Blind Source Separation in Nonlinear Mixtures

Bahram Ehsandoust

Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Préparée dans le cadre d'une co-tutelle entre **la Communauté Université Grenoble Alpes et Université de Technologie de Sharif**

Spécialité : **Signal, Image, Parole, Télécommunication (SIPT)**

Arrêté ministériel : le 6 janvier 2005 - 7 août, 2006

Présentée par

Bahram EHSANDOUST

Thèse dirigée par **Christian Jutten** et **Massoud Babaie-Zadeh**

et co-encadrée par **Bertrand Rivet**

préparée au sein des **Laboratoires Grenoble Images Parole Signal Automatique (GIPSA) et Digital Signal Processing (DSP)**

dans **les École Doctorale d'Electronique, Electrotechnique, Automatique, Traitement du Signal (EEATS) et le Département d'Ingénierie Électrique**

Séparation de Sources Dans les Mélanges non Linéaires

Thèse soutenue publiquement le **30 avril 2018**,

devant le jury composé de :

M. Farrokh Marvasti
Professeur, Université de Technologie de Sharif, Président

M. Yannick Deville
Professeur, Université Paul Sabatier Toulouse 3, Rapporteur

M. Reza Sameni
HDR, Université de Shiraz, Rapporteur

M. Mohammad Bagher Shamsollahi
Professeur, Université de Technologie de Sharif, Examinateur

M. Hamid Soltanian-Zadeh
Professeur, Université de Téhéran, Examinateur

M. Christian Jutten
Professeur, Université Grenoble Alpes, Directeur de thèse

M. Massoud Babaie-Zadeh
Professeur, Université de Technologie de Sharif, Co-Directeur de thèse

M. Bertrand Rivet
Maître de conférences, Grenoble INP, Co-Encadrant de thèse

Acknowledgments

Foremost, I would like to express my sincere gratitude to my advisors, Prof. Christian Jutten, Prof. Massoud Babaie-Zadeh, and Dr. Bertrand Rivet, and for their continuous support of my Ph.D. study and research, for their endless patience, motivation, enthusiasm, and immense knowledge. The large-hearted, caring, welcoming, supportive Christian: I will always remember, and hopefully learn from your kind patronage, not only in my studies, but also in all aspects of my life. The adroit, supportive, expert Massoud: I am honored to work with you for several years of my life, and forever grateful to you for your continued advice and inspiration, and introducing and recommending me to the people of GIPSA lab which brought me the opportunity of working there. The hardworking, smart, cordial Bertrand: please accept my heartfelt thanks for your enlightening guidance which gave direction to the work, and your constant support and impeccable expertise during all these years. I could not have ever imagined having better advisors and mentors for my Ph.D. study.

Beside my advisors, I would like to thank the rest of my thesis committee: Prof. Farrokh Marvasti, Prof. Yannick Deville, Dr. Reza Sameni, Prof. Mohammad Bagher Shamsollahi and Prof. Hamid Soltanian-Zadeh, for their encouragement, insightful comments, deep questions and candid critical insights. In particular, I am grateful to Mohammad Bagher for what I have learned from his special friendly character and his genuine helps.

This work was funded by the European project 2012-ERC-Adv-320684 CHESS. I appreciate the European Union for funding my Ph.D. project, especially France, the welcoming French people and their rich culture. They

generously accepted and integrated me among them, letting me have a pleasant memorable fantastic experience living there, in the space span of liberty, equality, and brotherhood. My sincere thanks also goes to Mme. Christine Richard, who voluntarily taught me French and supported me like a merciful mother.

I would also like to thank my always interested, encouraging and enthusiastic friends and colleagues. They were fundamental in supporting me during thesis stress and difficult moments, and I also thank them for contributing to the development of my research. With a special mention to Iman Mohammadi, Omid Zobeiri and Arash Khatibi, who were always keen to know what I was doing and how I was proceeding, although it is likely that not all of them have grasped what it was all about!

