{LU-2}}$-ICA and NN-COMP + $\text{JDC}^+_{\text{ADMM-3}}$-ICA methods, as well as CoM2, the locations of the peaks in the first four rows of figures 4.8(b) to 4.8(d) well match the original music sheet in figure 4.8(a), while the widths of these peaks seem identical since all the notes have the same duration. It is worth noting that in figure 4.8(d) the temporal weights of note C_5 obtained by CoM2 contain a large amount of negative values, which could bring some difficulties for the physical interpretation. The nonnegativity constraint of the weight matrix $\boldsymbol{A}$ helps the results of the proposed methods giving a more realistic physical meaning. Concerning NICA and NMF, the estimated weights for note C_5 are obviously incorrect, as illustrated in figures 4.8(e) and 4.8(f). Semi-NMF gives the worst result, since all the peaks in figure 4.8(g) are covered by noises. Additionally, the numerical complexities of all the six methods are listed in table 4.2. CoM2 has the lowest complexity. The complexities of the proposed NN-COMP + $\text{JDC}^+_{\text{LU-2}}$-ICA and NN-COMP + $\text{JDC}^+_{\text{ADMM-3}}$-ICA methods are moderately higher than that of CoM2, and between those of semi-NMF and NMF. Therefore, in this experimental context, the proposed SeNICA algorithms considering both the statistical independence of the source spectra and the nonnegativity of the temporal weights improve the separation result and also provide a good accuracy/complexity tradeoff.



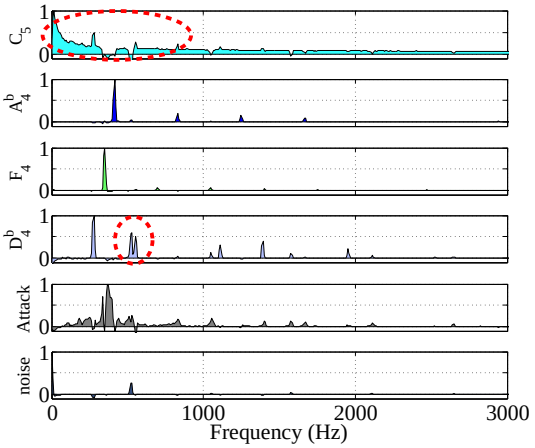
(a) Separated spectra obtained by the proposed NN-COMP + $\text{JDC}^{+}_{\text{LU-2}}$-ICA



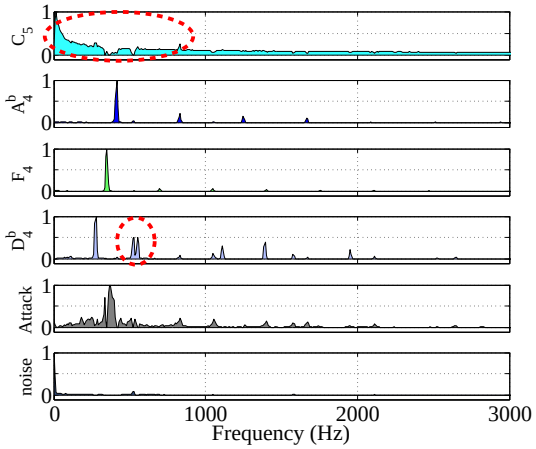
(b) Separated spectra obtained by the proposed NN-COMP + $\text{JDC}^{+}_{\text{ADMM-3}}$-ICA



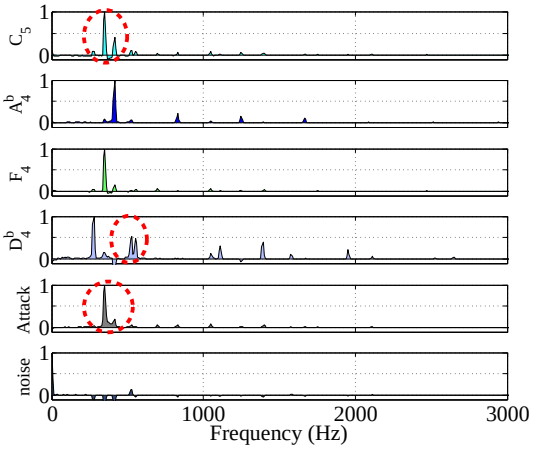
(c) Separated spectra obtained by CoM2



(d) Separated spectra obtained by NICA

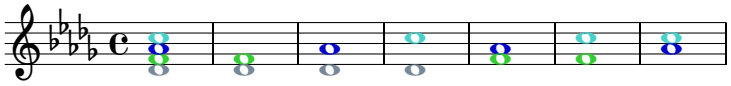


(e) Separated spectra obtained by NMF

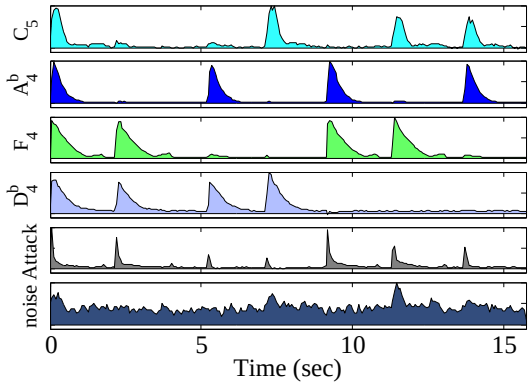


(f) Separated spectra obtained by Semi-NMF

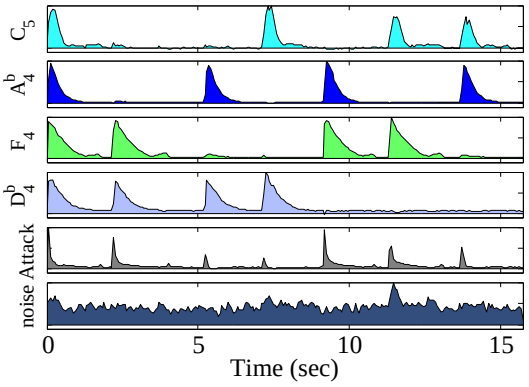
Figure 4.7: Separated spectra obtained from six BSS methods from decomposing the music magnitude spectrogram. The red dot circles indicate the significant estimation errors in figures (d) to (f).



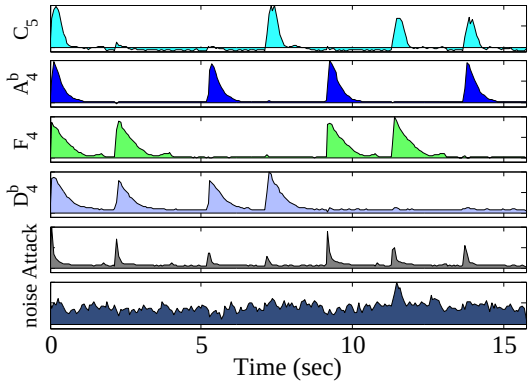
(a) Music sheet



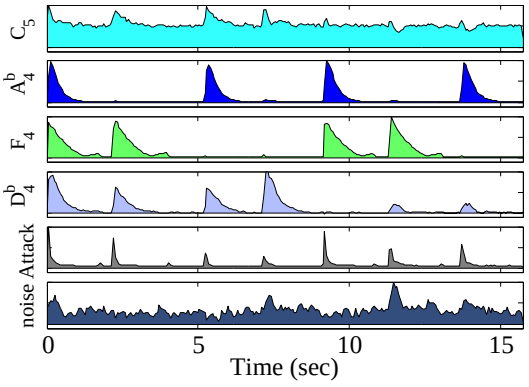
(b) Temporal weights obtained by the proposed NN-COMP + $\text{JDC}^{+}_{\text{LU-2}}$-ICA



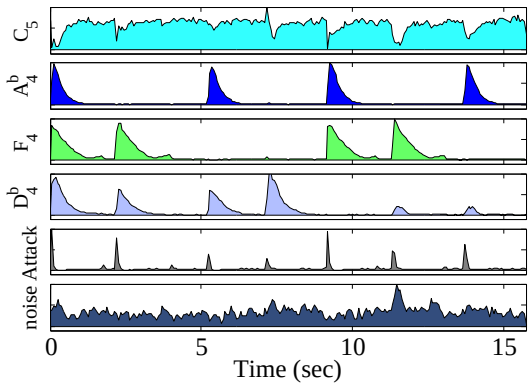
(c) Temporal weights obtained by the proposed NN-COMP + $\text{JDC}^{+}_{\text{ADMM-3}}$-ICA



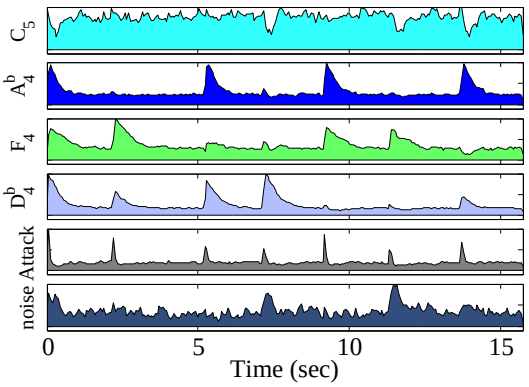
(d) Temporal weights obtained by CoM2



(e) Temporal weights obtained by NICA



(f) Temporal weights obtained by NMF



(g) Temporal weights obtained by Semi-NMF

Figure 4.8: Temporal weights obtained by six BSS methods by decomposing the music magnitude spectrogram. Each row in each subfigure represents the amplitude of the corresponding column vector of matrix $\boldsymbol{A}$.

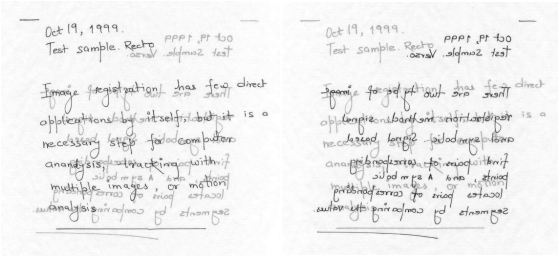
	Numerical complexity (flops)
CoM2	3.3017×10^7
NICA	7.7118×10^8
NMF	1.0613×10^{11}
Semi-NMF	1.4150×10^8
$\text{JDC}^+_{\text{LU-2}}$-ICA	2.1279×10^7
$\text{JDC}^+_{\text{ADMM-3}}$-ICA	2.1272×10^7

Table 4.3: Numerical complexities of six BSS methods in the experiment of document restoration.

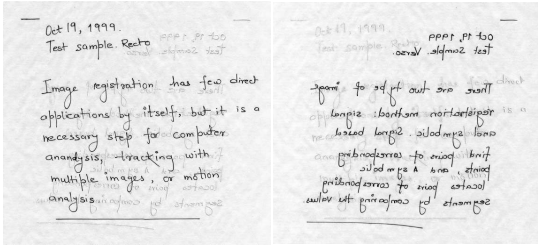
4.3 Document restoration

Some digital images of documents are degraded by the so-called show-through effect, which is usually caused by the seeping of ink from the reverse side or scanning a semi-transparent paper [Tonazzini et al., 2010]. In [Tonazzini et al., 2004, Tonazzini et al., 2007], Tonazzini et al. modeled the show-through effect as a linear instantaneous superimposition of the back and the front sides of the scanned images, and achieved the separation of two sides of images by using ICA. Thereafter, Merrikh-Bayat et al. proposed a NMF based technique for removing show-through [Merrikh-Bayat et al., 2010]. It suggests that imposing the nonnegativity constraint can improve the result. In addition, better separation result can be achieved by considering more sophisticated models, such as the linear convolutive mixing model [Tonazzini et al., 2010], the linear-quadratic mixing model [Merrikh-Bayat et al., 2011] and the nonlinear mixing model [Almeida and Almeida, 2012], however, leading to higher numerical complexities.

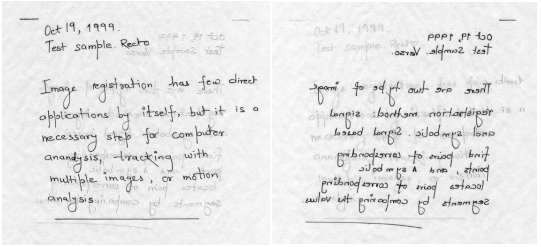
In this experiment, we apply six BSS methods to a real double-sided grayscale document of dimension (1136×1038), which is suffered from strong show-through effect. The considered methods include CoM2 [Comon, 1994], NICA [Plumbley, 2003], a projected gradient based NMF [Merrikh-Bayat et al., 2010], semi-NMF [Ding et al., 2010], and two of the proposed methods, namely $\text{JDC}^+_{\text{LU-2}}$-ICA and $\text{JDC}^+_{\text{ADMM-3}}$-ICA. The input front side and back side images displayed in figures 4.9(a) serve as 2 observations. Figures 4.9(b) to 4.9(g) show the restored front side and back side images by the considered six methods, respectively. It can be seen that for CoM2, NMF and the proposed methods, the show-through effect is mostly canceled. Their separation results are similar, although some small discrepancies exist in the detailed structure of the restored images, such as the straight line in the bottom of the left image. The results of $\text{JDC}^+_{\text{LU-2}}$-ICA and $\text{JDC}^+_{\text{ADMM-3}}$-ICA are visually identical. NICA and semi-NMF seem not able to separate both sides of images. Furthermore, the numerical complexities of all the methods are listed in table 4.3. It shows that NMF requires the heaviest workload, while the proposed $\text{JDC}^+_{\text{LU-2}}$-ICA and $\text{JDC}^+_{\text{ADMM-3}}$-ICA algorithms are the most computationally efficient. Hence, the proposed methods offer the best performance/complexity compromise in this experiment.



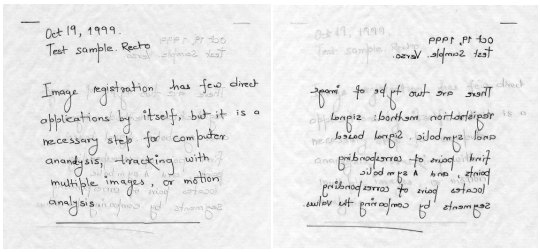
(a) Input front and back side images



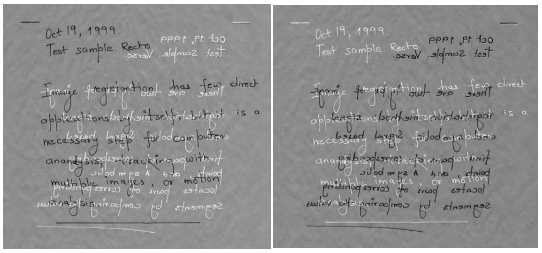
(b) Restored images by $\text{JDC}^{+}_{\text{LU-2}}$-ICA



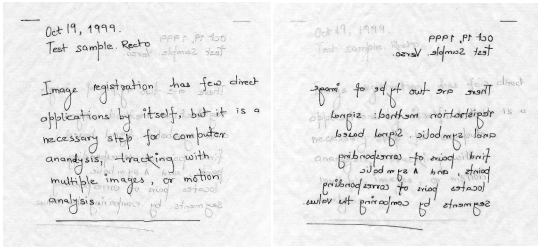
(c) Restored images by $\text{JDC}^{+}_{\text{ADMM-3}}$-ICA



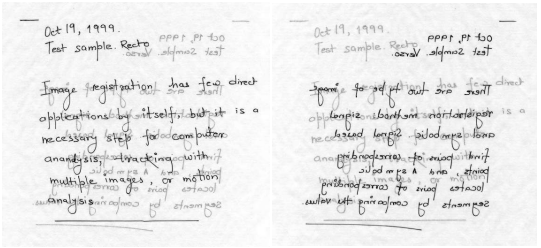
(d) Restored images by CoM2



(e) Restored images by NICA



(f) Restored images by NMF



(g) Restored images by Semi-NMF

Figure 4.9: Restoration of the grayscale front and back sides of a real document by six BSS methods. The input images are obtained from [Tonazzini et al., 2007].

4.4 Concluding remark

SeNICA turns up in many real-life BSS problems, which involve nonnegative static mixtures, such as in the blind separation of MRS signal, automatic music transcription of single-channel piano record and restoration of documents degraded by the show-through effect. The proposed SeNICA methods are applied to solve these problems. We compare them to five state-of-the-art BSS algorithms, namely two efficient ICA methods CoM2 [Comon, 1994] and SOBI [Belouchrani et al., 1997], NICA [Plumbley, 2003], NMF [Kim and Park, 2008, Merrikh-Bayat et al., 2010] and semi-NMF [Ding et al.,

2010]. Based on the presented studies, NICA, SOBI and semi-NMF sometimes fail to separate the sources. CoM2 and NMF offer better results, despite some imperfections. The proposed methods are most likely to provide better results than the classical methods, in terms of estimation accuracy of the sources, meanwhile maintaining moderate numerical complexities. These BSS applications demonstrate the validity and potential usefulness of the proposed algorithms. The experimental results have also shown that by fully exploiting the prior information of data, such as the nonnegativity of the mixing matrix and the statistical independence of the sources, the proposed methods achieve better estimation results.

Chapter 5

Conclusion and perspectives

5.1 Conclusion

The purpose of this thesis is motivated by the SeNICA problem, where the mixing matrix is constrained to have nonnegative components, and where the statistically independent sources may have mixed signs. Such a problem can be solved by means of JDC algorithms with a nonnegativity constraint on the joint transformation matrix, leading to the concept of NJDC. Therefore, we develop six novel NJDC algorithms, which are presented thoroughly in chapter 2. The six proposed methods can be separated into two groups according to the way of performing the optimization. The first group, containing five algorithms, is based on the Jacobi-like optimization. In this group, the nonnegativity constraint is ensured by a square change of variable. The general idea of our Jacobi-like optimization is to factorize the square root of the nonnegative joint transformation matrix as a product of elementary matrices which is defined by only one parameter, then to estimate these elementary matrices one by one in a specific order. The first method adopts the LU matrix factorization and performs the optimization on a simple direct-fit criterion, however, suffers from a high numerical complexity. In order to reduce the numerical complexity, we reformulate the indirect-fit criterion and consider both the LU and QR factorizations. The proposed reformulation simplifies the optimization and leads to two computationally efficient algorithms. Nevertheless, the validity of these two algorithms relies on some crucial assumptions ($\boldsymbol{A}$ is square and $\boldsymbol{D}^{(k)}$ does not contain zero diagonal elements). Consequently, the fourth and fifth algorithms that depend on less assumption are proposed. These two methods are based on a direct optimization of two indirect-fit criteria incorporating the LU matrix factorization. The second group, containing one algorithm, is based on the ADMM optimization. This algorithm is motivated by the fact that the NJDC problem can be recast as a direct least-squares fit problem subject to symmetry and nonnegativity constraints. Then such an algorithm is derived by successively minimizing the augmented Lagrangian function of the direct-fit criterion with respect to the variables and the multipliers. In addition, the numerical complexity of each algorithm has been analyzed, allowing a fair comparison of their performance.

In chapter 3, the performance of the proposed NJDC algorithms is evaluated on simulated jointly diagonalizable matrices. Five classical nonorthogonal JDC methods without nonnegativity constraints, including ACDC [Yeredor, 2002], CLU [Maurandi

et al., 2013], FFDIAG [Ziehe et al., 2004], LUJ1D [Afsari, 2006], LUJ2D [Afsari, 2006] and one NJDC method $\text{LM}^{+}_{\text{sym}}$ [Coloigner et al., 2014c], are tested as baseline methods. The performance is assessed in terms of the matrix estimation accuracy and the numerical complexity. Throughout the chapter, i) the convergence property, ii) the influence of model errors, iii) the impact of the number of input matrices, iv) the effect of coherence of the column vectors of the joint transformation matrix, and v) the influence of the modulus of uniqueness, are extensively studied through Monte Carlo experiments. Simulation results show that a better estimation accuracy can be achieved particularly in difficult contexts by means of exploiting the nonnegativity *a priori*. For a difficult NJDC problem with either a lower value of SNR, a small number of input matrices, a high level of coherence presented in the joint transformation matrix or a large value of modulus of uniqueness, generally the proposed Jacobi-like algorithms provide the most accurate estimation results. Among them, the $\text{JDC}^{+}_{\text{LU-2}}$ algorithm offers the best compromise between estimation accuracy and numerical complexity. When the considered NJDC problem is well-conditioned, that is to say, when a high value of SNR can be expected or a large number of input matrices is available, the algorithm based on the ADMM optimization, namely $\text{JDC}^{+}_{\text{ADMM-3}}$, is the most effective in terms of the estimation accuracy and is also computationally efficient. Compared to an existing nonnegative method based on the Levenberg Marquardt optimization [Coloigner et al., 2014c] whose performance is also competitive in some experiments, the proposed methods require less computation load, and save the total numerical complexity by up to three orders of magnitude. Generally speaking, the proposed $\text{JDC}^{+}_{\text{LU-2}}$ and $\text{JDC}^{+}_{\text{ADMM-3}}$ algorithms offer the best performance/complexity tradeoff.