Finally, my sincere gratitude goes to my eternal life-coaches: my parents Masoud Ehsandoust and Simin Delbari, and my sister Parisa Ehsandoust, for all their spiritual support throughout my life and my study. And last but by no means least, I owe it all to my lovely wife; Saman Noorzadeh, my friend, comrade, ambition, future and aspiration. Saman: thanks for providing me my life, through your moral and emotional support.

Unacknowledgement On 29 Jan. 2017, the U.S. Immigration Service under the president Trump administration canceled my visa appointment of 1 Feb. 2017, in compliance with the U.S. Presidential Executive Order on Protecting the Nation from Terrorist Attacks, which suspended visa issuance for aliens of Iran and a few other nations. This prevented me from participating in the international ICASSP 2017 conference, i.e. the most important gathering of the signal processing community. Other than the fees for the already-planned travel to the U.S. and for the conference, they did not even refund the visa application fee of 160 USD. Upon my refund request, they surprisingly responded “Under the current U.S. Department of State policy, U.S visa fees are non-refundable and non-transferable”. They continued: “Do not attend. You will not be permitted entry to the Embassy/Consulate.”

Abstract

Blind Source Separation (BSS) is a technique for estimating individual source components from their mixtures at multiple sensors, where the mixing model is unknown. Although it has been mathematically shown that for linear mixtures, under mild conditions, mutually independent sources can be reconstructed up to accepted ambiguities, there is not such theoretical basis for general nonlinear models. This is why there are relatively few results in the literature in this regard in the recent decades, which are focused on specific structured nonlinearities.

In the present study, the problem is tackled using a novel approach utilizing temporal information of the signals. The original idea followed in this purpose is to study a linear time-varying source separation problem deduced from the initial nonlinear problem by derivations. It is shown that already-proposed counter-examples showing inefficiency of Independent Component Analysis (ICA) for nonlinear mixtures, loose their validity, considering independence in the sense of stochastic processes instead of simple random variables. Based on this approach, both nice theoretical results and algorithmic developments are provided. Even though these achievements are not claimed to be a mathematical proof for the separability of nonlinear mixtures, it is shown that given a few assumptions, which are satisfied in most practical applications, they are separable.

Moreover, nonlinear BSS for two useful sets of source signals is also addressed: (1) spatially sparse sources and (2) Gaussian processes. Distinct BSS methods are proposed for these two cases, each of which has been widely studied in the literature and has been shown to be quite beneficial in modeling many practical applications.

Concerning Gaussian processes, it is demonstrated that not all nonlinear mappings can preserve Gaussianity of the input. For example being restricted to polynomial functions, the only Gaussianity-preserving function is linear. This idea is utilized for proposing a linearizing algorithm which, cascaded by a conventional linear BSS method, separates polynomial mixtures

of Gaussian processes.

Concerning spatially sparse sources, it is shown that spatially sparse sources make manifolds in the observations space, and can be separated once the manifolds are clustered and learned. For this purpose, multiple manifold learning problem has been generally studied, whose results are not limited to the proposed BSS framework and can be employed in other topics requiring a similar issue.

Keywords— Blind Source Separation, Independent Component Analysis, Nonlinear Signals Processing, Nonlinear Regression, Nonlinear Mixtures, Nonlinear Distortion, Gaussian Processes, Polynomial Mappings, Sparse Signals, Manifold Learning

Résumé

La séparation aveugle de sources aveugle (BSS) est une technique d'estimation des différents signaux observés au travers de leurs mélanges à l'aide de plusieurs capteurs, lorsque le mélange et les signaux sont inconnus. Bien qu'il ait été démontré mathématiquement que pour des mélanges linéaires, sous des conditions faibles, des sources mutuellement indépendantes peuvent être estimées, il n'existe dans de résultats théoriques généraux dans le cas de mélanges non-linéaires. La littérature sur ce sujet est limitée à des résultats concernant des mélanges non linéaires spécifiques.