In chapter 4, we apply the proposed NJDC algorithms on a set of FO cumulant matrix slices, giving birth to a class of SeNICA methods. SeNICA turns up in quite a number of real-life BSS problems, which involve nonnegative static mixtures, such as in the blind separation of MRS signals, automatic music transcription of single-channel piano records and restoration of documents degraded by the show-through effect. The proposed SeNICA methods are applied to solve these problems in order to verify their source separation ability. Our methods are compared with five state-of-the-art BSS algorithms, namely two efficient ICA methods CoM2 [Comon, 1994] and SOBI [Belouchrani et al., 1997], three nonnegativity constrained methods NICA [Plumbley, 2003], NMF [Kim and Park, 2008, Merrikh-Bayat et al., 2010] and semi-NMF [Ding et al., 2010]. Based on the obtained results, the classical methods, especially NICA, SOBI and semi-NMF sometimes fail to separate the latent sources. CoM2 and NMF offer better results, despite some imperfections. The proposed methods behave similarly and are most likely to provide better results than the classical methods, in terms of estimation accuracy of the sources. Our methods maintain moderate numerical complexities if the number of observations is not too large. However, their complexities grow very fast as the number of observations increases. Fortunately, by incorporating a nonnegative compression preprocessing step, the overall complexity of the proposed SeNICA procedure is acceptable. These BSS applications demonstrate the validity and potential usefulness of the proposed algorithms. The experimental results have also shown that by fully exploiting the prior information of data, such as the nonnegativity of the mixing matrix and the statistical independence of the sources, the proposed methods achieve better estimation results.

5.2 Forthcoming work

The future work is twofold. The first part consists of improving the NJDC methods proposed in this PhD dissertation. The second part is to develop novel BSS methods which are more suitable for separating partially correlated sources.

Throughout the chapter 2, our effort is not only to develop five Jacobi-like NJDC algorithms but also to reduce their potential numerical complexities by combining them with classical JDC algorithms. Nevertheless, the proposed Jacobi-like NJDC algorithms are still more costly than the classical unconstrained JDC algorithms, even though the numerical complexities of our methods are lower than that of one of the existing NJDC algorithms, which is based on the LM optimization. The high complexity reported in chapters 3 and 4 seems the main limitation of our methods, and restricts their usage especially for high dimensional data with a large number of observations and of sources. Fortunately, besides these Jacobi-like methods, the proposed NJDC algorithm based on the ADMM optimization and the direct-fit JDC criterion requires a smaller computational load which is similar to that of classical algorithms, and maintains competitive performance in terms of estimation accuracy. It seems that the ADMM optimization provides a good compromise between performance and complexity especially for large scale data. Consequently, we would like to continue to derive novel NJDC methods based on the minimization of the indirect-fit JDC criteria by using the ADMM approach. Since the indirect-fit JDC criteria involve less variables than the direct-fit criterion, we can expect that the forthcoming algorithms could be more efficient than the existing one. It is noteworthy that we will face two difficulties. The first potential difficulty is to impose the nonnegativity on the joint transformation matrix $\boldsymbol{A}$, meanwhile performing the ADMM optimization on its inverse $\boldsymbol{A}^{-1}$. The second one is to avoid the trivial solution $\boldsymbol{A}^{-1} = \boldsymbol{0}$ in the ADMM procedure. In the proposed Jacobi-like methods based on the indirect-fit criteria, the nonsingularity of $\boldsymbol{A}$ is guaranteed by the LU and QR decompositions, since the resulting matrix has a unit-determinant. In the ADMM method with the indirect-fit criteria, an additional nonsingularity constraint on $\boldsymbol{A}$ seems necessary. Moreover, further theoretical studies of the convergence of these approaches will be considered. On the other hand, recent studies on nonorthogonal Jacobi-like JDC algorithms showed that the convergence rate of the Jacobi-like procedure can be accelerated by considering the coupled LU and QR matrix factorizations [Maurandi et al., 2013, Maurandi and Moreau, 2014a, Maurandi and Moreau, 2014b, Maurandi et al., 2014]. We would like to develop new Jacobi-like NJDC algorithms by using such coupled factorizations in order to accelerate convergence and then to reduce the numerical complexity.

In chapter 4, we illustrate the usefulness of our SeNICA methods in the BSS framework. The success of SeNICA, as well as ICA and NICA, relies on the statistical independence, or at least on the quasi-statistical independence of the sources. However, in many practical problems, this assumption does not always hold. Hence, ICA, NICA and our SeNICA methods look for latent sources that are as statistically independent as possible. The degree of independence is measured through the use of mathematical tools from HOS, information theory and many others. Estimating the sources by minimizing the statistical dependence may not give satisfactory results, especially when the true source signals are known to be correlated [Naanaa and Nuzillard, 2005]. This situation appears in a number of real-world applications, such as hyperspectral unmixing [Chan

et al., 2009, Ma et al., 2014] and medical imaging [Wang et al., 2010]. Regarding the MRS data that we treated in this thesis, sometimes the source spectra can be highly correlated if the molecules share common structural features and therefore are difficult to separate. Consequently, an accurate separation of correlated or dependent sources still remains a challenging task. In [Wang et al., 2010], Wang et al. proposed a nonnegative Least-correlated Component Analysis (nLCA) method to separate nonnegative correlated sources by minimizing a joint correlation objective function among the estimated sources. Such an objective function assumes that the nonnegative sources are linearly independent rather than statistically independent, and exploits the convex geometry of the nonnegative mixtures of nonnegative sources. nLCA was shown to have a superior performance on real biomedical data over several existing benchmark methods, for instance, ICA and NMF. However, this method is constrained to work on a square mixing matrix and is sensitive to additional noise. We are working on a novel method which generalizes the nLCA algorithm to overdetermined mixtures of sources. Preliminary experimental results show that this novel method is more robust to additional noise and is also computationally efficient. Further studies on separating correlated MRS signals will be considered in order to systematically evaluate the performance of the new method.

Appendix A

Proofs of propositions

A.1 Proof of proposition 2.1

Multiplying $\widehat{\boldsymbol{B}}$ by an elementary triangular matrix $\boldsymbol{T}^{(i,j)}(t_{i,j})$ on the right side performs the elementary transformation of the j-th column vector of $\widehat{\boldsymbol{B}}$:

$$\widehat{\boldsymbol{B}}\boldsymbol{T}^{(i,j)}(t_{i,j}) = [\widehat{\boldsymbol{b}}_1, \widehat{\boldsymbol{b}}_2, \ldots, \underbrace{t_{i,j}\widehat{\boldsymbol{b}}_i + \widehat{\boldsymbol{b}}_j}_{j\text{-th column}}, \ldots, \widehat{\boldsymbol{b}}_N] \tag{A.1}$$

where $\widehat{\boldsymbol{b}}_i$, $i \in \{1, 2, \ldots, N\}$ is the i-th column vector of $\boldsymbol{B}$. By routine computation, the Hadamard square of equation (A.1) is:

$$\begin{aligned}
&[\widehat{\boldsymbol{b}}_1^{\boxdot 2}, \ldots, \underbrace{t_{i,j}^2\,\widehat{\boldsymbol{b}}_i^{\boxdot 2} + \widehat{\boldsymbol{b}}_j^{\boxdot 2} + 2t_{i,j}\,\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j}_{j\text{-th column}}, \ldots, \widehat{\boldsymbol{b}}_N^{\boxdot 2}] \\
&= [\widehat{\boldsymbol{b}}_1^{\boxdot 2}, \ldots, \underbrace{t_{i,j}^2\,\widehat{\boldsymbol{b}}_i^{\boxdot 2} + \widehat{\boldsymbol{b}}_j^{\boxdot 2}}_{j\text{-th column}}, \ldots, \widehat{\boldsymbol{b}}_N^{\boxdot 2}] + [\boldsymbol{0}, \ldots, \underbrace{2t_{i,j}\,\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j}_{j\text{-th column}}, \ldots, \boldsymbol{0}] \\
&= \widehat{\boldsymbol{B}}^{\boxdot 2}\boldsymbol{T}^{(i,j)}(t_{i,j}^2) + 2\,t_{i,j}(\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)\boldsymbol{e}_j^{\mathsf{T}}
\end{aligned} \tag{A.2}$$

A.2 Proof of proposition 2.2

It is straightforward to show that the elements of the j-th column vector except the (j,j)-th entry of the term ① in equation (2.51) can be expressed by $\widehat{C}_{n,i}^{(k)}u_{i,j}^2 + \widehat{C}_{n,j}^{(k)}$ with $n \in \{1, \ldots, N\}$ and $n \neq j$, and those elements of the term ② in (2.51) are equal to $\widehat{c}_n^{(k,1)}u_{i,j}$ with $n \in \{1, \ldots, N\}$ and $n \neq j$. The sum of these elements directly leads to equation (2.52). The terms ③ and ④ in equation (2.51) do not need to be considered, since they do not affect the off-diagonal elements in the j-th column vector.

A.3 Proof of proposition 2.3

Multiplying $\widehat{\boldsymbol{B}}$ by a Givens rotation matrix $\boldsymbol{Q}^{(i,j)}(\theta_{i,j})$ on the right side performs the elementary transformation of the i-th and j-th column vectors of $\widehat{\boldsymbol{B}}$:

$$\widehat{\boldsymbol{B}}\boldsymbol{Q}^{(i,j)}(\theta_{i,j}) = [\widehat{\boldsymbol{b}}_1, \ldots, \underbrace{\cos(\theta_{i,j})\,\widehat{\boldsymbol{b}}_i + \sin(\theta_{i,j})\,\widehat{\boldsymbol{b}}_j}_{i\text{-th column}}, \ldots, \underbrace{-\sin(\theta_{i,j})\,\widehat{\boldsymbol{b}}_i + \cos(\theta_{i,j})\,\widehat{\boldsymbol{b}}_j}_{j\text{-th column}}, \ldots, \widehat{\boldsymbol{b}}_N] \tag{A.3}$$

where $\widehat{\boldsymbol{b}}_i$, $i \in \{1, 2, \ldots, N\}$ is the i-th column vector of $\boldsymbol{B}$. By routine computation, the Hadamard square of equation (A.3) is:

$$\begin{aligned}
&[\widehat{\boldsymbol{b}}_1^{\boxdot 2}, \ldots, \underbrace{\sin^2(\theta_{i,j})\,\widehat{\boldsymbol{b}}_i^{\boxdot 2} + \cos^2(\theta_{i,j})\,\widehat{\boldsymbol{b}}_j^{\boxdot 2} + \sin(2\theta_{i,j})\,\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j}_{i\text{-th column}}, \ldots, \\
&\qquad \underbrace{\cos^2(\theta_{i,j})\,\widehat{\boldsymbol{b}}_i^{\boxdot 2} + \sin^2(\theta_{i,j})\,\widehat{\boldsymbol{b}}_j^{\boxdot 2} - \sin(2\theta_{i,j})\,\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j}_{j\text{-th column}}, \ldots, \widehat{\boldsymbol{b}}_N^{\boxdot 2}] \\
&= [\widehat{\boldsymbol{b}}_1^{\boxdot 2}, \ldots, \underbrace{\sin^2(\theta_{i,j})\,\widehat{\boldsymbol{b}}_i^{\boxdot 2} + \cos^2(\theta_{i,j})\,\widehat{\boldsymbol{b}}_j^{\boxdot 2}}_{i\text{-th column}}, \ldots, \underbrace{\cos^2(\theta_{i,j})\,\widehat{\boldsymbol{b}}_i^{\boxdot 2} + \sin^2(\theta_{i,j})\,\widehat{\boldsymbol{b}}_j^{\boxdot 2}}_{j\text{-th column}}, \ldots, \widehat{\boldsymbol{b}}_N^{\boxdot 2}] \\
&\quad + \sin(2\theta_{i,j})[\boldsymbol{0}, \ldots, \underbrace{\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j}_{i\text{-th column}}, \ldots, \underbrace{-\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j}_{j\text{-th column}}, \ldots, \boldsymbol{0}]
\end{aligned} \tag{A.4}$$

Clearly, equation (A.4) is equal to $\widehat{\boldsymbol{B}}^{\boxdot 2}\,(\boldsymbol{Q}^{(i,j)}(\theta_{i,j}))^{\boxdot 2} + \sin(2\theta_{i,j})(\widehat{\boldsymbol{b}}_i \boxdot \widehat{\boldsymbol{b}}_j)(\boldsymbol{e}_i^{\mathsf{T}} - \boldsymbol{e}_j^{\mathsf{T}})$.

A.4 Proof of proposition 2.4

It is straightforward to show that the (i,j)-th entry of the term ① in (2.61) can be expressed by $\sin^2(\theta_{i,j})\cos^2(\theta_{i,j})(\widehat{C}_{i,i}^{(k)} + \widehat{C}_{j,j}^{(k)}) + \sin^4(\theta_{i,j})\widehat{C}_{j,i}^{(k)} + \cos^4(\theta_{i,j})\widehat{C}_{i,j}^{(k)}$, the (i,j)-th element of the term ② in (2.61) is $\sin(2\theta_{i,j})(\cos^2(\theta_{i,j})\widehat{c}_i^{(k,1)} + \sin^2(\theta_{i,j})\widehat{c}_j^{(k,1)})$, the (i,j)-th component of the term ③ in (2.61) is equal to $-\sin(2\theta_{i,j})(\sin^2(\theta_{i,j})\widehat{c}_i^{(k,2)} + \cos^2(\theta_{i,j})\widehat{c}_j^{(k,2)})$, and that of the term ④ in (2.61) is $-\sin^2(2\theta_{i,j})\widehat{c}^{(k,3)}$. Then proposition 2.4 can be proved.

A.5 Proofs of propositions 2.6 and 2.7

Let's expand equation (2.86) as follows:

$$\begin{aligned}
\widehat{\boldsymbol{C}}^{(k,\text{new})} &= \boldsymbol{T}^{(i,j)}(-t_{i,j}^2)\,\boldsymbol{Q}\,\widehat{\boldsymbol{C}}^{(k)}\boldsymbol{Q}^{\mathsf{T}}\,\boldsymbol{T}^{(i,j)}(-t_{i,j}^2)^{\mathsf{T}} \\
&= \boldsymbol{T}^{(i,j)}(-t_{i,j}^2)(\boldsymbol{I}_N - \boldsymbol{Q}_1)\,\widehat{\boldsymbol{C}}^{(k)}(\boldsymbol{I}_N - \boldsymbol{Q}_1^{\mathsf{T}})\,\boldsymbol{T}^{(i,j)}(-t_{i,j}^2)^{\mathsf{T}} \\
&= \underbrace{\boldsymbol{T}^{(i,j)}(-t_{i,j}^2)\widehat{\boldsymbol{C}}^{(k)}\boldsymbol{T}^{(i,j)}(-t_{i,j}^2)^{\mathsf{T}}}_{①} + \underbrace{\boldsymbol{T}^{(i,j)}(-t_{i,j}^2)\boldsymbol{Q}_1\widehat{\boldsymbol{C}}^{(k)}\boldsymbol{T}^{(i,j)}(-t_{i,j}^2)^{\mathsf{T}}}_{②} \\
&\quad + \underbrace{\boldsymbol{T}^{(i,j)}(-t_{i,j}^2)\widehat{\boldsymbol{C}}^{(k)}\boldsymbol{Q}_1^{\mathsf{T}}\boldsymbol{T}^{(i,j)}(-t_{i,j}^2)^{\mathsf{T}}}_{③} + \underbrace{\boldsymbol{T}^{(i,j)}(-t_{i,j}^2)\boldsymbol{Q}_1\widehat{\boldsymbol{C}}^{(k)}\boldsymbol{Q}_1^{\mathsf{T}}\boldsymbol{T}^{(i,j)}(-t_{i,j}^2)^{\mathsf{T}}}_{④}
\end{aligned} \tag{A.5}$$

where $\boldsymbol{Q}_1 = 2\,t_{i,j}/(1+2\beta_j t_{i,j})\boldsymbol{\beta}\boldsymbol{e}_j^\mathsf{T}$, where $\boldsymbol{\beta}$ is defined in equation (2.85), and where β_j is the j-th element of vector $\boldsymbol{\beta}$. In the following, we compute the components of the terms ①, ②, ③ and ④ in equation (A.5).

In ①, $\boldsymbol{T}^{(i,j)}(-t_{i,j}^2)$ and $\boldsymbol{T}^{(i,j)}(-t_{i,j}^2)^\mathsf{T}$ transform the i-th row vector and i-th column vector of matrix $\widehat{\boldsymbol{C}}^{(k)}$ as follows:

$$① = \begin{array}{l} \\ i\text{-th row} - \end{array} \begin{array}{c} \\ \\ \\ \\ \\ \\ \end{array} \overset{\overset{i\text{-th column}}{\downarrow}}{\begin{pmatrix} \widehat{C}_{1,1}^{(k)} & \widehat{C}_{1,2}^{(k)} & \dots & -\widehat{C}_{1,j}^{(k)}t_{i,j}^2 + \widehat{C}_{1,i}^{(k)} & \dots & \widehat{C}_{1,N}^{(k)} \\ \widehat{C}_{2,1}^{(k)} & \widehat{C}_{2,2}^{(k)} & \dots & -\widehat{C}_{2,j}^{(k)}t_{i,j}^2 + \widehat{C}_{2,i}^{(k)} & \dots & \widehat{C}_{2,N}^{(k)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \begin{matrix}-\widehat{C}_{j,1}^{(k)}t_{i,j}^2 \\ +\widehat{C}_{i,1}^{(k)}\end{matrix} & \begin{matrix}-\widehat{C}_{j,2}^{(k)}t_{i,j}^2 \\ +\widehat{C}_{i,2}^{(k)}\end{matrix} & \dots & \begin{matrix}\widehat{C}_{j,j}^{(k)}t_{i,j}^4 - (\widehat{C}_{j,i}^{(k)} \\ +\widehat{C}_{i,j}^{(k)})t_{i,j}^2 + \widehat{C}_{i,i}^{(k)}\end{matrix} & \dots & \begin{matrix}-\widehat{C}_{j,N}^{(k)}t_{i,j}^2 \\ +\widehat{C}_{i,N}^{(k)}\end{matrix} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \widehat{C}_{N,1}^{(k)} & \widehat{C}_{N,2}^{(k)} & \dots & -\widehat{C}_{N,j}^{(k)}t_{i,j}^2 + \widehat{C}_{N,i}^{(k)} & \dots & \widehat{C}_{N,N}^{(k)} \end{pmatrix}} \tag{A.6}$$

In ②, matrix $\boldsymbol{Q}_1\widehat{\boldsymbol{C}}^{(k)}$ can be expressed as follows:

$$\boldsymbol{Q}_1\widehat{\boldsymbol{C}}^{(k)} = \frac{2\,t_{i,j}}{1+2\beta_j t_{i,j}}\,\boldsymbol{\beta}\boldsymbol{e}_j^\mathsf{T}\widehat{\boldsymbol{C}}^{(k)} = \frac{2\,t_{i,j}}{1+2\beta_j t_{i,j}}\left[\widehat{C}_{j,1}^{(k)}\boldsymbol{\beta}, \widehat{C}_{j,2}^{(k)}\boldsymbol{\beta}, \dots, \widehat{C}_{j,N}^{(k)}\boldsymbol{\beta}\right] \tag{A.7}$$

$\boldsymbol{T}^{(i,j)}(-t_{i,j}^2)$ and $\boldsymbol{T}^{(i,j)}(-t_{i,j}^2)^\mathsf{T}$ transform the i-th row vector and i-th column vector of matrix $\boldsymbol{Q}_1\widehat{\boldsymbol{C}}^{(k)}$ as follows:

$$② = \frac{2\,t_{i,j}}{1+2\beta_j t_{i,j}}\times$$

$$\begin{array}{l} \\ i\text{-th row} - \end{array} \begin{array}{c} \\ \\ \\ \\ \\ \\ \end{array} \overset{\overset{i\text{-th column}}{\downarrow}}{\begin{pmatrix} \widehat{C}_{j,1}^{(k)}\beta_1 & \widehat{C}_{j,2}^{(k)}\beta_1 & \dots & -\widehat{C}_{j,j}^{(k)}\beta_1 t_{i,j}^2 + \widehat{C}_{j,i}^{(k)}\beta_1 & \dots & \widehat{C}_{j,N}^{(k)}\beta_1 \\ \widehat{C}_{j,1}^{(k)}\beta_2 & \widehat{C}_{j,2}^{(k)}\beta_2 & \dots & -\widehat{C}_{j,j}^{(k)}\beta_2 t_{i,j}^2 + \widehat{C}_{j,i}^{(k)}\beta_2 & \dots & \widehat{C}_{j,N}^{(k)}\beta_2 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \begin{matrix}-\widehat{C}_{j,1}^{(k)}\beta_j t_{i,j}^2 \\ +\widehat{C}_{j,1}^{(k)}\beta_i\end{matrix} & \begin{matrix}-\widehat{C}_{j,2}^{(k)}\beta_j t_{i,j}^2 \\ +\widehat{C}_{j,2}^{(k)}\beta_i\end{matrix} & \dots & \begin{matrix}\widehat{C}_{j,j}^{(k)}\beta_j t_{i,j}^4 - (\widehat{C}_{j,j}^{(k)}\beta_i + \widehat{C}_{j,i}^{(k)}\beta_j)t_{i,j}^2 \\ +\widehat{C}_{j,i}^{(k)}\beta_i\end{matrix} & \dots & \begin{matrix}-\widehat{C}_{j,N}^{(k)}\beta_j t_{i,j}^2 \\ +\widehat{C}_{j,N}^{(k)}\beta_i\end{matrix} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \widehat{C}_{j,1}^{(k)}\beta_N & \widehat{C}_{j,2}^{(k)}\beta_N & \dots & -\widehat{C}_{j,j}^{(k)}\beta_N t_{i,j}^2 + \widehat{C}_{j,i}^{(k)}\beta_N & \dots & \widehat{C}_{j,N}^{(k)}\beta_N \end{pmatrix}} \tag{A.8}$$

Term ③ is the transpose of ②. Its expression is therefore omitted. In ④, matrix $\boldsymbol{Q}_1\widehat{\boldsymbol{C}}^{(k)}\boldsymbol{Q}_1^\mathsf{T}$ can be expressed as follows:

$$\boldsymbol{Q}_1\widehat{\boldsymbol{C}}^{(k)}\boldsymbol{Q}_1^\mathsf{T} = \frac{2\,t_{i,j}}{1+2\beta_j t_{i,j}}\,\boldsymbol{\beta}\boldsymbol{e}_j^\mathsf{T}\widehat{\boldsymbol{C}}^{(k)}\boldsymbol{e}_j\boldsymbol{\beta}^\mathsf{T}\frac{2\,t_{i,j}}{1+2\beta_j t_{i,j}} = \frac{4\,\widehat{C}_{j,j}^{(k)}t_{i,j}^2}{(1+2\beta_j t_{i,j})^2}\boldsymbol{\beta}\boldsymbol{\beta}^\mathsf{T} \tag{A.9}$$

$\boldsymbol{T}^{(i,j)}(-t_{i,j}^2)$ and $\boldsymbol{T}^{(i,j)}(-t_{i,j}^2)^\mathsf{T}$ transform the i-th row vector and i-th column vector of

matrix $\boldsymbol{Q}_1\widehat{\boldsymbol{C}}^{(k)}\boldsymbol{Q}_1^{\mathsf{T}}$ as follows:

$$
\textcircled{4} = \frac{4\,\widehat{C}^{(k)}_{j,j}t^2_{i,j}}{(1+2\beta_j t_{i,j})^2}\times
$$

$$
i\text{-th row} - \begin{pmatrix}
 & & & i\text{-th column} & & \\
 & & & \downarrow & & \\
\beta_1\beta_1 & \beta_2\beta_1 & \dots & -\beta_j\beta_1 t^2_{i,j}+\beta_i\beta_1 & \dots & \beta_N\beta_1 \\
\beta_1\beta_2 & \beta_2\beta_2 & \dots & -\beta_j\beta_2 t^2_{i,j}+\beta_i\beta_2 & \dots & \beta_N\beta_2 \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
\begin{matrix}-\beta_1\beta_j t^2_{i,j}\\ +\beta_1\beta_i\end{matrix} & \begin{matrix}-\beta_2\beta_j t^2_{i,j}\\ +\beta_2\beta_i\end{matrix} & \dots & \beta_j^2 t^4_{i,j}-2\beta_j\beta_i t^2_{i,j}+\beta_i^2 & \dots & \begin{matrix}-\beta_N\beta_j t^2_{i,j}\\ +\beta_N\beta_i\end{matrix} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
\beta_1\beta_N & \beta_2\beta_N & \dots & -\beta_j\beta_N t^2_{i,j}+\beta_i\beta_N & \dots & \beta_N^2
\end{pmatrix} \tag{A.10}
$$

The sum of equations (A.6), (A.8), (A.10) and the transpose of (A.8) will directly lead to propositions 2.6 and 2.7.

A.6 Proof of proposition 2.8

According to proposition 2.1, the differences between $\widehat{\boldsymbol{A}}^{(\text{new})}$ and $\widehat{\boldsymbol{A}}$ are their j-th column vectors. By the definition of Khatri-Rao product [Cichocki et al., 2009, Chapter 1], the following equality exists:

$$
\widehat{\boldsymbol{A}}\odot\widehat{\boldsymbol{A}} = \Big[\operatorname{vec}(\widehat{\boldsymbol{a}}_1\widehat{\boldsymbol{a}}_1^{\mathsf{T}}),\cdots,\operatorname{vec}(\widehat{\boldsymbol{a}}_j\widehat{\boldsymbol{a}}_j^{\mathsf{T}}),\cdots,\operatorname{vec}(\widehat{\boldsymbol{a}}_N\widehat{\boldsymbol{a}}_N^{\mathsf{T}})\Big] \tag{A.11}
$$

where $\widehat{\boldsymbol{a}}_j$ is the j-th column vector of $\widehat{\boldsymbol{A}}$. Hence, the differences between $\widehat{\boldsymbol{A}}^{(\text{new})}\odot\widehat{\boldsymbol{A}}^{(\text{new})}$ and $\widehat{\boldsymbol{A}}\odot\widehat{\boldsymbol{A}}$ are their j-th column vectors. The j-th column vector of $\widehat{\boldsymbol{A}}^{(\text{new})}\odot\widehat{\boldsymbol{A}}^{(\text{new})}$ can be expressed as follows:

$$
\begin{aligned}
&\operatorname{vec}(\widehat{\boldsymbol{a}}_j^{(\text{new})}(\widehat{\boldsymbol{a}}_j^{(\text{new})})^{\mathsf{T}}) \\
=&\operatorname{vec}\Big[(t^2_{i,j}\widehat{\boldsymbol{a}}_i+\widehat{\boldsymbol{a}}_j+2t_{i,j}\widehat{\boldsymbol{b}}_i\boxdot\widehat{\boldsymbol{b}}_j)(t^2_{i,j}\widehat{\boldsymbol{a}}_i+\widehat{\boldsymbol{a}}_j+2t_{i,j}\widehat{\boldsymbol{b}}_i\boxdot\widehat{\boldsymbol{b}}_j)^{\mathsf{T}}\Big] \\
=&\operatorname{vec}(\widehat{\boldsymbol{a}}_j\widehat{\boldsymbol{a}}_j^{\mathsf{T}})+\boldsymbol{g}_1t^4_{i,j}+\boldsymbol{g}_2t^3_{i,j}+\boldsymbol{g}_3t^2_{i,j}+\boldsymbol{g}_4t_{i,j} \\
=&\operatorname{vec}(\widehat{\boldsymbol{a}}_j\widehat{\boldsymbol{a}}_j^{\mathsf{T}})+\boldsymbol{G}\,\boldsymbol{\zeta}_{i,j}
\end{aligned} \tag{A.12}
$$

where $\{\boldsymbol{g}_i\}_{i=1}^4$, $\boldsymbol{G}$ and $\boldsymbol{\zeta}_{i,j}$ are defined in proposition 2.8. The $(N^2\times 1)$ column vector $\operatorname{vec}(\widehat{\boldsymbol{a}}_j\widehat{\boldsymbol{a}}_j^{\mathsf{T}})$ is the j-th column vector of $\widehat{\boldsymbol{A}}\odot\widehat{\boldsymbol{A}}$. Then proposition 2.8 holds.

A.7 Proof of proposition 2.9

We can rearrange the ADMM update formulas in equations (2.136) to (2.141) into the following form:

$$
\begin{aligned}
&\Big(\boldsymbol{A}_1^{(it+1)}-\boldsymbol{A}_1^{(it)}\Big)\left(\sum_{k=1}^{K}\boldsymbol{D}^{(k,it)}(\boldsymbol{A}_2^{(it)})^{\mathsf{T}}\boldsymbol{A}_2^{(it)}\boldsymbol{D}^{(k,it)}+\alpha\boldsymbol{I}_P\right) \\
=&\sum_{k=1}^{K}\Big(\boldsymbol{C}^{(k)}-\boldsymbol{A}_1^{(it)}\boldsymbol{D}^{(k,it)}(\boldsymbol{A}_2^{(it)})^{\mathsf{T}}\Big)\Big(\boldsymbol{A}_2^{(it)}\boldsymbol{D}^{(k,it)}\Big)-\boldsymbol{\Pi}_1^{(it)}+\alpha(\boldsymbol{U}^{(it)}-\boldsymbol{A}_1^{(it)})
\end{aligned} \tag{A.13}
$$

$$\left(\boldsymbol{A}_2^{(it+1)} - \boldsymbol{A}_2^{(it)}\right)\left(\sum_{k=1}^{K} \boldsymbol{D}^{(k,it)}(\boldsymbol{A}_1^{(it+1)})^{\mathsf{T}}\boldsymbol{A}_1^{(it+1)}\boldsymbol{D}^{(k,it)} + \beta \boldsymbol{I}_P\right)$$
$$= \sum_{k=1}^{K}\left(\boldsymbol{C}^{(k)} - \boldsymbol{A}_2^{(it)}\boldsymbol{D}^{(k,it)}(\boldsymbol{A}_1^{(it+1)})^{\mathsf{T}}\right)\left(\boldsymbol{A}_1^{(it+1)}\boldsymbol{D}^{(k,it)}\right) - \boldsymbol{\Pi}_2^{(it)} + \beta(\boldsymbol{U}^{(it)} - \boldsymbol{A}_2^{(it)}) \tag{A.14}$$

$$\boldsymbol{U}^{(it+1)} - \boldsymbol{U}^{(it)} = \mathrm{P}_+\left\{\left(\alpha \boldsymbol{A}_1^{(it+1)} + \beta \boldsymbol{A}_2^{(it+1)} + \boldsymbol{\Pi}_1^{(it)} + \boldsymbol{\Pi}_2^{(it)}\right)/(\alpha+\beta)\right\} - \boldsymbol{U}^{(it)} \tag{A.15}$$

$$\left[\left(\boldsymbol{A}_2^{(it+1)} \odot \boldsymbol{A}_1^{(it+1)}\right)^{\mathsf{T}}\left(\boldsymbol{A}_2^{(it+1)} \odot \boldsymbol{A}_1^{(it+1)}\right)\right]\left(\boldsymbol{d}^{(k,it+1)} - \boldsymbol{d}^{(k,it)}\right)$$
$$= \left(\boldsymbol{A}_2^{(it+1)} \odot \boldsymbol{A}_1^{(it+1)}\right)^{\mathsf{T}}\left(\boldsymbol{c}(k) - (\boldsymbol{A}_2^{(it+1)} \odot \boldsymbol{A}_1^{(it+1)})\boldsymbol{d}^{(k,it)}\right) \tag{A.16}$$

$$\boldsymbol{\Pi}_1^{(it+1)} - \boldsymbol{\Pi}_1^{(it)} = \gamma\alpha(\boldsymbol{A}_1^{(it+1)} - \boldsymbol{U}^{(it+1)}) \tag{A.17}$$

$$\boldsymbol{\Pi}_2^{(it+1)} - \boldsymbol{\Pi}_2^{(it)} = \gamma\beta(\boldsymbol{A}_2^{(it+1)} - \boldsymbol{U}^{(it+1)}) \tag{A.18}$$

where $\boldsymbol{c}^{(k)} = \mathrm{vec}(\boldsymbol{C}^{(k)})$, and where $\boldsymbol{d}^{(k,it)}$, $\boldsymbol{d}^{(k,it+1)} \in \mathbb{R}^P$ are column vectors containing the diagonal elements of the diagonal matrices $\boldsymbol{D}^{(k,it)}$ and $\boldsymbol{D}^{(k,it+1)}$, respectively. The assumption $\boldsymbol{Z}^{(it+1)} - \boldsymbol{Z}^{(it)} \to \boldsymbol{0}$ implies that the left and right hand sides of equations (A.13) to (A.18) go to zero. Let it go to infinity. We have $\boldsymbol{A}_1^{(it+1)} = \boldsymbol{A}_1^{(it)} + (\boldsymbol{A}_1^{(it+1)} - \boldsymbol{A}_1^{(it)})$, ... , and so on for each variable, where the second terms $\boldsymbol{A}_1^{(it+1)} - \boldsymbol{A}_1^{(it)}$, ... , vanishes asymptotically. Consequently, we obtain:

$$\sum_{k=1}^{K}(\boldsymbol{C}^{(k)} - \boldsymbol{A}_1^{(it+1)}\boldsymbol{D}^{(k,it+1)}(\boldsymbol{A}_2^{(it+1)})^{\mathsf{T}})(\boldsymbol{A}_2^{(it+1)}\boldsymbol{D}^{(k,it+1)}) - \boldsymbol{\Pi}_1^{(it+1)} \to \boldsymbol{0} \tag{A.19}$$

$$\sum_{k=1}^{K}(\boldsymbol{C}^{(k)} - \boldsymbol{A}_2^{(it+1)}\boldsymbol{D}^{(k,it+1)}(\boldsymbol{A}_1^{(it+1)})^{\mathsf{T}})(\boldsymbol{A}_1^{(it+1)}\boldsymbol{D}^{(k,it+1)}) - \boldsymbol{\Pi}_2^{(it+1)} \to \boldsymbol{0} \tag{A.20}$$

$$\mathrm{P}_+\left\{\left(\alpha \boldsymbol{A}_1^{(it+1)} + \beta \boldsymbol{A}_2^{(it+1)} + \boldsymbol{\Pi}_1^{(it+1)} + \boldsymbol{\Pi}_2^{(it+1)}\right)/(\alpha+\beta)\right\} - \boldsymbol{U}^{(it+1)} \to \boldsymbol{0} \tag{A.21}$$

$$(\boldsymbol{A}_2^{(it+1)} \odot \boldsymbol{A}_1^{(it+1)})^{\mathsf{T}}\left(\boldsymbol{c}(k) - (\boldsymbol{A}_2^{(it+1)} \odot \boldsymbol{A}_1^{(it+1)})\boldsymbol{d}^{(k,it+1)}\right) \to \boldsymbol{0} \tag{A.22}$$

$$\boldsymbol{A}_1^{(it+1)} - \boldsymbol{U}^{(it+1)} \to \boldsymbol{0} \tag{A.23}$$

$$\boldsymbol{A}_2^{(it+1)} - \boldsymbol{U}^{(it+1)} \to \boldsymbol{0} \tag{A.24}$$

where the terms $\alpha(\boldsymbol{U}^{(it+1)} - \boldsymbol{A}_1^{(it+1)})$ and $\beta(\boldsymbol{U}^{(it+1)} - \boldsymbol{A}_2^{(it+1)})$ have been eliminated from equations (A.19) and (A.20), respectively, by invoking equation (A.23) and (A.24). Obviously, the first five equations (2.149) to (2.153) in the KKT conditions of problem (2.129) are satisfied at any limit point:

$$\boldsymbol{Z}^* = (\boldsymbol{A}_1^*, \boldsymbol{A}_2^*, \boldsymbol{U}^*, \{\boldsymbol{D}^{(k,*)}\}, \boldsymbol{\Pi}_1^*, \boldsymbol{\Pi}_2^*) \tag{A.25}$$

Now we verify conditions (2.154) and (2.155). In (2.154) the nonnegativity of $\boldsymbol{U}^*$ is guaranteed by the algorithm construction. Hence, we only need to verify the non-positivity of $\boldsymbol{\Pi}_1^* + \boldsymbol{\Pi}_2^*$ (2.154), and the complementarity between $\boldsymbol{U}^*$ and $\boldsymbol{\Pi}_1^* + \boldsymbol{\Pi}_2^*$ (2.155). From equation (A.21), we can derive the following equality:

$$\mathrm{P}_+\left\{(\alpha \boldsymbol{A}_1^* + \beta \boldsymbol{A}_2^* + \boldsymbol{\Pi}_1^* + \boldsymbol{\Pi}_2^*)/(\alpha+\beta)\right\} = \boldsymbol{U}^* \tag{A.26}$$

From equations (A.23) and (A.24), we have $\boldsymbol{A}_1^* = \boldsymbol{A}_2^* = \boldsymbol{U}^*$. If $\boldsymbol{A}_1^* = \boldsymbol{A}_2^* = \boldsymbol{U}^* = \mathbf{0}$, then equation (A.26) reduces to $\mathrm{P}_+ \{(\boldsymbol{\Pi}_1^* + \boldsymbol{\Pi}_2^*)/(\alpha + \beta)\} = \mathbf{0}$ yielding $(\boldsymbol{\Pi}_1^* + \boldsymbol{\Pi}_2^*) \leq \mathbf{0}$ since $\alpha > 0$ and $\beta > 0$. On the other hand, if $\boldsymbol{A}_1^* = \boldsymbol{A}_2^* = \boldsymbol{U}^* > \mathbf{0}$, then equation (A.26) implies that $(\boldsymbol{\Pi}_1^* + \boldsymbol{\Pi}_2^*) = \mathbf{0}$. Therefore, the following condition can be proved:

$$(\boldsymbol{\Pi}_1^* + \boldsymbol{\Pi}_2^*) \leq \mathbf{0} \leq \boldsymbol{U}^*,\ (\boldsymbol{\Pi}_1^* + \boldsymbol{\Pi}_2^*) \boxdot \boldsymbol{U}^* = \mathbf{0} \tag{A.27}$$

Hence, proposition 2.9 holds.

Appendix B

Nonnegative compression method for SeNICA

B.1 Problem formulation

In some practical SeNICA problems defined in equation (1.50), the dimension of the observation space $\boldsymbol{x}$ must be reduced. The classical dimension compression procedure, such as the spatial prewhitening, compresses the N-dimensional vector $\boldsymbol{x}$ into a vector $\bar{\boldsymbol{x}}$ of dimension $P \ll N$. The estimate of the rank P is determined by the number of eigenvalues of the covariance matrix of $\boldsymbol{x}$ not exceedingly close to zero. The compressed observation vector $\bar{\boldsymbol{x}}$ is expressed as follows [Comon, 1994, Belouchrani et al., 1997]:

$$\bar{\boldsymbol{x}} = \boldsymbol{W}\boldsymbol{x} = (\boldsymbol{W}\boldsymbol{A})\boldsymbol{s} = \overline{\boldsymbol{A}}\boldsymbol{s} \tag{B.1}$$

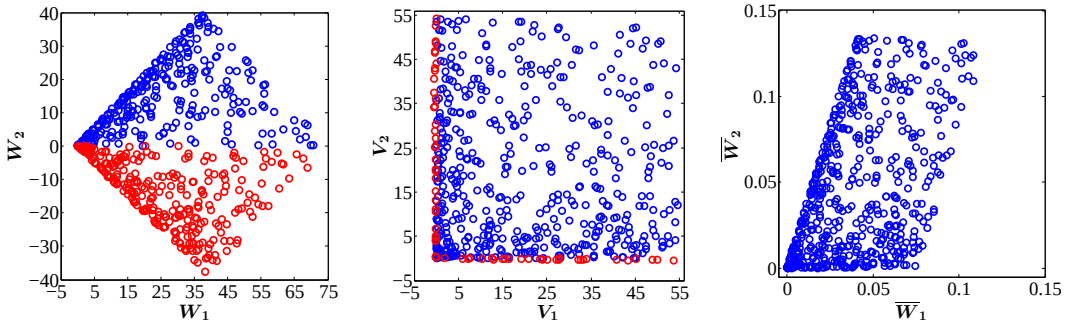
where $\boldsymbol{W} \in \mathbb{R}^{P\times N}$ is called a prewhitening matrix. In order to compute $\boldsymbol{W}$, we obtain the square root of the covariance matrix, denoted by $\boldsymbol{\Gamma} \in \mathbb{R}^{N\times P}$, such that $\boldsymbol{\Gamma}\boldsymbol{\Gamma}^{\mathsf{T}} = \mathrm{E}\{\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}\}$ assuming $\boldsymbol{x}$ being centered. Then $\boldsymbol{W} = \boldsymbol{\Gamma}^{\sharp}$ where $\sharp$ denotes the pseudo-inverse operator [Belouchrani et al., 1997]. However, the prewhitening matrix $\boldsymbol{W}$ breaks the nonnegativity property of the compressed mixing matrix $\overline{\boldsymbol{A}} = \boldsymbol{W}\boldsymbol{A}$. Thus the proposed NJDC algorithms cannot be applied to estimate $\overline{\boldsymbol{A}}$. In this section, we introduce a new Non-Negative COMPression method, namely NN-COMP, which guarantees the nonnegativity of the compressed mixing matrix $\overline{\boldsymbol{A}}$. The NN-COMP algorithm computes a linear transformation matrix $\boldsymbol{\Phi} \in \mathbb{R}^{P\times P}$ such that the resulting compression matrix $\overline{\boldsymbol{W}} = \boldsymbol{\Phi}\boldsymbol{W}$ has nonnegative components. Consequently, now the compressed observation vector $\bar{\boldsymbol{x}}$ can expressed as follows:

$$\bar{\boldsymbol{x}} = \overline{\boldsymbol{W}}\boldsymbol{x} = (\overline{\boldsymbol{W}}\boldsymbol{A})\boldsymbol{s} = \overline{\boldsymbol{A}}_{+}\boldsymbol{s} \tag{B.2}$$

where the compressed mixing matrix $\overline{\boldsymbol{A}}_{+} = \overline{\boldsymbol{W}}\boldsymbol{A} \in \mathbb{R}^{P\times P}$ is a nonnegative square matrix. Then the proposed NJDC algorithms can be used to compute the compressed matrix $\overline{\boldsymbol{A}}_{+}$ from the HO cumulant array of $\bar{\boldsymbol{x}}$. Once $\overline{\boldsymbol{A}}_{+}$ is estimated, the original matrix $\boldsymbol{A}$ is obtained as follows:

$$\boldsymbol{A} = \boldsymbol{W}^{\sharp}\boldsymbol{\Phi}^{-1}\overline{\boldsymbol{A}}_{+} = \boldsymbol{\Upsilon}\boldsymbol{\Phi}^{-1}\overline{\boldsymbol{A}}_{+} \tag{B.3}$$

It should be noted that the source $\boldsymbol{s}$ can be recovered by $\boldsymbol{s} = \overline{\boldsymbol{A}}_{+}^{-1}\bar{\boldsymbol{x}}$. Therefore generally it is not necessary to compute the original mixing matrix $\boldsymbol{A}$. Such a novel compression method can reduce the numerical complexity of the proposed NJDC algorithms.