Dans la présente étude, le problème est abordé en utilisant une nouvelle approche utilisant l'information temporelle des signaux. L'idée originale conduisant à ce résultat, est d'étudier le problème de mélanges linéaires, mais variant dans le temps, déduit du problème non linéaire initial par dérivation. Il est démontré que les contre-exemples déjà présentés, démontrant l'inefficacité de l'analyse par composants indépendants (ACI) pour les mélanges non-linéaires, perdent leur validité, considérant l'indépendance au sens des processus stochastiques, au lieu de l'indépendance au sens des variables aléatoires. Sur la base de cette approche, de bons résultats théoriques et des développements algorithmiques sont fournis. Bien que ces réalisations ne soient pas considérées comme une preuve mathématique de la séparabilité des mélanges non-linéaires, il est démontré que, compte tenu de quelques hypothèses satisfaites dans la plupart des applications pratiques, elles sont séparables.

De plus, les BSS non-linéaires pour deux ensembles utiles de signaux sources sont également traités, lorsque les sources sont (1) spatialement parcimonieuses, ou (2) des processus Gaussiens. Des méthodes BSS particulières sont proposées pour ces deux cas, dont chacun a été largement étudié dans la littérature qui correspond à des propriétés réalistes pour de nombreuses applications pratiques.

Dans le cas de processus Gaussiens, il est démontré que toutes les applications non-linéaires ne peuvent pas préserver la gaussianité de l'entrée,

cependant, si on restreint l'étude aux fonctions polynomiales, la seule fonction préservant le caractère gaussiens des processus (signaux) est la fonction linéaire. Cette idée est utilisée pour proposer un algorithme de linéarisation qui, en cascade par une méthode BSS linéaire classique, sépare les mélanges polynomiaux de processus Gaussiens.

En ce qui concerne les sources parcimonieuses, on montre qu'elles constituent des variétés distinctes dans l'espaces des observations et peuvent être séparées une fois que les variétés sont apprises. À cette fin, plusieurs problèmes d'apprentissage multiple ont été généralement étudiés, dont les résultats ne se limitent pas au cadre proposé du SRS et peuvent être utilisés dans d'autres domaines nécessitant un problème similaire.

Mots clés— Séparation des sources aveugles, Analyse en composantes indépendantes, Traitement des signaux non-linéaires, Régression non-linéaire, Mélanges non-linéaires, Distorsion non-linéaire, Processus Gaussiens, Fonctions polynomiales, Signaux parcimonieux, Apprentissage sur variétés

Contents

List of Figures

List of Tables

List of Abbreviations

AATVL Adaptive Algorithm for Time-Variant Linear mixtures

BATIN Batch Algorithm for Time-Invariant Nonlinear mixtures

BSS Blind Source Separation

EEG Electroencephalogram

EASI Equivariant Adaptive Separation via Independence

ELMM Extended Linear Mixing Model

FECG Fetal Electrocardiogram

GP Gaussian Process

HT Hard Thresholding

ICA Independent Component Analysis

iid Independent and Identically Distributed

IVA Independent Vector Analysis

LMM Linear Mixing Model

LTI Linear Time-Invariant

LVA Latent Variable Analysis

MEG Magnetoencephalogram

MIMO Multiple Input Multiple Output

N-ENF Normalized Error of Nonlinear Fit

pdf Probability Density Function

PNL Post Non-Linear

RV Random Variable

RMS Root Mean Squared

SP Stochastic Process

1 Introduction

The Blind Source Separation (BSS) problem was firstly introduced in 1980's, and since then, it has been thoroughly studied in the signal processing community [Hérault and Jutten, 1986]. Roughly speaking, in this problem several source signals are mixed through an unknown mixing function to make a number (probably not the same number as the sources) of observation signals. The goal is to reconstruct the sources having access only to the observations, i.e. knowing neither the sources nor the mixing model.

BSS problem is formally described as follows. At each time (more generally, sample) t let us consider m observations $x_i(t)$, $i = 1, \ldots, m$, which are unknown time-invariant functions $f_i(\cdot)$ of unknown sources $s_j(t)$, $j = 1, \ldots, n$. For $t = 1, \ldots, T$ measurements, we can mathematically express the model as

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_m(t) \end{bmatrix} = \begin{bmatrix} f_1(\mathbf{s}(t)) \\ f_2(\mathbf{s}(t)) \\ \vdots \\ f_m(\mathbf{s}(t)) \end{bmatrix} = \mathbf{f}(\mathbf{s}(t)), \qquad t = 1, \ldots, T \tag{1.1}$$

where $\mathbf{x}(t) = [x_1(t), ..., x_m(t)]^\dagger$ ($\dagger$ stands for matrix transposition) and $\mathbf{s}(t) = [s_1(t), ..., s_n(t)]^\dagger$ represent the observation and source vectors, respectively, and $\mathbf{f}(\cdot)$ denotes a function from $\mathbb{R}^n$ to $\mathbb{R}^m$. The goal is to find a separating system $\mathbf{g}(\mathbf{x})$ reconstructing the sources based only on the observations $\mathbf{x}(t)$ knowing neither the sources nor the mixing function $\mathbf{f}$.

The problem model is depicted in Fig. 1.1. In this model, we generally expect each of the elements of $\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t))$ to be a function of only one of

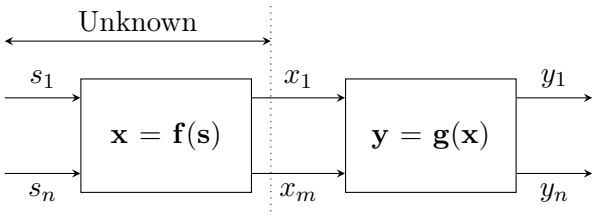


Figure 1.1: Nonlinear BSS problem basic model

the source signals (and each source signal appears in only one entry of $\mathbf{y}(t)$).

The problem is generally ill-posed, but it has been shown that assuming $\mathbf{f}$ has some particular structure, and/or given some statistical properties of the sources, it can be solved to some extent and the sources can be reconstructed with ambiguities in their amplitude and their order. The book [Comon and Jutten, 2010] provides a comprehensive survey on different structures and proposed algorithms. The key idea to perform separation is trying to recover some characteristics of the sources by estimating a mapping on the observations able to inverse $\mathbf{f}$. Mostly these characteristics include one of the "non-properties" (a word borrowed from [Mei et al., 2009]); e.g. non-dependence (independence), non-Gaussianity [Comon, 1994], non-stationarity [Parra and Spence, 2000], non-whiteness [Buchner et al., 2003] and non-negativity [Cichocki et al., 2006].

The simplest form of the problem is when the mixture model is instantaneous *linear* and the number of the sources is equal to the number of the observations, i.e. $n = m$, so that (1.1) becomes $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$ where $\mathbf{A}$ is an unknown mixing matrix. Even this problem is still ill-posed and that priors on sources are mandatory for it to be solved. Source independence is the earliest example of such prior which was used in [Hérault and Jutten, 1986, Comon, 1994] for introducing the concept of Independent Component Analysis (ICA). The independence employed in ICA is in the sense of random variables assuming that each source consists of Independent and Identically Distributed (iid) samples, i.e. without taking care of the sample order.

It should be recalled that if two random variables U and V are mutually

independent, the joint probability density function (pdf) of them $\rho_{U,V}(u,v)$ factorizes as

$$\rho_{U,V}(u,v) = \rho_U(u)\rho_V(v) \tag{1.2}$$

where $\rho_U(u)$ and $\rho_V(v)$ are the marginal pdf's of U and V respectively.