(a) The original prewhitening matrix $\boldsymbol{W}$ (b) Step 1 of the proposed NN-COMP algorithm (c) Step 2 of the proposed NN-COMP algorithm

Figure B.1: An illustration of the proposed NN-COMP algorithm. (a) the input prewhitening matrix $\boldsymbol{W}$; (b) and (c) the proposed algorithm. (b) $\boldsymbol{V}$ is obtained by multiplying $\boldsymbol{W}$ by a Givens rotation matrix. (c) $\overline{\boldsymbol{W}}$ is obtained by multiplying $\boldsymbol{V}$ by an elementary upper triangular matrix. The blue circles denote the nonnegative values and the red circles denote the negative values.

It is particularly important for the iJDC$^{+}_{\text{LU-1}}$ and iJDC$^{+}_{\text{QR-1}}$ algorithms, since they are constrained to work on a square matrix $\boldsymbol{A}$.

Now, let's define the nonnegative compression problem as follows:

Problem B.1. *Given a prewhitening matrix $\boldsymbol{W} \in \mathbb{R}^{P\times N}$ of an N-dimensional random vector process $\boldsymbol{x}$, find a sequence of Givens rotation matrices and elementary upper triangular matrices, such that their product:*

$$\overline{\boldsymbol{W}} = \prod_{i=1}^{P}\prod_{j=i+1}^{P} \boldsymbol{U}^{(i,j)}(u_{i,j}) \underbrace{\prod_{i=1}^{P}\prod_{j=i+1}^{P} \boldsymbol{R}^{(i,j)}(\theta_{i,j})\,\boldsymbol{W}}_{\stackrel{def}{=}\boldsymbol{V}} = \boldsymbol{\Phi}\boldsymbol{W} \tag{B.4}$$

have nonnegative components. Then the transformation matrix $\boldsymbol{\Phi} \in \mathbb{R}^{P\times P}$ is defined as the product of all these Givens rotation matrices and elementary upper triangular matrices in the proper order.

Figure B.1 illustrates, in the case of $P = 2$, the process of transforming a prewhitening matrix $\boldsymbol{W}$ into the nonnegative quadrant by equation (B.4). Prewhitening makes the axes of the matrix $\boldsymbol{W}$ orthogonal to each other (figure B.1(a)). A Givens rotation matrix searches for a rotation angle that makes the outputs matrix $\boldsymbol{V}$ as nonnegative as possible. However, sometimes it still remains some negative values near the quadrant boundaries (figure B.1(b)). That is because the row vectors of $\boldsymbol{W}$ are neither well-grounded nor statistically independent [Plumbley, 2003]. We propose to use an elementary upper triangular matrix which projects the remaining negative values of $\boldsymbol{V}$ into the nonnegative quadrant (figure B.1(c)). We thus obtain a nonnegative compression matrix $\overline{\boldsymbol{W}}$ and the $(P \times P)$ compressed mixing matrix $\overline{\boldsymbol{A}}_{+} = \overline{\boldsymbol{W}}\,\boldsymbol{A}$, which preserves the nonnegativity property. Now the challenge is how to compute the Givens rotation and the elementary upper triangular matrices.

B.2 Algorithm derivation

A new two-step NN-COMP method (figure B.1) is presented in this section.

B.2.1 Step 1: estimation of the Givens rotation matrices

The Givens rotation matrix $\boldsymbol{R}^{(i,j)}(\theta_{i,j})$ transforms each pair of row vectors (i,j) of $\boldsymbol{W}$ as follows:

$$\begin{bmatrix} \boldsymbol{V}_{i,:} \\ \boldsymbol{V}_{j,:} \end{bmatrix} = \begin{bmatrix} \cos(\theta_{i,j}) & \sin(\theta_{i,j}) \\ -\sin(\theta_{i,j}) & \cos(\theta_{i,j}) \end{bmatrix} \begin{bmatrix} \boldsymbol{W}_{i,:} \\ \boldsymbol{W}_{j,:} \end{bmatrix} \tag{B.5}$$

where $\boldsymbol{V}_{i,:}$ ($\boldsymbol{V}_{j,:}$) and $\boldsymbol{W}_{i,:}$ ($\boldsymbol{W}_{j,:}$) are the i-th (j-th) row vectors of matrices $\boldsymbol{V}$ and $\boldsymbol{W}$, respectively. Now, let us consider the following negativity measure criterion defined in [Plumbley, 2003, Ouedraogo et al., 2010]:

$$\Psi(\theta_{i,j}) = \frac{1}{2}\sum_{n=1}^{N}(V_{i,n}^2 \mathbb{1}_{V_{i,n}<0} + V_{j,n}^2 \mathbb{1}_{V_{j,n}<0}) \tag{B.6}$$

where $V_{i,n}$ and $V_{j,n}$ are the (i,n)-th and (j,n)-th components of $\boldsymbol{V}$, respectively, and where $\mathbb{1}_{\alpha<0}$ is a Heaviside-step-like function defined as follows:

$$\mathbb{1}_{\alpha<0} = \begin{cases} 1, & \text{if } \alpha < 0 \\ 0, & \text{otherwise} \end{cases} \tag{B.7}$$

The purpose is to find an angle, $\theta_{i,j}$, which minimizes the total sum of squares of negative components of $\boldsymbol{V}$. The global optimum of $\Psi(\theta_{i,j})$ is difficult to obtain analytically due to the existence of the Heaviside-step-like function $\mathbb{1}_{\alpha<0}$. We propose to compute $\theta_{i,j}$ iteratively by a Newton's method. For a given iteration (it), let us consider the second order Taylor expansion, $\Psi_{\mathrm{T}}(\theta_{i,j})$, of $\Psi(\theta_{i,j})$ around $\theta_{i,j}^{(it)}$:

$$\Psi_{\mathrm{T}}(\theta_{i,j}^{(it)} + \Delta\theta) = \Psi(\theta_{i,j}^{(it)}) + \frac{\partial\Psi(\theta_{i,j}^{(it)})}{\partial\theta_{i,j}^{(it)}}\Delta\theta + \frac{\partial^2\Psi(\theta_{i,j}^{(it)})}{2\partial(\theta_{i,j}^{(it)})^2}(\Delta\theta)^2 \tag{B.8}$$

where $\theta_{i,j}^{(it)}$ is the solution at the it-th iteration, $\Delta\theta = \theta_{i,j} - \theta_{i,j}^{(it)}$, $\partial\Psi(\theta_{i,j}^{(it)})/\partial\theta_{i,j}^{(it)}$ and $\partial^2\Psi(\theta_{i,j}^{(it)})/\partial(\theta_{i,j}^{(it)})^2$ are the first and second order derivatives of the cost function (B.6) with respect to $\theta_{i,j}^{(it)}$, respectively, which are given as follows [Ouedraogo et al., 2010]:

$$\frac{\partial\Psi(\theta_{i,j}^{(it)})}{\partial\theta_{i,j}^{(it)}} = \sum_{n=1}^{N} V_{i,n}^{(it)} V_{j,n}^{(it)} \left(\mathbb{1}_{V_{i,n}^{(it)}<0} \mathbb{1}_{V_{j,n}^{(it)}>0} - \mathbb{1}_{V_{i,n}^{(it)}>0} \mathbb{1}_{V_{j,n}^{(it)}<0} \right) \tag{B.9}$$

$$\frac{\partial^2\Psi(\theta_{i,j}^{(it)})}{\partial(\theta_{i,j}^{(it)})^2} = \sum_{n=1}^{N} \left((V_{j,n}^{(it)})^2 - (V_{i,n}^{(it)})^2 \right) \left(\mathbb{1}_{V_{i,n}^{(it)}<0} \mathbb{1}_{V_{j,n}^{(it)}>0} - \mathbb{1}_{V_{i,n}^{(it)}>0} \mathbb{1}_{V_{j,n}^{(it)}<0} \right) \tag{B.10}$$

and where

$$\begin{bmatrix} V_{i,n}^{(it)} \\ V_{j,n}^{(it)} \end{bmatrix} = \begin{bmatrix} \cos(\theta_{i,j}^{(it)}) & \sin(\theta_{i,j}^{(it)}) \\ -\sin(\theta_{i,j}^{(it)}) & \cos(\theta_{i,j}^{(it)}) \end{bmatrix} \begin{bmatrix} W_{i,n} \\ W_{j,n} \end{bmatrix} \tag{B.11}$$

Then the minimum of (B.8) with respect to $\Delta\theta$ can be reached at:

$$\Delta\theta = -\left(\frac{\partial^2\Psi(\theta_{i,j}^{(it)})}{\partial(\theta_{i,j}^{(it)})^2}\right)^{-1}\frac{\partial\Psi(\theta_{i,j}^{(it)})}{\partial\theta_{i,j}^{(it)}} \tag{B.12}$$

$\partial^2\Psi(\theta_{i,j}^{(it)})/\partial(\theta_{i,j}^{(it)})^2 = 0$ means that all the elements of the row vectors $\boldsymbol{V}_{i,:}^{(it)}$ and $\boldsymbol{V}_{j,:}^{(it)}$ have the same sign. In this situation it is not necessary to perform the optimization since it will not decrease the criterion [Ouedraogo et al., 2010]. The numerical complexity of estimating one $\theta_{i,j}$ costs $(9N+1)J_{\theta_{i,j}}^{(it)}$, where $J_{\theta_{i,j}}^{(it)}$ denotes the number of consumed iterations. The process of all the rotation angles $\theta_{i,j}$ is called a rotation sweep. The rotation sweep can be stopped when the relative decrease of the total sum of squares of negative components of $\boldsymbol{V}$ between two sweeps falls below a fixed small threshold.

B.2.2 Step 2: estimation of the elementary upper triangular matrices

After the rotation step, the elementary upper triangular matrix $\boldsymbol{U}^{(i,j)}(u_{i,j})$ transforms each pair of row vectors (i,j) of $\boldsymbol{V}$ as follows:

$$\begin{bmatrix} \overline{\boldsymbol{W}}_{j,:} \\ \overline{\boldsymbol{W}}_{i,:} \end{bmatrix} = \begin{bmatrix} 1 & u_{i,j} \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \boldsymbol{V}_{j,:} \\ \boldsymbol{V}_{i,:} \end{bmatrix} \tag{B.13}$$

where $\overline{\boldsymbol{W}}_{i,:}$ and $\overline{\boldsymbol{W}}_{j,:}$ are the i-th and j-th row vectors of matrix $\overline{\boldsymbol{W}}$ respectively, and where $u_{i,j}$ is also called a shearing factor. From equation (B.13), the vector $\overline{\boldsymbol{W}}_{i,:}$ remains unchanged. The negativity measure criterion is then defined as follows:

$$\Psi(u_{i,j}) = \frac{1}{2}\sum_{n=1}^{N}\overline{W}_{j,n}^2\mathbb{1}_{\overline{W}_{j,n}<0} \tag{B.14}$$

Similarly, the second order Taylor expansion $\Psi_{\mathrm{T}}(u_{i,j})$ of $\Psi(u_{i,j})$ around $u_{i,j}^{(it)}$ can be expressed as follows:

$$\Psi_{\mathrm{T}}(u_{i,j}^{(it)}+\Delta u) = \Psi(u_{i,j}^{(it)}) + \frac{\partial\Psi(u_{i,j}^{(it)})}{\partial u_{i,j}^{(it)}}\Delta u + \frac{\partial^2\Psi(u_{i,j}^{(it)})}{2\partial(u_{i,j}^{(it)})^2}(\Delta u)^2 \tag{B.15}$$

where $u_{i,j}^{(it)}$ is the solution at the it-th iteration, $\Delta u = u_{i,j} - u_{i,j}^{(it)}$, $\partial\Psi(u_{i,j}^{(it)})/\partial u_{i,j}^{(it)}$ and $\partial^2\Psi(u_{i,j}^{(it)})/\partial(u_{i,j}^{(it)})^2$ are the first and second order derivatives of the cost function (B.14) with respect to $u_{i,j}^{(it)}$, respectively, which are given as follows:

$$\frac{\partial\Psi(u_{i,j}^{(it)})}{\partial u_{i,j}^{(it)}} = \sum_{n=1}^{N}\overline{W}_{i,n}^{(it)}\overline{W}_{j,n}^{(it)}\mathbb{1}_{\overline{W}_{j,n}^{(it)}<0} \tag{B.16}$$

$$\frac{\partial^2\Psi(u_{i,j}^{(it)})}{\partial(u_{i,j}^{(it)})^2} = \sum_{n=1}^{N}(\overline{W}_{i,n}^{(it)})^2\mathbb{1}_{\overline{W}_{j,n}^{(it)}<0} \tag{B.17}$$

and where

$$\begin{bmatrix} \overline{W}_{j,n}^{(it)} \\ \overline{W}_{i,n}^{(it)} \end{bmatrix} = \begin{bmatrix} 1 & u_{i,j}^{(it)} \\ 0 & 1 \end{bmatrix}\begin{bmatrix} V_{j,n} \\ V_{i,n} \end{bmatrix} \tag{B.18}$$

Then the minimum of (B.15) with respect to Δu can be reached at:

$$\Delta u = -\left(\frac{\partial^2 \Psi(u_{i,j}^{(it)})}{\partial (u_{i,j}^{(it)})^2}\right)^{-1} \frac{\partial \Psi(u_{i,j}^{(it)})}{\partial u_{i,j}^{(it)}} \tag{B.19}$$

When $\partial^2 \Psi(u_{i,j}^{(it)})/\partial (u_{i,j}^{(it)})^2 = 0$, it means that all the elements of the row vector $\overline{\boldsymbol{W}}_{j,:}^{(it)}$ have the same sign and that equation (B.19) does not need to be computed. The numerical complexity of estimating one $u_{i,j}$ costs $(4N+1)J_{u_{i,j}}^{(it)}$, where $J_{u_{i,j}}^{(it)}$ denotes the number of used iterations. The process of all the shearing factors $u_{i,j}$ is called a shearing sweep. The shearing sweep can be stopped when the relative decrease of the total sum of squares of negative components of $\overline{\boldsymbol{W}}$ between two sweeps falls below a predefined positive threshold.

In practice, the new NN-COMP algorithm contains some rotation sweeps followed by several shearing sweeps, in order to ensure the convergence. The overall numerical complexity of NN-COMP is given as follows:

$$\begin{aligned} &\min(N^2M/2 + 4N^3/3 + NPM, 4N^2M + 8N^3) \\ &\quad + \left(\sum_{i=1}^{P}\sum_{j=i+1}^{P}(9N+1)J_{\theta_{i,j}}^{(it)}\right) J_{\text{Givens}}^{(sweep)} + \left(\sum_{i=1}^{P}\sum_{j=i+1}^{P}(4N+1)J_{u_{i,j}}^{(it)}\right) J_{\text{Shearing}}^{(sweep)} \end{aligned} \tag{B.20}$$

where N, P and M are the numbers of the observations, of the sources and of the sample points, respectively, where $J_{\theta_{i,j}}^{(it)}$ and $J_{u_{i,j}}^{(it)}$ are the numbers of inner iterations for estimating one $\theta_{i,j}$ and one $u_{i,j}$, respectively, and where $J_{\text{Givens}}^{(sweep)}$ and $J_{\text{Shearing}}^{(sweep)}$ are the numbers of used sweeps for estimating all the Givens rotation matrices and all the shearing transformation matrices, respectively. The first term in equation (B.20) corresponds to the complexity of computing the classical prewhitening matrix. The pseudo-code of the NN-COMP algorithm is provided in appendix C.8.

Appendix C

Pseudo-codes of algorithms

C.1 Pseudo-code of the cyclic Jacobi-like iteration

Algorithm 1 Cyclic Jacobi-like iteration for NJDC

procedure $\boldsymbol{B}^{\text{OPT}} = \text{argmin}_{\boldsymbol{B}}\, \Psi(\boldsymbol{B})$
  Initialization: $\boldsymbol{B}^{(0)} = \boldsymbol{I}$, $it = 0$
  while $\Psi(\boldsymbol{B}^{(it)}) > \epsilon$ **do**
    for $i = 1$ to N **do**
      for $j = 1$ to N and $j \neq i$ **do**
        $it = it + 1$
        Compute $\boldsymbol{\Theta}^{(i,j)}(\xi_{i,j}))$ by equation (2.12)
        Update $\boldsymbol{B}^{(it-1)}$ by equation (2.13)
      end for
    end for
  end while
  Return: $\boldsymbol{B}^{\text{opt}} = \boldsymbol{B}^{(it)}$
end procedure

C.2 Pseudo-code of the $\text{JDC}^{+}_{\text{LU-3}}$ algorithm

Algorithm 2 $\text{JDC}^{+}_{\text{LU-3}}$ algorithm for NJDC

procedure $[\boldsymbol{B}^{\text{OPT}}, \{\boldsymbol{D}^{(k,\text{OPT})}\}] = \text{argmin}_{\boldsymbol{B},\{\boldsymbol{D}^{(k)}\}} \Psi_3^+(\boldsymbol{B}, \{\boldsymbol{D}^{(k)}\})$
  Initialization: $\widehat{\boldsymbol{B}} = \boldsymbol{B}^{(0)}$, $\widehat{\boldsymbol{A}} = \widehat{\boldsymbol{B}}^{\boxdot 2}$, $\widehat{\boldsymbol{D}}^{(k)} = \widehat{\boldsymbol{D}}^{(k,0)}$ and $it = 0$
  while $\Psi_3^+(\widehat{\boldsymbol{B}}, \{\widehat{\boldsymbol{D}}^{(k)}\}) > \epsilon$ **do**
    $it = it + 1$
    ▷ The AC phase
    for $j = 1$ to N **do** ▷ Estimate $\boldsymbol{L}$
      for $i = j + 1$ to N **do**
        Compute $\ell_{i,j}$ by minimizing (2.44)
        $\varsigma = \text{sign}(\widehat{\boldsymbol{a}}_j + \ell_{i,j}\widehat{\boldsymbol{a}}_i)$
        if $|(\widehat{\boldsymbol{a}}_j + \ell_{i,j}\widehat{\boldsymbol{a}}_i)| == \varsigma(\widehat{\boldsymbol{a}}_j + \ell_{i,j}\widehat{\boldsymbol{a}}_i)$ **then**
          $\widehat{\boldsymbol{a}}_j^{(\text{new})} = \varsigma(\widehat{\boldsymbol{a}}_j + \ell_{i,j}\widehat{\boldsymbol{a}}_i)$, $\widehat{\boldsymbol{b}}_j^{(\text{new})} = \sqrt{\widehat{\boldsymbol{a}}_j^{(\text{new})}}$
        else
          Compute $\ell_{i,j}$ by minimizing (2.36)
          Update $\widehat{\boldsymbol{B}}^{(\text{new})}$ by (2.27), $\widehat{\boldsymbol{A}}^{(\text{new})} = (\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}$
        end if
        $\widehat{\boldsymbol{B}} = \widehat{\boldsymbol{B}}^{(\text{new})}$, $\widehat{\boldsymbol{A}} = \widehat{\boldsymbol{A}}^{(\text{new})}$,
      end for
    end for
    for $i = 1$ to N **do** ▷ Estimate $\boldsymbol{U}$
      for $j = i + 1$ to N **do**
        Compute $u_{i,j}$ by minimizing (2.44)
        $\varsigma = \text{sign}(\widehat{\boldsymbol{a}}_j + u_{i,j}\widehat{\boldsymbol{a}}_i)$
        if $|(\widehat{\boldsymbol{a}}_j + u_{i,j}\widehat{\boldsymbol{a}}_i)| == \varsigma(\widehat{\boldsymbol{a}}_j + u_{i,j}\widehat{\boldsymbol{a}}_i)$ **then**
          $\widehat{\boldsymbol{a}}_j^{(\text{new})} = \varsigma(\widehat{\boldsymbol{a}}_j + u_{i,j}\widehat{\boldsymbol{a}}_i)$, $\widehat{\boldsymbol{b}}_j^{(\text{new})} = \sqrt{\widehat{\boldsymbol{a}}_j^{(\text{new})}}$
        else
          Compute $u_{i,j}$ by minimizing (2.36)
          Update $\widehat{\boldsymbol{B}}^{(\text{new})}$ by (2.27), $\widehat{\boldsymbol{A}}^{(\text{new})} = (\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}$
        end if
        $\widehat{\boldsymbol{B}} = \widehat{\boldsymbol{B}}^{(\text{new})}$, $\widehat{\boldsymbol{A}} = \widehat{\boldsymbol{A}}^{(\text{new})}$,
      end for
    end for
    ▷ The DC phase
    for $k = 1$ to K **do** ▷ Estimate $\{\boldsymbol{D}^{(k)}\}$
      Compute $\widehat{\boldsymbol{D}}^{(k,\text{new})}$ by (2.41)
    end for
    $\{\widehat{\boldsymbol{D}}^{(k)}\} = \{\widehat{\boldsymbol{D}}^{(k,\text{new})}\}$
  end while
  Return: $\boldsymbol{B}^{\text{opt}} = \widehat{\boldsymbol{B}}^{(\text{new})}$, $\{\widehat{\boldsymbol{D}}^{(k),\text{opt}}\} = \{\widehat{\boldsymbol{D}}^{(k,\text{new})}\}$
end procedure