On the other hand, two stochastic processes $U(t)$ and $V(t)$ are said to be mutually independent iff they are mutually independent for any sequence of time instants, i.e. for any positive integer $r < \infty$ and any sequence $\mathbf{t} = (t_1, \ldots, t_r)$, random vectors $\mathbf{U_t} = (U(t_1), \ldots, U(t_r))$ and $\mathbf{V_t} = (V(t_1), \ldots, V(t_r))$ are mutually independent, i.e.

$$\rho_{\mathbf{U_t},\mathbf{V_t}}(u(t_1), \ldots, u(t_r), v(t_1), \ldots, v(t_r)) = \rho_{\mathbf{U_t}}(u(t_1), \ldots, u(t_r))\rho_{\mathbf{V_t}}(v(t_1), \ldots, v(t_r)) \tag{1.3}$$

where $\rho_{\mathbf{U_t},\mathbf{V_t}}(\cdot,\cdot)$, $\rho_{\mathbf{U_t}}(\cdot)$ and $\rho_{\mathbf{V_t}}(\cdot)$ denote the joint pdf of $(\mathbf{U_t}, \mathbf{V_t})$ and the marginal pdf's of $\mathbf{U_t}$ and $\mathbf{V_t}$, respectively. Accordingly, the two notions: random variable (RV) independence and stochastic process (SP) independence, should be distinguished.

Nevertheless, it is shown that in linear BSS problem, if the sources are mutually RV independent, i.e. ignoring the sample dependence of each source, they can be blindly reconstructed up to ambiguities in their scale and their order (for the reference and more details, refer to the following chapter). However, this result cannot be generalized to the general nonlinear BSS problem. Indeed it is shown by counter-examples, e.g. [Hosseini and Jutten, 2003,Babaie-Zadeh, 2002], that ICA in the sense of random variables is not able to separate the sources in nonlinear mixtures (see the following chapter).

For this reason, the general nonlinear BSS problem had been left almost unexplored. However, in this work, novel approaches for performing nonlinear BSS are proposed. The proposed general approaches assume that signals have temporal correlation, i.e. colored, which usually happens in realistic physical signals.

The first proposed approach is based on finding the connection between the linear and nonlinear problem, and discovering circumstances under which it works [Ehsandoust et al., 2017a]. Another approach is based on modeling the sources by Gaussian processes, and approximating the mixture by a polynomial [Ehsandoust et al., 2017b]. In addition, a special case of the problem where sources are assumed to be spatially sparse, is investigated in this work [Ehsandoust et al., 2016]. For each of these methods, interesting theoretical results along with separating algorithms are provided.

Thesis Overview

In the following, the chapters of this document are briefly described.

Chapter 2

In Chapter 2 the state of the art in BSS, especially nonlinear BSS is presented. For this purpose, to better clarify the problem, we first review linear BSS techniques and approaches. Then it is shown through counter-examples why conventional approaches are not applicable to nonlinear problems.

Chapter 3

The key innovative idea for tackling general nonlinear BSS problems is presented in Section 3.1. A recent application of this approach in hyperspectral images is also briefly explained. Then the problem and its assumptions are precisely described, and the principal idea is mathematically expressed. In order to shed light on the proposed approach, in Section 3.1.4 it is shown that given a parametric model for the unmixing function, how the parameter would be estimated for the separation. Afterwards, separating algorithms are proposed in details and source reconstruction indeterminacies are discussed. Finally in this chapter, by investigating the simulation results, a discussion is made and directions for future studies are suggested.

Chapter 4

Gaussian Processes have recently attracted a lot of attentions in different fields of signal processing and are shown to be very beneficial in modeling several signals. As a consequence, this chapter is concerned with the nonlinear problem given the sources to follow Gaussian distribution. The problem of interest in this chapter is even more general than nonlinear BSS; in fact, it is questioned under which conditions an unknown nonlinear mixture of Gaussian signals can be blindly transformed to a linear one. Evidently, a nonlinear BSS approach can be structured by such linearizing techniques cascaded by traditional linear BSS methods for Gaussian signals. In this chapter, the theory of the proposed idea is firstly studied, based on which a linearizing algorithm is proposed and simulated.

Chapter 5

This chapter addresses the nonlinear BSS problem conditioned that the sources are spatially sparse. For this purpose, after a summarized review on the related works on linear mixtures, the approach is proposed to be split into two consecutive steps: 1) clustering and multiple manifold learning 2) Separating the sources. Since the first step may have diverse applications in other domains of signal processing and pattern recognition, it is investigated more deeply in Section 5.2.1. Then in Section 5.2.1, the sources are separated and reconstructed up to accepted ambiguities. Like the other chapters, the proposed method is supported by simulation results on synthetic data.