C.3 Pseudo-code of the iJDC$^{+}_{\text{LU-1}}$ algorithm

Algorithm 3 iJDC$^{+}_{\text{LU-1}}$ algorithm for NJDC

procedure $[\boldsymbol{B}^{\text{OPT}}] = \text{argmin}_{\boldsymbol{B}}\, \Psi^{+}_{1'}(\boldsymbol{B})$
  Initialization: $\widehat{\boldsymbol{B}} = \boldsymbol{B}^{(0)}$, $\widehat{\boldsymbol{A}} = \widehat{\boldsymbol{B}}^{\boxdot 2}$, $\boldsymbol{C}^{(k,-1)} = (\boldsymbol{C}^{(k)})^{-1}$ and $it = 0$
  while $\Psi^{+}_{1'}(\widehat{\boldsymbol{B}}) > \epsilon$ **do**
    $it = it + 1$
    for $j = 1$ to N **do** ▷ Estimate $\boldsymbol{L}$
      for $i = j + 1$ to N **do**
        Compute $\ell_{i,j}$ by (2.72)
        $\varsigma = \text{sign}(\widehat{\boldsymbol{a}}_j + \ell_{i,j}\widehat{\boldsymbol{a}}_i)$
        if $|(\widehat{\boldsymbol{a}}_j + \ell_{i,j}\widehat{\boldsymbol{a}}_i)| == \varsigma(\widehat{\boldsymbol{a}}_j + \ell_{i,j}\widehat{\boldsymbol{a}}_i)$ **then**
          $\widehat{\boldsymbol{a}}_j^{(\text{new})} = \varsigma(\widehat{\boldsymbol{a}}_j + \ell_{i,j}\widehat{\boldsymbol{a}}_i)$, $\widehat{\boldsymbol{b}}_j^{(\text{new})} = \sqrt{\widehat{\boldsymbol{a}}_j^{(\text{new})}}$
        else
          Compute $\ell_{i,j}$ by minimizing (2.53)
          Update $\widehat{\boldsymbol{B}}^{(\text{new})}$ by (2.27), $\widehat{\boldsymbol{A}}^{(\text{new})} = (\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}$
        end if
        Update $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ by (2.51)
        $\widehat{\boldsymbol{B}} = \widehat{\boldsymbol{B}}^{(\text{new})}$, $\widehat{\boldsymbol{A}} = \widehat{\boldsymbol{A}}^{(\text{new})}$, $\widehat{\boldsymbol{C}}^{(k)} = \widehat{\boldsymbol{C}}^{(k,\text{new})}$
      end for
    end for
    for $i = 1$ to N **do** ▷ Estimate $\boldsymbol{U}$
      for $j = i + 1$ to N **do**
        Compute $u_{i,j}$ by (2.72)
        $\varsigma = \text{sign}(\widehat{\boldsymbol{a}}_j + u_{i,j}\widehat{\boldsymbol{a}}_i)$
        if $|(\widehat{\boldsymbol{a}}_j + u_{i,j}\widehat{\boldsymbol{a}}_i)| == \varsigma(\widehat{\boldsymbol{a}}_j + u_{i,j}\widehat{\boldsymbol{a}}_i)$ **then**
          $\widehat{\boldsymbol{a}}_j^{(\text{new})} = \varsigma(\widehat{\boldsymbol{a}}_j + u_{i,j}\widehat{\boldsymbol{a}}_i)$, $\widehat{\boldsymbol{b}}_j^{(\text{new})} = \sqrt{\widehat{\boldsymbol{a}}_j^{(\text{new})}}$
        else
          Compute $u_{i,j}$ by minimizing (2.53)
          Update: $\widehat{\boldsymbol{B}}^{(\text{new})}$ by (2.27), $\widehat{\boldsymbol{A}}^{(\text{new})} = (\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}$
        end if
        Update $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ by (2.51)
        $\widehat{\boldsymbol{B}} = \widehat{\boldsymbol{B}}^{(\text{new})}$, $\widehat{\boldsymbol{A}} = \widehat{\boldsymbol{A}}^{(\text{new})}$, $\widehat{\boldsymbol{C}}^{(k)} = \widehat{\boldsymbol{C}}^{(k,\text{new})}$
      end for
    end for
    if $(it \mod q) == 0$ **then** ▷ Row balancing of $\widehat{\boldsymbol{C}}^{(k)}$
      Perform row balancing scheme
    end if
  end while
  Return: $\boldsymbol{B}^{\text{opt}} = \widehat{\boldsymbol{B}}^{(\text{new})}$
end procedure

C.4 Pseudo-code of the iJDC$^{+}_{\text{QR-1}}$ algorithm

Algorithm 4 iJDC$^{+}_{\text{QR-1}}$ algorithm for NJDC

procedure $[\boldsymbol{B}^{\text{OPT}}] = \text{argmin}_{\boldsymbol{B}}\, \Psi^{+}_{1'}(\boldsymbol{B})$
  Initialization: $\widehat{\boldsymbol{B}} = \boldsymbol{B}^{(0)}$, $\widehat{\boldsymbol{A}} = \widehat{\boldsymbol{B}}^{\boxdot 2}$, $\boldsymbol{C}^{(k,-1)} = (\boldsymbol{C}^{(k)})^{-1}$ and $it = 0$
  while $\Psi^{+}_{1'}(\widehat{\boldsymbol{B}}) > \epsilon$ **do**
    $it = it + 1$
    for $i = 1$ to N **do** ▷ Estimate $\boldsymbol{Q}$
      for $j = i + 1$ to N **do**
        Compute $\theta_{i,j}$ by minimizing (2.73)
        $\varsigma_i = \text{sign}(\cos(\theta_{i,j})\widehat{\boldsymbol{a}}_i - \sin(\theta_{i,j})\widehat{\boldsymbol{a}}_j)$, $\varsigma_j = \text{sign}(\sin(\theta_{i,j})\widehat{\boldsymbol{a}}_i + \cos(\theta_{i,j})\widehat{\boldsymbol{a}}_j)$
        if $|(\cos(\theta_{i,j})\widehat{\boldsymbol{a}}_i - \sin(\theta_{i,j})\widehat{\boldsymbol{a}}_j)| == \varsigma_i(\cos(\theta_{i,j})\widehat{\boldsymbol{a}}_i - \sin(\theta_{i,j})\widehat{\boldsymbol{a}}_j)$ and
          $|(\sin(\theta_{i,j})\widehat{\boldsymbol{a}}_i + \cos(\theta_{i,j})\widehat{\boldsymbol{a}}_j)| == \varsigma_j(\sin(\theta_{i,j})\widehat{\boldsymbol{a}}_i + \cos(\theta_{i,j})\widehat{\boldsymbol{a}}_j)$ **then**
          $\widehat{\boldsymbol{a}}_i^{(\text{new})} = \varsigma_i(\cos(\theta_{i,j})\widehat{\boldsymbol{a}}_i - \sin(\theta_{i,j})\widehat{\boldsymbol{a}}_j)$, $\widehat{\boldsymbol{b}}_i^{(\text{new})} = \sqrt{\widehat{\boldsymbol{a}}_i^{(\text{new})}}$
          $\widehat{\boldsymbol{a}}_j^{(\text{new})} = \varsigma_j(\sin(\theta_{i,j})\widehat{\boldsymbol{a}}_i + \cos(\theta_{i,j})\widehat{\boldsymbol{a}}_j)$, $\widehat{\boldsymbol{b}}_j^{(\text{new})} = \sqrt{\widehat{\boldsymbol{a}}_j^{(\text{new})}}$
        else
          Compute $\theta_{i,j}$ by minimizing (2.65)
          Update: $\widehat{\boldsymbol{B}}^{(\text{new})}$ by (2.57), $\widehat{\boldsymbol{A}}^{(\text{new})} = (\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}$
        end if
        Update $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ by (2.61)
        $\widehat{\boldsymbol{B}} = \widehat{\boldsymbol{B}}^{(\text{new})}$, $\widehat{\boldsymbol{A}} = \widehat{\boldsymbol{A}}^{(\text{new})}$, $\widehat{\boldsymbol{C}}^{(k)} = \widehat{\boldsymbol{C}}^{(k,\text{new})}$
      end for
    end for
    for $i = 1$ to N **do** ▷ Estimate $\boldsymbol{R}$
      for $j = i + 1$ to N **do**
        Compute $u_{i,j}$ by (2.72)
        $\varsigma = \text{sign}(\widehat{\boldsymbol{a}}_j + u_{i,j}\widehat{\boldsymbol{a}}_i)$
        if $|(\widehat{\boldsymbol{a}}_j + u_{i,j}\widehat{\boldsymbol{a}}_i)| == \varsigma(\widehat{\boldsymbol{a}}_j + u_{i,j}\widehat{\boldsymbol{a}}_i)$ **then**
          $\widehat{\boldsymbol{a}}_j^{(\text{new})} = \varsigma(\widehat{\boldsymbol{a}}_j + u_{i,j}\widehat{\boldsymbol{a}}_i)$, $\widehat{\boldsymbol{b}}_j^{(\text{new})} = \sqrt{\widehat{\boldsymbol{a}}_j^{(\text{new})}}$
        else
          Compute $u_{i,j}$ by minimizing (2.53)
          Update: $\widehat{\boldsymbol{B}}^{(\text{new})}$ by (2.27), $\widehat{\boldsymbol{A}}^{(\text{new})} = (\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}$
        end if
        Update $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ by (2.51)
        $\widehat{\boldsymbol{B}} = \widehat{\boldsymbol{B}}^{(\text{new})}$, $\widehat{\boldsymbol{A}} = \widehat{\boldsymbol{A}}^{(\text{new})}$, $\widehat{\boldsymbol{C}}^{(k)} = \widehat{\boldsymbol{C}}^{(k,\text{new})}$
      end for
    end for
    if $(it \mod q) == 0$ **then** ▷ Row balancing of $\widehat{\boldsymbol{C}}^{(k)}$
      Perform row balancing scheme
    end if
  end while
  Return: $\boldsymbol{B}^{\text{opt}} = \widehat{\boldsymbol{B}}^{(\text{new})}$
end procedure

C.5 Pseudo-code of the $\text{JDC}^{+}_{\text{LU-1}}$ algorithm

Algorithm 5 $\text{JDC}^{+}_{\text{LU-1}}$ algorithm for NJDC

procedure $[\boldsymbol{B}^{\text{OPT}}] = \text{argmin}_{\boldsymbol{B}} \Psi_1^{+}(\boldsymbol{B})$
  Initialization: $\widehat{\boldsymbol{B}} = \boldsymbol{B}^{(0)}$, $\widehat{\boldsymbol{A}} = \widehat{\boldsymbol{B}}^{\boxdot 2}$, and $it = 0$
  while $\Psi_1^{+}(\widehat{\boldsymbol{B}}) > \epsilon$ **do**
    $it = it + 1$
    for $j = 1$ to N **do** ▷ Estimate $\boldsymbol{L}$
      for $i = j + 1$ to N **do**
        Compute $\ell_{i,j}$ by (2.126)
        $\varsigma = \text{sign}(\widehat{\boldsymbol{a}}_j + \ell_{i,j}\widehat{\boldsymbol{a}}_i)$
        if $|(\widehat{\boldsymbol{a}}_j + \ell_{i,j}\widehat{\boldsymbol{a}}_i)| == \varsigma(\widehat{\boldsymbol{a}}_j + \ell_{i,j}\widehat{\boldsymbol{a}}_i)$ **then**
          $\widehat{\boldsymbol{a}}_j^{(\text{new})} = \varsigma(\widehat{\boldsymbol{a}}_j + \ell_{i,j}\widehat{\boldsymbol{a}}_i)$, $\widehat{\boldsymbol{b}}_j^{(\text{new})} = \sqrt{\widehat{\boldsymbol{a}}_j^{(\text{new})}}$
        else
          Compute $\ell_{i,j}$ by minimizing (2.98)
          Update $\widehat{\boldsymbol{B}}^{(\text{new})}$ by (2.27), $\widehat{\boldsymbol{A}}^{(\text{new})} = (\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}$
        end if
        Update $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ by (2.86)
        $\widehat{\boldsymbol{B}} = \widehat{\boldsymbol{B}}^{(\text{new})}$, $\widehat{\boldsymbol{A}} = \widehat{\boldsymbol{A}}^{(\text{new})}$, $\widehat{\boldsymbol{C}}^{(k)} = \widehat{\boldsymbol{C}}^{(k,\text{new})}$
      end for
    end for
    for $i = 1$ to N **do** ▷ Estimate $\boldsymbol{U}$
      for $j = i + 1$ to N **do**
        Compute $u_{i,j}$ by (2.126)
        $\varsigma = \text{sign}(\widehat{\boldsymbol{a}}_j + u_{i,j}\widehat{\boldsymbol{a}}_i)$
        if $|(\widehat{\boldsymbol{a}}_j + u_{i,j}\widehat{\boldsymbol{a}}_i)| == \varsigma(\widehat{\boldsymbol{a}}_j + u_{i,j}\widehat{\boldsymbol{a}}_i)$ **then**
          $\widehat{\boldsymbol{a}}_j^{(\text{new})} = \varsigma(\widehat{\boldsymbol{a}}_j + u_{i,j}\widehat{\boldsymbol{a}}_i)$, $\widehat{\boldsymbol{b}}_j^{(\text{new})} = \sqrt{\widehat{\boldsymbol{a}}_j^{(\text{new})}}$
        else
          Compute $u_{i,j}$ by minimizing (2.98)
          Update: $\widehat{\boldsymbol{B}}^{(\text{new})}$ by (2.27), $\widehat{\boldsymbol{A}}^{(\text{new})} = (\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}$
        end if
        Update $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ by (2.86)
        $\widehat{\boldsymbol{B}} = \widehat{\boldsymbol{B}}^{(\text{new})}$, $\widehat{\boldsymbol{A}} = \widehat{\boldsymbol{A}}^{(\text{new})}$, $\widehat{\boldsymbol{C}}^{(k)} = \widehat{\boldsymbol{C}}^{(k,\text{new})}$
      end for
    end for
    if $(it \mod q) == 0$ **then** ▷ Row balancing of $\widehat{\boldsymbol{C}}^{(k)}$
      Perform row balancing scheme
    end if
  end while
  Return: $\boldsymbol{B}^{\text{opt}} = \widehat{\boldsymbol{B}}^{(\text{new})}$
end procedure

C.6 Pseudo-code of the $\text{JDC}^{+}_{\text{LU-2}}$ algorithm

Algorithm 6 $\text{JDC}^{+}_{\text{LU-2}}$ algorithm for NJDC

procedure $[\boldsymbol{B}^{\text{OPT}}] = \text{argmin}_{\boldsymbol{B}} \Psi_2^+(\boldsymbol{B})$
  Initialization: $\widehat{\boldsymbol{B}} = \boldsymbol{B}^{(0)}$, $\widehat{\boldsymbol{A}} = \widehat{\boldsymbol{B}}^{\boxdot 2}$, and $it = 0$
  while $\Psi_2^+(\widehat{\boldsymbol{B}}) > \epsilon$ **do**
    $it = it + 1$
    for $j = 1$ to N **do** ▷ Estimate $\boldsymbol{L}$
      for $i = j + 1$ to N **do**
        Compute $\ell_{i,j}$ by minimizing (2.127)
        $\varsigma = \text{sign}(\widehat{\boldsymbol{a}}_j + \ell_{i,j}\widehat{\boldsymbol{a}}_i)$
        if $|(\widehat{\boldsymbol{a}}_j + \ell_{i,j}\widehat{\boldsymbol{a}}_i)| == \varsigma(\widehat{\boldsymbol{a}}_j + \ell_{i,j}\widehat{\boldsymbol{a}}_i)$ **then**
          $\widehat{\boldsymbol{a}}_j^{(\text{new})} = \varsigma(\widehat{\boldsymbol{a}}_j + \ell_{i,j}\widehat{\boldsymbol{a}}_i)$, $\widehat{\boldsymbol{b}}_j^{(\text{new})} = \sqrt{\widehat{\boldsymbol{a}}_j^{(\text{new})}}$
        else
          Compute $\ell_{i,j}$ by minimizing (2.115)
          Update $\widehat{\boldsymbol{B}}^{(\text{new})}$ by (2.27), $\widehat{\boldsymbol{A}}^{(\text{new})} = (\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}$
        end if
        Update $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ by (2.86)
        $\widehat{\boldsymbol{B}} = \widehat{\boldsymbol{B}}^{(\text{new})}$, $\widehat{\boldsymbol{A}} = \widehat{\boldsymbol{A}}^{(\text{new})}$, $\widehat{\boldsymbol{C}}^{(k)} = \widehat{\boldsymbol{C}}^{(k,\text{new})}$
      end for
    end for
    for $i = 1$ to N **do** ▷ Estimate $\boldsymbol{U}$
      for $j = i + 1$ to N **do**
        Compute $u_{i,j}$ by minimizing (2.127)
        $\varsigma = \text{sign}(\widehat{\boldsymbol{a}}_j + u_{i,j}\widehat{\boldsymbol{a}}_i)$
        if $|(\widehat{\boldsymbol{a}}_j + u_{i,j}\widehat{\boldsymbol{a}}_i)| == \varsigma(\widehat{\boldsymbol{a}}_j + u_{i,j}\widehat{\boldsymbol{a}}_i)$ **then**
          $\widehat{\boldsymbol{a}}_j^{(\text{new})} = \varsigma(\widehat{\boldsymbol{a}}_j + u_{i,j}\widehat{\boldsymbol{a}}_i)$, $\widehat{\boldsymbol{b}}_j^{(\text{new})} = \sqrt{\widehat{\boldsymbol{a}}_j^{(\text{new})}}$
        else
          Compute $u_{i,j}$ by minimizing (2.115)
          Update: $\widehat{\boldsymbol{B}}^{(\text{new})}$ by (2.27), $\widehat{\boldsymbol{A}}^{(\text{new})} = (\widehat{\boldsymbol{B}}^{(\text{new})})^{\boxdot 2}$
        end if
        Update $\widehat{\boldsymbol{C}}^{(k,\text{new})}$ by (2.86)
        $\widehat{\boldsymbol{B}} = \widehat{\boldsymbol{B}}^{(\text{new})}$, $\widehat{\boldsymbol{A}} = \widehat{\boldsymbol{A}}^{(\text{new})}$, $\widehat{\boldsymbol{C}}^{(k)} = \widehat{\boldsymbol{C}}^{(k,\text{new})}$
      end for
    end for
    if $(it \mod q) == 0$ **then** ▷ Row balancing of $\widehat{\boldsymbol{C}}^{(k)}$
      Perform row balancing scheme
    end if
  end while
  Return: $\boldsymbol{B}^{\text{opt}} = \widehat{\boldsymbol{B}}^{(\text{new})}$
end procedure

C.7 Pseudo-code of the $\text{JDC}^{+}_{\text{ADMM-3}}$ algorithm

Algorithm 7 $\text{JDC}^{+}_{\text{ADMM-3}}$ algorithm for NJDC

procedure $[\boldsymbol{A}_1^{\text{OPT}}, \boldsymbol{A}_2^{\text{OPT}}, \boldsymbol{U}^{\text{OPT}}, \{\boldsymbol{D}^{(k,\text{OPT})}\}] = \text{argmin}_{\boldsymbol{A}_1,\boldsymbol{A}_2,\boldsymbol{U},\{\boldsymbol{D}^{(k)}\}} \Psi_3^{+}(\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{U}, \{\boldsymbol{D}^{(k)}\})$
  Initialization: $\boldsymbol{A}_1^{(0)}$, $\boldsymbol{A}_2^{(0)}$, $\boldsymbol{U}^{(0)}$, $\{\boldsymbol{D}^{(k,0)}\}$, $\boldsymbol{\Pi}_1^{(0)}$, $\boldsymbol{\Pi}_2^{(0)}$ and $it = 0$
  while $\Psi_3^{+}(\boldsymbol{A}_1^{(it)}, \boldsymbol{A}_2^{(it)}, \boldsymbol{U}^{(it)}, \{\boldsymbol{D}^{(k,it)}\}) > \epsilon$ **do**
    $it = it + 1$
    Compute $\boldsymbol{A}_1^{(it+1)}$ by equation (2.136)
    Compute $\boldsymbol{A}_2^{(it+1)}$ by equation (2.137)
    Compute $\boldsymbol{U}^{(it+1)}$ by equation (2.138)
    Compute each $\boldsymbol{D}^{(k,it+1)}$ by equation (2.139)
    Compute $\boldsymbol{\Pi}_1^{(it+1)}$ by equation (2.140)
    Compute $\boldsymbol{\Pi}_2^{(it+1)}$ by equation (2.141)
  end while
  Return: $\boldsymbol{A}_1^{\text{opt}} = \boldsymbol{A}_1^{(it)}$, $\boldsymbol{A}_2^{\text{opt}} = \boldsymbol{A}_2^{(it)}$, $\boldsymbol{U}^{\text{opt}} = \boldsymbol{U}^{(it)}$, $\{\boldsymbol{D}^{(k,\text{opt})}\} = \{\boldsymbol{D}^{(k,it)}\}$
end procedure

C.8 Pseudo-code of the NN-COMP algorithm

Algorithm 8 Nonnegative compression algorithm for SeNICA

procedure COMPUTE A NONNEGATIVE COMPRESSION MATRIX $\overline{\boldsymbol{W}}$ FROM A REALIZATION OF AN OBSERVATION VECTOR $\{\boldsymbol{x}\}$
  Initialization: compute the prewhitening matrix $\boldsymbol{\Gamma}$ such that $\boldsymbol{\Gamma}\boldsymbol{\Gamma}^{\mathsf{T}} = \mathrm{E}\{\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}\}$ by EVD;
  Define a criterion: $\Psi_{\mathrm{neg}}(\boldsymbol{V}) = \frac{1}{2}\sum_{p=1}^{P}\sum_{n=1}^{N} V_{p,n}^2 \mathbb{1}_{V_{p,n}<0}$
  Sweep counter: $swp_r = 0$
  while $\Psi_{\mathrm{neg}}(\boldsymbol{V})$ can still be minimized **do**
    $swp_r = swp_r + 1$
    for $i = 1$ to N **do** ▷ Estimate the Given rotation matrices
      for $j = i + 1$ to N **do** $it = 0$
        while criterion (B.6) can still be minimized **do**
          $it = it + 1$
          Compute $\theta_{i,j}^{(it)}$ by equation (B.12)
          Update $\boldsymbol{V}_{i,:}^{(it)}$ and $\boldsymbol{V}_{j,:}^{(it)}$ by equation (B.11)
        end while
      end for
    end for
  end while
  Sweep counter: $swp_u = 0$
  while $\Psi_{\mathrm{neg}}(\overline{\boldsymbol{W}})$ can still be minimized **do**
    $swp_u = swp_u + 1$
    for $i = 1$ to N **do** ▷ Estimate the elementary upper triangular matrices
      for $j = i + 1$ to N **do** $it = 0$
        while criterion (B.14) can still be minimized **do**
          $it = it + 1$
          Compute $u_{i,j}^{(it)}$ by equation (B.19)
          Update $\overline{\boldsymbol{W}}_{i,:}^{(it)}$ and $\overline{\boldsymbol{W}}_{j,:}^{(it)}$ by equation (B.18)
        end while
      end for
    end for
  end while
  Return: $\overline{\boldsymbol{W}}$ is the obtained nonnegative compression matrix
end procedure

Bibliography

[Adalı et al., 2014] Adalı, T., Anderson, M., and Fu, G. S. (2014). Diversity in independent component and vector analyses: Identifiability, algorithms, and applications in medical imaging. *IEEE Signal Processing Magazine*, 31(3):18–33.

[Afsari, 2006] Afsari, B. (2006). Simple LU and QR based non-orthogonal matrix joint diagonalization. In *Independent Component Analysis and Blind Signal Separation, Lecture Notes in Computer Science*, volume 3889, pages 1–7.

[Afsari, 2008] Afsari, B. (2008). Sensitivity analysis for the problem of matrix joint diagonalization. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1148–1171.

[Albera et al., 2010] Albera, L., Comon, P., Parra, L., Karfoul, A., Kachenoura, A., and Senhadji, L. (2010). Biomedical applications. In Comon, P. and Jutten, C., editors, *Handbook of Blind Source Separation*, chapter 18, pages 737–777. Elsevier, Oxford, UK.

[Albera et al., 2005] Albera, L., Ferréol, A., Chevalier, P., and Comon, P. (2005). ICAR: a tool for blind source separation using fourth-order statistics only. *IEEE Transactions on Signal Processing*, 53(10):3633–3643.

[Albera et al., 2004] Albera, L., Ferréol, A., Comon, P., and Chevalier, P. (2004). Blind identification of overcomplete mixtures of sources (BIOME). *Linear Algebra and its Applications*, 391:3–30.

[Albera et al., 2008] Albera, L., Ferréol, A., Cosandier-Rimele, D., Merlet, I., and Wendling, F. (2008). Brain source localization using a fourth-order deflation scheme. *IEEE Transactions on Biomedical Engineering*, 55(2):490–501.

[Albera et al., 2012] Albera, L., Kachenoura, A., Comon, P., Karfoul, A., Wendling, F., Senhadji, L., and Merlet, I. (2012). ICA-based EEG denoising: a comparative analysis of fifteen methods. *Bulletin of the Polish Academy of Sciences - Technical sciences*, 60(3):407–418.

[Albera and Karfoul, 2009] Albera, L. and Karfoul, A. (2009). A fast INDSCAL analysis for blind underdetermined mixture identification from several even higher order cumulants. In *DSP/SPE '09, Proceedings of IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop*, pages 393–398, Marco Island, FL.

[Almeida, 2003] Almeida, L. B. (2003). MISEP – linear and nonlinear ICA based on mutual information. *Journal of Machine Learning Research*, 4:1297–1318.

[Almeida and Almeida, 2012] Almeida, M. S. and Almeida, L. B. (2012). Nonlinear separation of show-through image mixtures using a physical model trained with ICA. *Signal Processing*, 92(4):872–884.

[Babaie-Zadeh and Jutten, 2005] Babaie-Zadeh, M. and Jutten, C. (2005). A general approach for mutual information minimization and its application to blind source separation. *Signal Processing*, 85(5):975–995.

[Babaie-Zadeh et al., 2004] Babaie-Zadeh, M., Jutten, C., and Nayebi, K. (2004). Differential of the mutual information. *IEEE Signal Processing Letters*, 11(1):48–51.

[Bakir et al., 2006] Bakir, T., Peter, A., Riley, R., and Hackett, J. (2006). Non-negative maximum likelihood ICA for blind source separation of images and signals with application to hyperspectral image subpixel demixing. In *Proceedings of 2006 IEEE International Conference on Image Processing*, pages 3237–3240, Atlanta, GA.

[Bartlett, 1951] Bartlett, M. S. (1951). An inverse matrix adjustment arising in discriminant analysis. *Annals of Mathematical Statistics*, 22(1):107–111.

[Bartlett et al., 2002] Bartlett, M. S., Movellan, J. R., and Sejnowski, T. J. (2002). Face recognition by independent component analysis. *IEEE Transactions on Neural Networks*, 13(6):1450–1464.

[Be'ery and Yeredor, 2008] Be'ery, E. and Yeredor, A. (2008). Blind separation of superimposed shifted images using parameterized joint diagonalization. *IEEE Transactions on Image Processing*, 17(3):340–353.

[Befroy and Shulman, 2011] Befroy, D. E. and Shulman, G. I. (2011). Magnetic resonance spectroscopy studies of human metabolism. *Diabetes*, 60(5):1361–1369.

[Bell and Sejnowski, 1995] Bell, A. J. and Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159.

[Belouchrani et al., 1997] Belouchrani, A., Abed-Meraim, K., Cardoso, J. F., and Moulines, E. (1997). A blind source separation technique using second-order statistics. *IEEE Transactions on Signal Processing*, 45(2):434–444.

[Berry et al., 2007] Berry, W., Browne, M., Langville, A. N., Pauca, V. P., and Plemmons, R. J. (2007). Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics and Data Analysis*, 52(7):155–173.

[Bertsekas, 1996] Bertsekas, D. P. (1996). *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific.

[Bertsekas, 1999] Bertsekas, D. P. (1999). *Nonlinear Programming*. Athena Scientific, second edition.

[Bevilacqua et al., 2012] Bevilacqua, M., Roumy, A., Guillemot, C., and Morel, M.-L. (2012). Neighbor embedding based single-image super-resolution using semi-nonnegative matrix factorization. In *ICASSP'12, Proceedings of 2012 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1289–1292, Kyoto.

[Blaschke and Wiskott, 2004] Blaschke, T. and Wiskott, L. (2004). CuBICA: independent component analysis by simultaneous third- and fourth-order cumulant diagonalization. *IEEE Transactions on Signal Processing*, 52(5):1250–1256.

[Boscolo et al., 2004] Boscolo, R., Pan, H., and Roychowdhury, V. (2004). Independent component analysis based on nonparametric density estimation. *IEEE Transactions on Neural Networks*, 15(1):55–65.

[Boyd et al., 2010] Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2010). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122.

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, Cambridge, UK.

[Bro and De Jong, 1997] Bro, R. and De Jong, S. (1997). A fast non-negativity-constrained least squares algorithm. *Journal of Chemometrics*, 11:393–401.

[Brown and Smaragdis, 2004] Brown, J. C. and Smaragdis, P. (2004). Independent component analysis for automatic note extraction from musical trills. *Journal of the Acoustical Society of America*, 115(5):2295–2306.

[Brown et al., 2002] Brown, J. C., Todd, J. G., , and Smaragdis, P. (2002). Independent component analysis for onset detection in piano trills. *Journal of the Acoustical Society of America*, 111(5):2446–2446.

[Buciu, 2008] Buciu, J. (2008). Non-negative matrix factorization, a new tool for feature extraction: Theory and applications. *International Journal of Computers, Communications and Control*, 3(3):67–74.

[Burg, 2008] Burg, J. (2008). *The Science of Digital Media*. Prentice Hall, United States, 1 edition.

[Cardoso, 1991] Cardoso, J. F. (1991). Super-symmetric decomposition of the fourth-order cumulant tensor. Blind identification of more sources than sensors. In *ICASSP'91, Proceedings of 1991 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 3109–3112, Toronto, Ont.

[Cardoso, 1997] Cardoso, J. F. (1997). Infomax and maximum likelihood for blind source separation. *IEEE Signal Processing Letters*, 4(4):112–114.

[Cardoso, 2010] Cardoso, J. F. (2010). Likelihood. In Comon, P. and Jutten, C., editors, *Handbook of Blind Source Separation*, chapter 4, pages 107–154. Elsevier, Oxford, UK.

[Cardoso and Souloumiac, 1993] Cardoso, J. F. and Souloumiac, A. (1993). Blind beamforming for non-Gaussian signals. *IEE Proceedings F Radar and Signal Processing*, 140(6):362–370.

[Cardoso and Souloumiac, 1996] Cardoso, J. F. and Souloumiac, A. (1996). Jacobi angles for simultaneous diagonalization. *SIAM Journal on Matrix Analysis and Applications*, 17:161–164.

[Carroll and Chang, 1970] Carroll, J. D. and Chang, J.-J. (1970). Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart-Young decomposition. *Psychometrika*, 35(3):283–319.

[Chabriel and Barrère, 2012] Chabriel, G. and Barrère, J. (2012). A direct algorithm for nonorthogonal approximate joint diagonalization. *IEEE Transactions on Signal Processing*, 60(1):39–47.

[Chabriel et al., 2014] Chabriel, G., Kleinsteuber, M., Moreau, E., Shen, H., Tichavský, P., and Yeredor, A. (2014). Joint matrices decompositions and blind source separation: A survey of methods, identification, and applications. *IEEE Signal Processing Magazine*, 31(3):34–43.

[Chan et al., 2009] Chan, T. H., Chi, C. Y., Huang, Y. M., and Ma, W. K. (2009). A convex analysis-based minimum-volume enclosing simplex algorithm for hyperspectral unmixing. *IEEE Transactions on Signal Processing*, 57(11):4418–4432.

[Chevalier and Chevreuil, 2010] Chevalier, P. and Chevreuil, A. (2010). Application to telecommunications. In Comon, P. and Jutten, C., editors, *Handbook of Blind Source Separation*, chapter 17, pages 683–735. Elsevier, Oxford, UK.

[Chu et al., 2004] Chu, M., Diele, F., Plemmons, R., and Ragni, S. (2004). Optimality computation and interpretation of nonnegative matrix factorizations. Technical report, North Carolina State University.

[Cichocki, 2013] Cichocki, A. (2013). Unsupervised learning algorithms and latent variable models: PCA/SVD, CCA/PLS, ICA, NMF. In Chellappa, R. and Theodoridis, S., editors, *Signal Processing Theory and Machine Learning*, volume 1, chapter 21, pages 1151–1238. Academic Press.

[Cichocki and Amari, 2010] Cichocki, A. and Amari, S. (2010). Families of alpha- beta- and gamma- divergences: Flexible and robust measures of similarities. *Entropy*, 12(6):1532–1568.

[Cichocki et al., 2008] Cichocki, A., Lee, H., Kim, Y. D., and Choi, S. (2008). Non-negative matrix factorization with α-divergence. *Pattern Recognition Letters*, 29(9):1433–1440.

[Cichocki and Phan, 2009] Cichocki, A. and Phan, A.-H. (2009). Fast local algorithms for large scale nonnegative matrix and tensor factorizations. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E92.A(3):708–721.

[Cichocki et al., 2009] Cichocki, A., Zdunek, R., Phan, A. H., and Amari, S. (2009). *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. WILEY, West Sussex, United Kingdom.

[Coloigner, 2012] Coloigner, J. (2012). *Line search and trust region strategies for canonical decomposition of semi-nonnegative semi-symmetric tensors*. PhD thesis, Université de Rennes 1.

[Coloigner et al., 2014a] Coloigner, J., Albera, L., Kachenoura, A., Noury, F., and Senhadji, L. (2014a). Semi-nonnegative joint diagonalization by congruence and semi-nonnegative ICA. *Signal Processing*, 105:185–197.

[Coloigner et al., 2011a] Coloigner, J., Albera, L., Kachenoura, A., and Senhadji, L. (2011a). Canonical decomposition of semi-symmetric semi-nonnegative three-way arrays. In *ASILOMAR'11, Proceedings of IEEE Forty-Fifth Asilomar Conference on Signals, Systems and Computers*, Asilomar.

[Coloigner et al., 2011b] Coloigner, J., Albera, L., Karfoul, A., and Comon, P. (2011b). Décomposition de tableaux d'ordre trois semi-nonnégatifs et semi-symétriques. In *GRETSI'11, Vingt-troisième colloque sur le Traitement du Signal et des Images*, Bordeaux, France.

[Coloigner et al., 2010] Coloigner, J., Albera, L., Karfoul, A., Kachenoura, A., Comon, P., and Senhadji, L. (2010). Semi-nonnegative independent component analysis: the (3,4)-SENICA$^{\text{exp}}$ method. In *Independent Component Analysis and Signal Separation Lecture Notes in Computer Science*, volume 6365, pages 612–619, Saint Malo, France.

[Coloigner et al., 2014b] Coloigner, J., Fargeas, A., Kachenoura, A., Wang, L., Senhadji, L., De Crevoisier, R., Acosta, O., and Albera, L. (2014b). A novel classification method for prediction of rectal bleeding in prostate cancer radiotherapy based on a semi-nonnegative ICA of 3D planned dose distributions. Accepted for publication in *IEEE Journal of Biomedical and Health Informatics*.

[Coloigner et al., 2014c] Coloigner, J., Karfoul, A., Albera, L., and Comon, P. (2014c). Line search and trust region strategies for canonical decomposition of semi-nonnegative semi-symmetric 3rd order tensors. *Linear Algebra and its Applications*, 450(1):334–374.

[Comon, 1994] Comon, P. (1994). Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314.

[Comon, 2002] Comon, P. (2002). Tensor decompositions, state of the art and applications. In McWhirter, J. G. and Proudler, I. K., editors, *Mathematics in Signal Processing V*, pages 1–24. Clarendon Press, Oxford, UK.

[Comon, 2004] Comon, P. (2004). Blind identification and source separation in 2×3 under-determined mixtures. *IEEE Transactions on Signal Processing*, 52(1):1–13.

[Comon, 2014] Comon, P. (2014). Tensors: A brief introduction. *IEEE Signal Processing Magazine*, 31(3):44–53.

[Comon and Jutten, 2010] Comon, P. and Jutten, C., editors (2010). *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Academic Press.

[Comon et al., 2009] Comon, P., Luciani, X., and de Almeida, A. L. F. (2009). Tensor decompositions, alternating least squares and other tales. *Journal of Chemometrics*, 23:393–405.

[Comon and Moreau, 1997] Comon, P. and Moreau, E. (1997). Improved contrast dedicated to blind separation in communications. In *ICASSP'97, Proceedings of 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 3453–3456.

[Comon and Rajih, 2006] Comon, P. and Rajih, M. (2006). Blind identification of under-determined mixtures based on the characteristic function. *Signal Processing*, 86(9):2271–2281.

[Cover and Thomas, 1991] Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. Wiley, New York.

[Das Gupta and Xiao, 2011] Das Gupta, M. and Xiao, J. (2011). Non-negative matrix factorization as a feature selection tool for maximum margin classifiers. In *CVPR'11, Proceedings of 2011 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2841–2848.

[Davies, 2004] Davies, M. (2004). Identifiability issues in noisy ICA. *IEEE Signal Processing Letters*, 11(5):470–473.

[De Lathauwer, 2006] De Lathauwer, L. (2006). A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization. *SIAM Journal on Matrix Analysis and Applications*, 28(3):642–666.

[De Lathauwer, 2010] De Lathauwer, L. (2010). Algebraic methods after prewhitening. In Comon, P. and Jutten, C., editors, *Handbook of Blind Source Separation*, chapter 5, pages 155–177. Elsevier, Oxford, UK.

[De Lathauwer and Castaing, 2008] De Lathauwer, L. and Castaing, J. (2008). Blind identification of underdetermined mixtures by simultaneous matrix diagonalization. *IEEE Transactions on Signal Processing*, 56(3):1096–1105.

[De Lathauwer et al., 2000] De Lathauwer, L., De Moor, B., and Vandewalle, J. (2000). Fetal electrocardiogram extraction by blind source subspace separation. *IEEE Transactions on Biomedical Engineering*, 47(5):567–572.

[De Lathauwer et al., 2001] De Lathauwer, L., De Moor, B., and Vandewalle, J. (2001). Independent component analysis and (simultaneous) third-order tensor diagonalization. *IEEE Transactions on Signal Processing*, 49(10):2262–2271.

[De Lathauwer et al., 2004] De Lathauwer, L., De Moor, B., and Vandewalle, J. (2004). Computation of the canonical decomposition by means of a simultaneous generalized Schur decomposition. *SIAM Journal on Matrix Analysis and Applications*, 26(2):295–327.

[Dégerine and Kane, 2007] Dégerine, S. and Kane, E. (2007). A comparative study of approximate joint diagonalization algorithms for blind source separation in presence of additive noise. *IEEE Transactions on Signal Processing*, 55(6):3022–3031.

[Dégerine and Zaïdi, 2004] Dégerine, S. and Zaïdi, A. (2004). Separation of an instantaneous mixture of Gaussian autoregressive sources by the exact maximum likelihood approach. *IEEE Transactions on Signal Processing*, 52(6):1499–1512.

[Delfosse and Loubaton, 1995] Delfosse, N. and Loubaton, P. (1995). Adaptive blind separation of independent sources: A deflation approach. *Signal Processing*, 45(1):59–83.

[Ding et al., 2010] Ding, C., Li, T., and Jordan, M. (2010). Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):45–55.

[Dittmar and Uhle, 2004] Dittmar, C. and Uhle, C. (2004). Further steps towards drum transcription of polyphonic music. In *Audio Engineering Society Convention 116*.

[Donoho and Stodden, 2003] Donoho, D. L. and Stodden, V. (2003). When does nonnegative matrix factorization give a correct decomposition into parts? In *Proceedings of Neural Information Processing Systems*, volume 16, pages 1–8.

[Eriksson and Koivunen, 2004] Eriksson, J. and Koivunen, V. (2004). Identifiability, separability, uniqueness of linear ICA models. *IEEE Signal Processing Letters*, 11(7):601–604.

[Fadaili et al., 2007] Fadaili, E. M., Thirion-Moreau, N., and Moreau, E. (2007). Nonorthogonal joint diagonalization/zero diagonalization for source separation based on time-frequency distributions. *IEEE Transactions on Signal Processing*, 55(5):1673–1687.

[Ferréol et al., 2005] Ferréol, A., Albera, L., and Chevalier, P. (2005). Fourth order blind identification of underdetermined mixtures of sources (FOBIUM). *IEEE Transactions on Signal Processing*, 53(4):1640–1653.

[Févotte et al., 2009] Févotte, C., Bertin, N., and Durrieu, J.-L. (2009). Nonnegative matrix factorization with the Itakura-Saito divergence: with application to music analysis. *Neural Computation*, 21(3):793–830.

[Févotte and Idier, 2011] Févotte, C. and Idier, J. (2011). Algorithms for nonnegative matrix factorization with the β-divergence. *Neural Computation*, 23(9):2421–2456.

[Flury and Gautsch, 1986] Flury, B. N. and Gautsch, W. (1986). An algorithm for simultaneous orthogonal transformation of several positive definite symmetric matrices to nearly diagonal form. *SIAM Journal on Scientific and Statistical Computing*, 7:169–184.

[Fortin and Glowinski, 1983] Fortin, M. and Glowinski, R., editors (1983). *Augmented Lagrangian methods: applications to the numerical solution of boundary-value problems*. Elsevier, Amsterdam, North-Holland.

[Fuentes et al., 2013] Fuentes, B., Badeau, R., and Richard, G. (2013). Harmonic adaptive latent component analysis of audio and application to music transcription. *IEEE Transactions on Audio, Speech and Language Processing*, 21(9):1854–1866.

[Gabay and Mercier, 1976] Gabay, D. and Mercier, B. (1976). A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers and Mathematics with Applications*, 2(1):17–40.

[Gao et al., 2011] Gao, B., Woo, W. L., and Dlay, S. S. (2011). Adaptive sparsity non-negative matrix factorization for single-channel source separation. *IEEE Journal of Selected Topics in Signal Processing*, 5(5):989–1001.

[Ge and Ma, 2010] Ge, F. and Ma, J. (2010). Spurious solution of the maximum likelihood approach to ICA. *IEEE Signal Processing Letters*, 17(7):655–658.

[Ghennioui et al., 2009] Ghennioui, H., Thirion-Moreau, N., and Moreau, E. (2009). An optimal step size relative gradient based joint diagonalization algorithm. In *CAMSAP 09, Proceedings of 3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, pages 320–323.

[Gillis, 2012] Gillis, N. (2012). Sparse and unique nonnegative matrix factorization through data preprocessing. *Journal of Machine Learning Research*, 13(11):3349–3386.

[Gillis and Glineur, 2010] Gillis, N. and Glineur, F. (2010). Using underapproximations for sparse nonnegative matrix factorization. *Pattern Recognition*, 43(4):1676–1687.

[Girolami, 1998] Girolami, M. (1998). An alternative perspective on adaptive independent component analysis algorithms. *Neural Computation*, 10(8):2103–2114.

[Glowinski and Marroco, 1975] Glowinski, R. and Marroco, A. (1975). Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 9(R2):41–76.

[Golub and Van Loan, 1996] Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, 3rd edition.

[Gong et al., 2007] Gong, T., Xuan, J., Wang, C., Li, H., Hoffman, E., Clarke, R., and Wang, Y. (2007). Gene module identification from microarray data using nonnegative independent component analysis. *Journal of Gene Regulation and Systems Biology*, 1:349–363.

[Gong et al., 2012] Gong, X. F., Wang, K., and Lin, Q. H. (2012). Complex non-orthogonal joint diagonalization with successive Givens and hyperbolic rotations. In *ICASSP'12, Proceedings of 2012 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1189–1892.

[Gonzalez and Zhang, 2005] Gonzalez, E. F. and Zhang, Y. (2005). Accelerating the Lee-Seung algorithm for non-negative matrix factorization. Technical report, Department of Computational and Applied Mathematics, Rice University.

[Guan et al., 2012] Guan, N. Y., Tao, D. C., Luo, Z. G., and Yuan, B. (2012). NeNMF: An optimal gradient method for nonnegative matrix factorization. *IEEE Transactions on Signal Processing*, 60(6):2882–2898.

[Guo et al., 2010] Guo, X., Zhu, S., Miron, S., and Brie, D. (2010). Approximate joint diagonalization by nonorthogonal nonparametric Jacobi transformations. In *ICASSP'10, Proceedings of 2010 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 3774–3777.

[Hajipour Sardouie et al., 2013] Hajipour Sardouie, S., Albera, L., Shamsollahi, M. B., and Merlet, I. (2013). Canonical polyadic decomposition of complex-valued multiway arrays based on simultaneous Schur decomposition. In *ICASSP'13, Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4178–4182, Vancouver, BC.

[Hajipour Sardouie et al., 2014] Hajipour Sardouie, S., Shamsollahi, M. B., Albera, L., and Merlet, I. (2014). Denoising of ictal EEG data using semi-blind source separation methods based on time-frequency priors. Accepted for publication in *IEEE Journal of Biomedical and Health Informatics*.

[Harshman, 1970] Harshman, R. A. (1970). Foundations of the PARAFAC procedure: Models and conditions for an explanatory multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16(1):1–84.

[Harshman and Lundy, 1984] Harshman, R. A. and Lundy, M. E. (1984). The PARAFAC model for three-way factor analysis and multidimensional scaling. In Law, H. G., Snyder, C. W., Jr., J. H., and McDonald, R. P., editors, *Research methods for multimode data analysis*, pages 122–215. Praeger.

[Harshman and Lundy, 1994] Harshman, R. A. and Lundy, M. E. (1994). PARAFAC: Parallel factor analysis. *Computational Statistics and Data Analysis*, 18(1):39–72.

[He et al., 2000] He, Z. Y., Yang, L. X., Liu, J., Lu, Z. Y., He, C., and Shi, Y. H. (2000). Blind source separation using clustering-based multivariate density estimation algorithm. *IEEE Transactions on Signal Processing*, 48(2):575–579.

[Hérault and Jutten, 1986] Hérault, J. and Jutten, C. (1986). Space or time adaptive signal processing by neural network models. In *Neural networks for computing*, volume 151, pages 206–211.

[Hestenes, 1969] Hestenes, M. R. (1969). Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4(5):303–320.

[Holzapfel and Stylianou, 2008] Holzapfel, A. and Stylianou, Y. (2008). Musical genre classification using nonnegative matrix factorization-based features. *IEEE Transactions on Audio, Speech and Language Processing*, 16(2):424–434.

[Horská and Tkáč, 2012] Horská, A. and Tkáč, I. (2012). Magnetic resonance spectroscopy: Clinical applications. In Faro, S. H., Mohamed, F. B., Law, M., and Ulmer, J. T., editors, *Functional Neuroradiology: Principles and Clinical Applications*, chapter 9, pages 155–194. Springer.

[Huang et al., 2014] Huang, K. J., Sidiropoulos, N. D., and Swami, A. (2014). Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition. *IEEE Transactions on Signal Processing*, 62(1):211–224.

[Hulle, 2008] Hulle, M. M. V. (2008). Sequential fixed-point ICA based on mutual information minimization. *Neural Computation*, 20(5):1344–1365.

[Hyvärinen, 1997] Hyvärinen, A. (1997). New approximations of differential entropy for independent component analysis and projection pursuit. In *NIPS'97, Proceedings of the 1997 conference on advances in neural information processing systems 10*, pages 273–279.

[Hyvärinen, 1999] Hyvärinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634.

[Hyvärinen, 2013] Hyvärinen, A. (2013). Independent component analysis: Recent advances. *Philosophical Transactions of the Royal Society A*, 371(1984):1–19.

[Hyvärinen et al., 2001] Hyvärinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis*. Wiley.

[Hyvärinen and Oja, 1997] Hyvärinen, A. and Oja, E. (1997). A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492.

[Hyvärinen and Oja, 2000] Hyvärinen, A. and Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural Networks*, 13(4–5):411–430.

[Jiang and Sidiropoulos, 2004] Jiang, T. and Sidiropoulos, N. D. (2004). Kruskal's permutation lemma and the identification of CANDECOMP/PARAFAC and bilinear models with constant modulus constraints. *IEEE Transactions on Signal Processing*, 52(9):2625–2636.

[Joho, 2008] Joho, M. (2008). Newton method for joint approximate diagonalization of positive definite Hermitian matrices. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1205–1218.

[Joho et al., 2000] Joho, M., Mathis, H., and Lambert, R. H. (2000). Overdetermined blind source separation: Using more sensors than source signals in a noisy mixture. In *Proceedings of International Conference on Independent Component Analysis and Blind Signal Separation*, pages 81–86.

[Jorgensen, 1991] Jorgensen, O. H. (1991). *Tuning*. Michigan State University Press, East Lansing, MI.

[Jutten, 1987] Jutten, C. (1987). *Calcul neuromimétique et traitement du signal: analyse en composantes indépendantes*. PhD thesis, Université Joseph Fourier et Institut Polytechnique de Grenoble.

[Jutten and Hérault, 1991] Jutten, C. and Hérault, J. (1991). Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24(1):1–10.

[Kachenoura et al., 2008] Kachenoura, A., Albera, L., Senhadji, L., and Comon, P. (2008). ICA: a potential tool for BCI systems. *IEEE Signal Processing Magazine*, 25(1):57–68.

[Karfoul et al., 2010] Karfoul, A., Albera, L., and Birot, G. (2010). Blind underdetermined mixture identification by joint canonical decomposition of HO cumulants. *IEEE Transactions on Signal Processing*, 58(2):638–649.

[Karfoul et al., 2011] Karfoul, A., Albera, L., and De Lathauwer, L. (2011). Iterative methods for the canonical decomposition of multi-way arrays: Application to blind underdetermined mixture identification. *Signal Processing*, 91(8):1789–1802.

[Kim et al., 2007] Kim, D., Sra, S., and Dhillon, I. S. (2007). Fast Newton-type methods for the least squares nonnegative matrix approximation problem. In *Proceedings of SIAM Data Mining Conference*, pages 343–354.

[Kim and Park, 2008] Kim, H. and Park, H. (2008). Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM Journal on Matrix Analysis and Applications*, 30(2):713–730.

[Kim et al., 2014] Kim, J., He, Y., and Park, H. (2014). Algorithms for nonnegative matrix and tensor factorizations: a unified view based on block coordinate descent framework. *Journal of Global Optimization*, 58(2):285–319.

[Kim et al., 2005] Kim, J. S., Choi, J., Yi, J., and Turk, M. (2005). Effective representation using ICA for face recognition robust to local distortion and partial occlusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1977–1981.

[Kleinsteuber and Shen, 2013] Kleinsteuber, M. and Shen, H. (2013). Uniqueness analysis of non-unitary matrix joint diagonalization. *IEEE Transactions on Signal Processing*, 61(7):1786–1796.

[Kokkinakis and Nandi, 2007] Kokkinakis, K. and Nandi, A. K. (2007). Generalized gamma density-based score functions for fast and flexible ICA. *Signal Processing*, 87(5):1156–1162.

[Kolda and Bader, 2009] Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Review*, 51(3):455–500.

[Koldovský et al., 2006] Koldovský, Z., Tichavský, P., and Oja, E. (2006). Efficient variant of algorithm fastICA for independent component analysis attaining the Cramár-Rao lower bound. *IEEE Transactions on Neural Networks*, 17(5):1262–1277.

[Koldovský et al., 2011] Koldovský, Z., Tichavský, P., and Phan, A. H. (2011). Stability analysis and fast damped-Gauss-Newton algorithm for INDSCAL tensor decomposition. In *SSP '11, Proceedings of 2011 IEEE Statistical Signal Processing Workshop*, pages 581–584, Nice.

[Kompass, 2007] Kompass, R. (2007). A generalized divergence measure for nonnegative matrix factorization. *Neural Computation*, 19(3):780–791.

[Kraskov et al., 2004] Kraskov, A., Stögbauer, H., and Grassberger, P. (2004). Estimating mutual information. *Physical Review E*, 69(066138):1–16.

[Kruskal, 1977] Kruskal, J. B. (1977). Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra and its Applications*, 18(2):98–138.

[Kumar et al., 2011] Kumar, V. B., Patras, I., and Kotsia, I. (2011). Max-margin semi-NMF. In *Proceedings of the British Machine Vision Conference*, pages 129.1–129.11.

[Laurberg et al., 2008] Laurberg, H., Christensen, M. G., Plumbley, M. D., Hansen, L. K., and Jensen, S. H. (2008). Theorems on positive data: On the uniqueness of NMF. *Computational Intelligence and Neuroscience*, 2008:1–10.

[Lawson and Hanson, 1974] Lawson, C. L. and Hanson, R. J. (1974). *Solving Least Squares Problems*, volume 161. Prentice-hall, Englewood Cliffs, NJ.

[Le Roux et al., 2008] Le Roux, J., De Cheveigné, A., and Parra, L. C. (2008). Adaptive template matching with shift-invariant semi-NMF. In *Proceedings of Neural Information Processing Systems*.

[Lee and Seung, 1999] Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

[Lee and Seung, 2000] Lee, D. D. and Seung, H. S. (2000). Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, pages 556–562.

[Lee et al., 1999] Lee, T. W., Girolami, M., and Sejnowski, T. J. (1999). Independent component analysis using an extended infomax algorithm for mixed sub-Gaussian and super-Gaussian sources. *Neural Computation*, 11(2):417–441.

[Li and Zhang, 2009] Li, L. and Zhang, Y. J. (2009). FastNMF: highly efficient monotonic fixed-point nonnegative matrix factorization algorithm with good applicability. *Journal of Electronic Imaging*, 18(3):033004–033004–12.

[Li and Ding, 2006] Li, T. and Ding, C. (2006). The relationships among various nonnegative matrix factorization methods for clustering. In *ICDM'06, Proceedings of Sixth International Conference on Data Mining*, pages 362–371, Hong Kong.

[Li and Adalı, 2010] Li, X. L. and Adalı, T. (2010). Independent component analysis by entropy bound minimization. *IEEE Transactions on Signal Processing*, 58(10):5151–5164.

[Li and Zhang, 2007] Li, X. L. and Zhang, X. D. (2007). Nonorthogonal joint diagonalization free of degenerate solution. *IEEE Transactions on Signal Processing*, 55(5):1803–1814.

[Li et al., 2007] Li, X. L., Zhang, X. D., and Li, P. S. (2007). Gradient search nonorthogonal approximate joint diagonalization algorithm. *Tsinghua Science and Technology*, 12(6):669–673.

[Li and Liu, 1998] Li, Y. and Liu, K. (1998). Adaptive blind source separation and equalization for multiple-input/multiple-output systems. *IEEE Transactions on Information Theory*, 44(7):2864–2876.

[Lin, 2007] Lin, C. J. (2007). Projected gradient methods for nonnegative matrix factorization. *Neural Computation*, 19(10):2756–2779.

[Liu et al., 2010] Liu, C., chih Yang, H., Fan, J. L., He, L. W., and Wang, Y. M. (2010). Distributed nonnegative matrix factorization for web-scale dyadic data analysis on MapReduce. In *WWW'10, Proceedings of the 19th International World Wide Web Conference*, pages 681–690.

[Luciani and Albera, 2010] Luciani, X. and Albera, L. (2010). Joint eigenvalue decomposition using polar matrix factorization. In *Latent Variable Analysis and Signal Separation, Lecture Notes in Computer Science*, volume 6365, pages 555–562.

[Luciani and Albera, 2011] Luciani, X. and Albera, L. (2011). Semi-algebraic canonical decomposition of multi-way arrays and joint eigenvalue decomposition. In *ICASSP'11, Proceedings of 2011 IEEE International Conference on Acoustics Speech and Signal Processing*, pages 4104–4107, Prague, Czech Republic.

[Luciani and Albera, 2014] Luciani, X. and Albera, L. (2014). Canonical polyadic decomposition based on joint eigenvalue decomposition. *Chemometrics and Intelligent Laboratory Systems*, 132:152–167.

[Ma et al., 2014] Ma, W. K., Bioucas-Dias, J. M., Chan, T. H., Gillis, N., Gader, P., Plaza, A. J., and Ambikapathi, A. (2014). A signal processing perspective on hyperspectral unmixing: Insights from remote sensing. *IEEE Signal Processing Magazine*, 31(1):67–81.

[Mankad and Michailidis, 2013] Mankad, S. and Michailidis, G. (2013). Discovery of path-important nodes using structured semi-nonnegative matrix factorization. In *CAMSAP'13, Proceedings of 2013 IEEE 5th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, pages 288–291, St. Martin.

[Maurandi et al., 2013] Maurandi, V., Luigi, C. D., and Moreau, E. (2013). Fast Jacobi like algorithms for joint diagonalization of complex symmetric matrices. In *EUSIPCO'13, Proceedings of the 21th European Signal Processing Conference*, pages 1–5, Marrakech, Morocco.

[Maurandi and Moreau, 2014a] Maurandi, V. and Moreau, E. (2014a). A decoupled Jacobi-like algorithm for non-unitary joint diagonalization of complex-valued matrices. *IEEE Signal Processing Letters*, 21(12):1453–1456.

[Maurandi and Moreau, 2014b] Maurandi, V. and Moreau, E. (2014b). A new fast Jacobi-like algorithm for non-orthogonal joint diagonalization of real-valued matrices based on a QR parameterization. In *MLSP'14, Proceedings of IEEE International Workshop on Machine Learning for Signal Processing*, Reims, France.

[Maurandi et al., 2014] Maurandi, V., Moreau, E., and De Luigi, C. (2014). Jacobi like algorithm for non-orthogonal joint diagonalization of Hermitian matrices. In

ICASSP'14, Proceedings of 2014 IEEE International Conference on Acoustics, Speech, and Signal Processing, pages 6196–6200, Florence.

[McCullagh, 1987] McCullagh, P. (1987). *Tensor Methods in Statistics*. Monographs on Statistics and Applied Probability. Chapman and Hall, London.

[Merrikh-Bayat et al., 2010] Merrikh-Bayat, F., Babaie-Zadeh, M., and Jutten, C. (2010). Using non-negative matrix factorization for removing show-through. In *Independent Component Analysis and Signal Separation Lecture Notes in Computer Science*, volume 6365, pages 482–489.

[Merrikh-Bayat et al., 2011] Merrikh-Bayat, F., Babaie-Zadeh, M., and Jutten, C. (2011). Linear-quadratic blind source separating structure for removing show-through in scanned documents. *International Journal on Document Analysis and Recognition*, 14(4):319–333.

[Merritt and Zhang, 2005] Merritt, M. and Zhang, Y. (2005). Interior-point gradient method for large-scale totally nonnegative least squares problems. *Journal of Optimization Theory and Applications*, 126(1):191–202.

[Mesloub et al., 2014] Mesloub, A., Abed-Meraim, K., and Belouchrani, A. (2014). A new algorithm for complex non-orthogonal joint diagonalization based on shear and Givens rotations. *IEEE Transactions on Signal Processing*, 62(8):1913–1925.

[Miao and Qi, 2007] Miao, L. and Qi, H. (2007). Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 45(3):765–777.

[Miele et al., 1972] Miele, A., Moseley, P. E., Levy, A. V., and Coggins, G. M. (1972). On the method of multipliers for mathematical programming problems. *Journal of Optimization Theory and Applications*, 10(1):1–33.

[Mo and Draper, 2012] Mo, Q. Y. and Draper, B. A. (2012). Semi-nonnegative matrix factorization for motion segmentation with missing data. In *Computer Vision – ECCV 2012, Lecture Notes in Computer Science*, volume 7578, pages 402–415.

[Moreau, 2001] Moreau, E. (2001). A generalization of joint-diagonalization criteria for source separation. *IEEE Transactions on Signal Processing*, 49(3):530–541.

[Moreau, 2006] Moreau, E. (2006). Comments on "CuBICA: Independent component analysis by simultaneous third- and fourth-order cumulant diagonalization". *IEEE Transactions on Signal Processing*, 54(12):4826–4828.

[Moreau and Adalı, 2013] Moreau, E. and Adalı, T. (2013). *Blind Identification and Separation of Complex-Valued Signals*. ISTE and Wiley, Hoboken, NJ.

[Moussaoui, 2005] Moussaoui, S. (2005). *Séparation de sources non-négatives: application au traitement des signaux de spectroscopie*. PhD thesis, Université Henri Poincaré.

[Moussaoui et al., 2008] Moussaoui, S., Hauksdóttir, H., Schmidt, F., Jutten, C., Chanussot, J., Brie, D., Douté, S., and Benediktsson, J. A. (2008). On the decomposition of Mars hyperspectral data by ICA and Bayesian positive source separation. *Neurocomputing*, 71(10–12):2194–2208.

[Naanaa and Nuzillard, 2005] Naanaa, W. and Nuzillard, J.-M. (2005). Blind source separation of positive and partially correlated data. *Signal Processing*, 85(9):1711–1722.

[Nakai et al., 2004] Nakai, T., Muraki, S., Bagarinao, E., Miki, Y., Takehara, Y., Matsuo, K., Kato, C., Sakahara, H., and Isoda, H. (2004). Application of independent component analysis to magnetic resonance imaging for enhancing the contrast of gray and white matter. *NeuroImage*, 21(1):251–260.

[Nedić and Ozdaglar, 2010] Nedić, A. and Ozdaglar, A. (2010). Cooperative distributed multi-agent optimization. In Eldar, Y. and Palomar, D., editors, *Convex Optimization in Signal Processing and Communications*, pages 340–386. Cambridge University Press.

[Nikias and Mendel, 1993] Nikias, C. and Mendel, J. (1993). Signal processing with higher-order spectra. *IEEE Signal Processing Magazine*, 10(3):10–37.

[Niknazar, 2013] Niknazar, M. (2013). *Extraction and denoising of fetal ECG signals.* PhD thesis, Université de Grenoble.

[Obradovic and Deco, 1998] Obradovic, D. and Deco, G. (1998). Information maximization and independent component analysis: is there a difference? *Neural Computation*, 10(8):2085–2101.

[Oja, 1997] Oja, E. (1997). The nonlinear PCA learning rule in independent component analysis. *Neurocomputing*, 17(1):25–45.

[Oja and Plumbley, 2004] Oja, E. and Plumbley, M. D. (2004). Blind separation of positive sources by globally convergent gradient search. *Neural Computation*, 16(9):1811–1825.

[Ouedraogo et al., 2011] Ouedraogo, W. S. B., Jaidane, M., Souloumiac, A., and Jutten, C. (2011). Regularized gradient algorithm for non-negative independent component analysis. In *ICASSP'11, Proceedings of 2011 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2524–2527, Prague, Czech Republic.

[Ouedraogo et al., 2010] Ouedraogo, W. S. B., Souloumiac, A., and Jutten, C. (2010). Non-negative independent component analysis algorithm based on 2D Givens rotations and a Newton optimization. In *Latent Variable Analysis and Signal Separation, Lecture Notes in Computer Science*, volume 6365, pages 522–529.

[Paatero, 1997] Paatero, P. (1997). Least squares formulation of robust non-negative factor analysis. *Chemometrics and Intelligent Laboratory Systems*, 37(1):23–35.

[Paatero and Tapper, 1994] Paatero, P. and Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126.

[Papadias, 2000] Papadias, C. (2000). Globally convergent blind source separation based on a multiuser kurtosis maximization criterion. *IEEE Transactions on Signal Processing*, 48(12):3508–3519.

[Park and Kim, 2006] Park, H. and Kim, H. (2006). One-sided non-negative matrix factorization and non-negative centroid dimension reduction for text classification. In *Proceedings of the Workshop on Text Mining at the 6th SIAM International Conference on Data Mining*.

[Pesquet and Moreau, 2001] Pesquet, J. and Moreau, E. (2001). Cumulant-based independence measures for linear mixtures. *IEEE Transactions on Information Theory*, 47(5):1947–1956.

[Pham, 2000] Pham, D. T. (2000). Blind separation of instantaneous mixture of sources based on order statistics. *IEEE Transactions on Signal Processing*, 48(2):363–375.

[Pham, 2001a] Pham, D. T. (2001a). Blind separation of instantaneous mixture of sources via the Gaussian mutual information criterion. *Signal Processing*, 81(4):855–870.

[Pham, 2001b] Pham, D. T. (2001b). Joint approximate diagonalization of positive definite Hermitian matrices. *SIAM Journal on Matrix Analysis and Applications*, 22:1837–1848.

[Pham, 2004] Pham, D. T. (2004). Fast algorithms for mutual information based independent component analysis. *IEEE Transactions on Signal Processing*, 52(10):2690–2700.

[Pham and Cardoso, 2001] Pham, D. T. and Cardoso, J. F. (2001). Blind separation of instantaneous mixtures of nonstationary sources. *IEEE Transactions on Signal Processing*, 49(9):1837–1848.

[Pham and Congedo, 2009] Pham, D. T. and Congedo, M. (2009). Least square joint diagonalization of matrices under an intrinsic scale constraint. In *Independent Component Analysis and Signal Separation Lecture Notes in Computer Science*, volume 5441, pages 298–305.

[Pham and Garat, 1997] Pham, D. T. and Garat, P. (1997). Blind separation of mixture of independent sources through a quasi-maximum likelihood approach. *IEEE Transactions on Signal Processing*, 45(7):1712–1725.

[Pham et al., 1992] Pham, D. T., Garat, P., and Jutten, C. (1992). Separation of a mixture of independent sources through a maximum likelihood approach. In *EUSIPCO'92, Proceedings of the European Signal Processing Conference*, pages 771–774.

[Plumbley, 2002] Plumbley, M. D. (2002). Conditions for nonnegative independent component analysis. *IEEE Signal Processing Letters*, 9(6):177–180.

[Plumbley, 2003] Plumbley, M. D. (2003). Algorithms for nonnegative independent component analysis. *IEEE Transactions on Neural Networks*, 14(3):534–543.

[Plumbley, 2005] Plumbley, M. D. (2005). Geometrical methods for non-negative ICA: Manifolds, Lie groups and toral subalgebras. *Neurocomputing*, 67:161–197.

[Plumbley and Abdallah, 2003] Plumbley, M. D. and Abdallah, S. A. (2003). An independent component analysis approach to automatic music transcription. In *Proceedings of the 114th Audio Engineering Society Convention*, Amsterdam, The Netherlands.

[Plumbley et al., 2010] Plumbley, M. D., Cichocki, A., and Bro, R. (2010). Non-negative mixtures. In Comon, P. and Jutten, C., editors, *Handbook of Blind Source Separation*, chapter 13, pages 515–547. Elsevier, Oxford, UK.

[Plumbley and Oja, 2004] Plumbley, M. D. and Oja, E. (2004). A nonnegative PCA algorithm for independent component analysis. *IEEE Transactions on Neural Networks*, 15(1):66–76.

[Rajih et al., 2008] Rajih, M., Comon, P., and Harshman, R. A. (2008). Enhanced line search: A novel method to accelerate PARAFAC. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1128–1147.

[Rashid and Yu, 2012] Rashid, M. M. and Yu, J. (2012). A new dissimilarity method integrating multidimensional mutual information and independent component analysis for non-Gaussian dynamic process monitoring. *Chemometrics and Intelligent Laboratory Systems*, 115:44–58.

[Rockafellar, 1973] Rockafellar, R. T. (1973). The multiplier method of hestenes and powell applied to convex programming. *Journal of Optimization Theory and Applications*, 12(6):555–562.

[Röemer and Haardt, 2013] Röemer, F. and Haardt, M. (2013). A semi-algebraic framework for approximate CP decompositions via simultaneous matrix diagonalizations (SECSI). *Signal Processing*, 93(9):2722–2738.

[Royer, 2013] Royer, J.-P. (2013). *Identification aveugle de mélanges et décomposition canonique de tenseurs : application à l'analyse de l'eau.* PhD thesis, Université Nice Sophia Antipolis.

[Royer et al., 2011] Royer, J.-P., Thirion-Moreau, N., and Comon, P. (2011). Computing the polyadic decomposition of nonnegative third order tensors. *Signal Processing*, 91(9):2159–2171.

[Sajda et al., 2004] Sajda, P., Du, S., Brown, T. R., Stoyanova, R., Shungu, D. C., Mao, X., and Parra, L. C. (2004). Nonnegative matrix factorization for rapid recovery of constituent spectra in magnetic resonance chemical shift imaging of the brain. *IEEE Transactions on Medical Imaging*, 23(12):1453–1465.

[Sandler and Lindenbaum, 2011] Sandler, R. and Lindenbaum, M. (2011). Nonnegative matrix factorization with Earth mover's distance metric for image analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1590–1602.

[Shao et al., 2009] Shao, X., Liu, Z., and Cai, W. (2009). Extraction of chemical information from complex analytical signals by a non-negative independent component analysis. *Analyst*, 134:2095–2099.

[Sherman and Morrison, 1950] Sherman, J. and Morrison, W. J. (1950). Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, 21(1):124–127.

[Shor, 1985] Shor, N. Z. (1985). *Minimization Methods for Non-Differentiable Functions*, volume 3 of *Springer Series in Computational Mathematics*. Springer-Verlag. Translated from the Russian by Kiwiel, K. C. and Ruszczyński, A.

[Sidiropoulos and Bro, 2000] Sidiropoulos, N. D. and Bro, R. (2000). On the uniqueness of multilinear decomposition of n-way arrays. *Journal of Chemometrics*, 14:229–239.

[Slapak and Yeredor, 2011] Slapak, A. and Yeredor, A. (2011). "weighting for more": Enhancing characteristic-function based ICA with asymptotically optimal weighting. *Signal Processing*, 91(8):2016–2027.

[Smaragdis and Brown, 2003] Smaragdis, P. and Brown, J. C. (2003). Non-negative matrix factorization for polyphonic music transcription. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 177–180, New Paltz, NY.

[Smaragdis et al., 2014] Smaragdis, P., Févotte, C., Mysore, G., Mohammadiha, N., and Hoffman, M. (2014). Static and dynamic source separation using nonnegative factorizations: A unified view. *IEEE Signal Processing Magazine*, 31(3):66–75.

[Sørensen et al., 2009] Sørensen, M., Comon, P., Icart, S., and Deneire, L. (2009). Approximate tensor diagonalization by invertible transforms. In *EUSIPCO'09, Proceedings of the 17th European Signal Processing Conference*, pages 500–504.

[Souloumiac, 2009] Souloumiac, A. (2009). Nonorthogonal joint diagonalization by combining Givens and hyperbolic rotations. *IEEE Transactions on Signal Processing*, 57(6):2222–2231.

[Sun and Févotte, 2014] Sun, D. L. and Févotte, C. (2014). Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence. In *ICASSP'14, Proceedings of 2014 IEEE International Conference on Acoustics, Speech and Signal Processing*, Florence, Italy.

[Sun and Xin, 2012] Sun, Y. and Xin, J. (2012). Nonnegative sparse blind source separation for NMR spectroscopy by data clustering, model reduction, and l_1 minimization. *SIAM Journal on Imaging Sciences*, 5(3):886–911.

[Szabo de Edelenyi et al., 2005] Szabo de Edelenyi, F., Simonetti, A. W., Postma, G., Huo, R., and Buydens, L. M. C. (2005). Application of independent component analysis to h-1 MR spectroscopic imaging exams of brain tumours. *Analytica Chimica Acta*, 544(1–2):36–46.

[Ten Berge and Sidiropoulos, 2002] Ten Berge, J. M. F. and Sidiropoulos, N. D. (2002). On uniqueness in CANDECOMP/PARAFAC. *Psychometrika*, 67(3):399–409.

[Tichavský and Yeredor, 2009] Tichavský, P. and Yeredor, A. (2009). Fast approximate joint diagonalization incorporating weight matrices. *IEEE Transactions on Signal Processing*, 57(3):878–891.

[Todros and Tabrikian, 2010] Todros, K. and Tabrikian, J. (2010). QML-based joint diagonalization of positive-definite Hermitian matrices. *IEEE Transactions on Signal Processing*, 58(9):4656–4673.

[Tonazzini et al., 2004] Tonazzini, A., Bedini, L., and Salerno, E. (2004). Independent component analysis for document restoration. *International Journal on Document Analysis and Recognition*, 7:17–27.

[Tonazzini et al., 2010] Tonazzini, A., Gerace, I., and Martinelli, F. (2010). Multichannel blind separation and deconvolution of images for document analysis. *IEEE Transactions on Image Processing*, 19(4):912–925.

[Tonazzini et al., 2007] Tonazzini, A., Salerno, E., and Bedini, L. (2007). Fast correction of bleed-through distortion in grayscale documents by a blind source separation technique. *International Journal on Document Analysis and Recognition*, 10(1):17–25.

[Trainini et al., 2010] Trainini, T., Li, X.-L., Moreau, E., and Adalı, T. (2010). A relative gradient algorithm for joint decompositions of complex matrices. In *EUSIPCO'10, Proceedings of the 18th European Signal Processing Conference*, pages 1073–1076, Aalborg, Denmark.

[Trainini and Moreau, 2011] Trainini, T. and Moreau, E. (2011). A least squares algorithm for global joint decomposition of complex matrix sets. In *CAMSAP'11, Proceedings of 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, pages 313–316, San Juan, Puerto Rico.

[Trainini and Moreau, 2012] Trainini, T. and Moreau, E. (2012). Variations around gradient like algorithms for joint diagonalization of hermitian matrices. In *EUSIPCO'12, Proceedings of the 20th European Signal Processing Conference*, pages 280–284, Bucharest.

[Tropp, 2003] Tropp, J. A. (2003). Literature survey: Nonnegative matrix factorization. Technical report, University of Texas at Asutin.

[Vaidyanathan, 1993] Vaidyanathan, P. P. (1993). *Multirate Systems and Filter Banks*. PTR Prentice Hall, Englewood Cliffs, New Jersey.

[Valdés-Sosa et al., 2009] Valdés-Sosa, P. A., Vega-Hernández, M., Sánchez-Bornot, J. M., Martínez-Montes, E., and Bobes, M. A. (2009). EEG source imaging with spatio-temporal tomographic nonnegative independent component analysis. *Human Brain Mapping*, 30(6):1898–1910.

[Van Benthem and Keenan, 2004] Van Benthem, M. H. and Keenan, M. R. (2004). Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems. *Journal of Chemometrics*, 18(10):441–450.

[Van der Veen, 2001] Van der Veen, A. J. (2001). Joint diagonalization via subspace fitting techniques. In *ICASSP'01, Proceedings of 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 2773–2776.

[Vasiloglou et al., 2009] Vasiloglou, N., Gray, A. G., and Anderson, D. V. (2009). Non-negative matrix factorization, convexity and isometry. In *Proceedings of SIAM Data Mining Conference*.

[Vía et al., 2011] Vía, J., Palomar, D. P., Vielva, L., and Santamaría, I. (2011). Quaternion ICA from second-order statistics. *IEEE Transactions on Signal Processing*, 59(4):1586–1600.

[Vollgraf and Obermayer, 2006] Vollgraf, R. and Obermayer, K. (2006). Quadratic optimization for simultaneous matrix diagonalization. *IEEE Transactions on Signal Processing*, 54(9):3270–3278.

[Wang and Li, 2010] Wang, F. and Li, P. (2010). Efficient nonnegative matrix factorization with random projections. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 281–292.

[Wang et al., 2006] Wang, F. X., Liu, Z. K., and Zhang, J. (2006). A new joint diagonalization algorithm with application in blind source separation. *IEEE Signal Processing Letters*, 13(1):41–44.

[Wang et al., 2007] Wang, F. X., Liu, Z. K., and Zhang, J. (2007). Nonorthogonal joint diagonalization algorithm based on trigonometric parameterization. *IEEE Transactions on Signal Processing*, 55(11):5299–5308.

[Wang et al., 2010] Wang, F. Y., Chi, C. Y., Chan, T. H., and Wang, Y. (2010). Nonnegative least-correlated component analysis for separation of dependent sources by volume maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):875–888.

[Wang et al., 2013a] Wang, L., Albera, L., Kachenoura, A., Shu, H. Z., and Senhadji, L. (2013a). Nonnegative joint diagonalization by congruence based on LU matrix factorization. *IEEE Signal Processing Letters*, 20(8):807–810.

[Wang et al., 2014a] Wang, L., Albera, L., Kachenoura, A., Shu, H. Z., and Senhadji, L. (2014a). Canonical polyadic decomposition of 3rd order semi-nonnegative semi-symmetric tensors using LU and QR matrix factorizations. Accepted for publication in *EURASIP Journal on Advances in Signal Processing*.

[Wang et al., 2014b] Wang, L., Albera, L., Kachenoura, A., Shu, H. Z., and Senhadji, L. (2014b). CP decomposition of semi-nonnegative semi-symmetric tensors based on QR matrix factorization. In *SAM'14, Proceedings of the Eighth IEEE Sensor Array and Multichannel Signal Processing Workshop*, pages 449–452, A Coruna, Spain. (invited paper).

[Wang et al., 2014c] Wang, L., Albera, L., Kachenoura, A., Shu, H. Z., and Senhadji, L. (2014c). Parameterization of the inverse of the nonnegative matrix for nonnegative joint diagonalization. In preparation for submission to *IEEE Transactions on Signal Processing*.

[Wang et al., 2012] Wang, L., Albera, L., Shu, H. Z., and Senhadji, L. (2012). Decomposition of semi-nonnegative semi-symmetric three-way tensors based on LU matrix factorization. In *SIAM 2012 Conference on Applied Linear Algebra*, Valencia, Spain.

[Wang et al., 2013b] Wang, L., Albera, L., Shu, H. Z., and Senhadji, L. (2013b). A new Jacobi-like nonnegative joint diagonalization by congruence. In *EUSIPCO'13, Proceedings of the XXI European Signal Processing Conference*, pages 1–5, Marrakech, Morocco.

[Wang et al., 2014d] Wang, L., Kachenoura, A., Albera, L., Karfoul, A., Shu, H. Z., and Senhadji, L. (2014d). Nonnegative compression for semi-nonnegative independent component analysis. In *SAM'14, Proceedings of the Eighth IEEE Sensor Array and Multichannel Signal Processing Workshop*, pages 81–84, A Coruna, Spain.

[Wang et al., 2005] Wang, W. W., Sanei, S., and Chambers, J. A. (2005). Penalty function-based joint diagonalization approach for convolutive blind separation of non-stationary sources. *IEEE Transactions on Signal Processing*, 53(5):1654–1669.

[Wang and Zhang, 2013] Wang, Y. X. and Zhang, Y. J. (2013). Nonnegative matrix factorization: A comprehensive review. *IEEE Transactions on Knowledge and Data Engineering*, 25(6):1336–1353.

[Wax and Sheinvald, 1997] Wax, M. and Sheinvald, J. (1997). A least-squares approach to joint diagonalization. *IEEE Signal Processing Letters*, 4(2):52–53.

[Winther and Petersen, 2007] Winther, O. and Petersen, K. B. (2007). Bayesian independent component analysis: Variational methods and non-negative decompositions. *Digital Signal Processing*, 17(5):858–872.

[Xu et al., 2011] Xu, X. F., Feng, D. Z., and Zheng, W. X. (2011). A fast algorithm for nonunitary joint diagonalization and its application to blind source separation. *IEEE Transactions on Signal Processing*, 59(7):3457–3463.

[Xu et al., 2012] Xu, Y. Y., Yin, W. T., Wen, Z. W., and Zhang, Y. (2012). An alternating direction algorithm for matrix completion with nonnegative factors. *Frontiers of Mathematics in China*, 7(2):365–384.

[Xue et al., 2009] Xue, Y., Wang, Y., and Yang, J. (2009). Independent component analysis based on gradient equation and kernel density estimation. *Neurocomputing*, 72(7–9):1597–1604.

[Yang, 1997] Yang, H. H. (1997). Adaptive online learning algorithms for blind separation: Maximum entropy and minimum mutual information. *Neural Computation*, 9(7):1457–1482.

[Ye, 2006] Ye, M. (2006). Global convergence analysis of a discrete time nonnegative ICA algorithm. *IEEE Transactions on Neural Networks*, 17(1):253–256.

[Ye et al., 2006] Ye, M., Fan, X. Q., and Liu, Q. H. (2006). Monotonic convergence of a nonnegative ICA algorithm on Stiefel manifold. In *Neural Information Processing, Lecture Notes in Computer Science*, volume 4232, pages 1098–1106.

[Yeredor, 2000] Yeredor, A. (2000). Blind separation of gaussian sources via second-order statistics with asymptotically optimal weighting. *IEEE Signal Processing Letters*, 7(7):197–200.

[Yeredor, 2002] Yeredor, A. (2002). Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation. *IEEE Transactions on Signal Processing*, 50(7):1545–1553.

[Yeredor, 2005] Yeredor, A. (2005). On using exact joint diagonalization for noniterative approximate joint diagonalization. *IEEE Signal Processing Letters*, 12(9):645–648.

[Yeredor, 2012] Yeredor, A. (2012). Multiple-snapshots BSS with general covariance structures: A partial maximum likelihood approach involving weighted joint diagonalization. *Signal Processing*, 92(8):1832–1843.

[Yeredor et al., 2012] Yeredor, A., Song, B., Röemer, F., and Haardt, M. (2012). A sequentially drilled joint congruence (SeDJoCo) transformation with applications in blind source separation and multiuser MIMO systems. *IEEE Transactions on Signal Processing*, 60(6):2744–2757.

[Yeredor et al., 2004] Yeredor, A., Ziehe, A., and Müller, K.-R. (2004). Approximate joint diagonalization using a natural gradient approach. In *Independent Component Analysis and Blind Signal Separation, Lecture Notes in Computer Science*, volume 3195, pages 89–96.

[Yokoya et al., 2014] Yokoya, N., Chanussot, J., and Iwasaki, A. (2014). Nonlinear unmixing of hyperspectral data using semi-nonnegative matrix factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 52(2):1430–1437.

[Yu et al., 2013] Yu, X. C., Hu, D., and Xu, J. D. (2013). Non-negative independent component analysis and its application. In *Blind Source Separation: Theory and Applications*, chapter 11, pages 167–182. John Wiley & Sons.

[Yu et al., 2012] Yu, X. C., Ni, F., Long, S. L., and Pei, W. J. (2012). Remote sensing image fusion based on integer wavelet transformation and ordered nonnegative independent component analysis. *GIScience and Remote Sensing*, 49(3):364–377.

[Yuan and Oja, 2004] Yuan, Z. J. and Oja, E. (2004). A FastICA algorithm for non-negative independent component analysis. In *Independent Component Analysis and Blind Signal Separation, Lecture Notes in Computer Science*, volume 3195, pages 1–8.

[Zafeiriou et al., 2006] Zafeiriou, S., Tefas, A., Buciu, I., and Pitas, I. (2006). Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification. *IEEE Transactions on Neural Networks*, 17(3):683–695.

[Zarzoso and Comon, 2010] Zarzoso, V. and Comon, P. (2010). Robust independent component analysis by iterative maximization of the kurtosis contrast with algebraic optimal step size. *IEEE Transactions on Neural Networks*, 21(2):248–261.

[Zarzoso et al., 2000] Zarzoso, V., Millet-Roig, J., and Nandi, A. (2000). Fetal ECG extraction from maternal skin electrodes using blind source separation and adaptive noise cancellation techniques. In *Computers in Cardiology*, volume 27, pages 431–434, Cambridge, MA.

[Zdunek and Cichocki, 2007] Zdunek, R. and Cichocki, A. (2007). Nonnegative matrix factorization with constrained second-order optimization. *Signal Processing*, 87(8):1904–1916.

[Zeng and Feng, 2014] Zeng, T. J. and Feng, Q. Y. (2014). Non-orthogonal joint diagonalization algorithm based on hybrid trust region method and its application to blind source separation. *Neurocomputing*, 133:280–294.

[Zhang et al., 2009] Zhang, W. T., Liu, N., and Lou, S. T. (2009). Joint approximate diagonalization using bilateral rank-reducing Householder transform with application in blind source separation. *Chinese Journal of Electronics*, 18(3):471–476.

[Zhang and Lou, 2013] Zhang, W. T. and Lou, S. T. (2013). A recursive solution to nonunitary joint diagonalization. *Signal Processing*, 93(1):313–320.

[Zheng et al., 2006] Zheng, C. H., Huang, D. S., Sun, Z. L., Lyu, M. R., and Lok, T. M. (2006). Nonnegative independent component analysis based on minimizing mutual information technique. *Neurocomputing*, 69(7–9):878–883.

[Zhou et al., 2012] Zhou, G. X., Cichocki, A., and Xie, S. L. (2012). Fast nonnegative matrix/tensor factorization based on low-rank approximation. *IEEE Transactions on Signal Processing*, 60(6):2928–2940.

[Zhou et al., 2014] Zhou, G. X., Cichocki, A., Zhao, Q., and Xie, S. L. (2014). Nonnegative matrix and tensor factorizations : An algorithmic perspective. *IEEE Signal Processing Magazine*, 31(3):54–65.

[Zhou et al., 2009] Zhou, G. X., Xie, S., Yang, Z., and Zhang, J. (2009). Nonorthogonal approximate joint diagonalization with well-conditioned diagonalizers. *IEEE Transactions on Neural Networks*, 20(11):1810–1819.

[Ziehe et al., 2004] Ziehe, A., Laskov, P., Nolte, G., and Muller, K.-R. (2004). A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation. *Journal of Machine Learning Research*, 5:777–800.

VU :

Le Directeur de Thèse
(Nom et Prénom)

VU :

Le Responsable de l'École Doctorale

VU pour autorisation de soutenance

Rennes, le

Le Président de l'Université de Rennes 1

Guy CATHELINEAU

VU après soutenance pour autorisation de publication :

☐ Avec modifications ☐ Sans modifications

Le Président de Jury,
(Nom et Prénom)

Si modifications à faire, validation par le directeur de thèse du manuscrit corrigé avec attestation