2 State of the Art

Contents

In this chapter, the literature of BSS problem is shortly reviewed. For this purpose, ICA is introduced as the main approach for linear BSS, followed by its different algorithms, applications, and extensions. Then, it is shown why ICA is not able to separate general nonlinear mixtures and which specific nonlinear mappings are shown to be separable.

2.1 Linear BSS and its applications

Assuming the mixing function is linear, the system of Fig. 1.1 is simplified as Fig. 2.1. In this model, given the observations $\mathbf{x}(t)$ for $t = 1, \ldots, T$, the goal is to find matrix $\mathbf{B}$ such that $\mathbf{y}(t) = \mathbf{B}\mathbf{x}(t)$ is the best reconstruction of the unknown sources. The problem is equivalent to factorize the data matrix $\mathbf{X}$ (size $m \times T$) as the product of two matrices $\mathbf{A}$ (size $m \times n$) and $\mathbf{S}$ (size $n \times T$).

The indeterminacy in the problem is any regular matrix $\boldsymbol{\mathcal{M}}$ (size $n \times n$), since $\mathbf{X} = \mathbf{AS} = (\mathbf{A}\boldsymbol{\mathcal{M}})(\boldsymbol{\mathcal{M}}^{-1}\mathbf{S})$. Thus the problem is evidently ill-posed and priors on sources are mandatory for avoiding this unacceptable

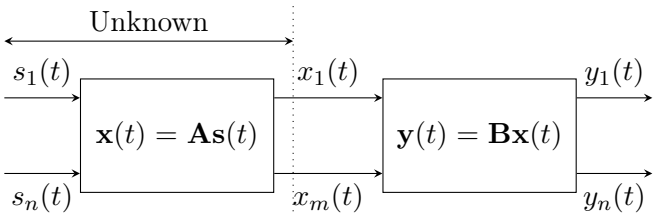


Figure 2.1: Linear BSS problem basic model

indeterminacy and solving the problem. Source independence is an example of such prior.

This problem has been intensively investigated for the past three decades (refer to [Comon and Jutten, 2010]). For linear instantaneous mixtures, where the number of the observations is equal to the number of the sources, a very nice result is that signal separation can be achieved if the sources $s_i(t)$ and $s_j(t)$, for any pair $i \neq j$, are mutually independent random variables [Comon, 1994]. More precisely, [Comon, 1994, theorem 11] says that

Theorem 1. *Let* $\mathbf{s}$ *be a vector with independent components, of which at most one is Gaussian, and whose densities are not reduced to a point-like mass. Let* $\mathbf{C}$ *be an orthogonal* $n \times n$ *matrix and* $\mathbf{y}$ *the vector* $\mathbf{y} = \mathbf{C}\mathbf{s}$*. Then the following three properties are equivalent:*

1. *The components* y_i *are pairwise independent.*
2. *The components* y_i *are mutually independent.*
3. $\mathbf{C} = \mathbf{\Lambda}\mathbf{P}$*,* $\mathbf{\Lambda}$ *diagonal,* $\mathbf{P}$ *permutation.*

This theorem gives the main idea of linear BSS as "find $\mathbf{B}$ such that the components of $\mathbf{y} = \mathbf{B}\mathbf{x}$ are pairwise independent" (see Fig. 2.1). It is thus outstanding to note that SP independence (refer to Chapter 1) is not required in the linear case.

It should be emphasized that according to Theorem 1, without further information, sources can be reconstructed up to a scaling indeterminacy and a change of order. This can be understood by considering the fact that

any matrix $\mathbf{B}$ satisfying $\mathbf{BA} = \mathbf{\Lambda P}$ is an acceptable answer. Therefore, the problem does not contain any information about either the order of the sources or their scale. Accordingly, a "linear copy" of a vector is defined as follows.

Definition Let $\mathbf{s}$ be an n-dimensional vector. $\mathbf{y}$ is called a "linear copy" [Cardoso, 1998] of $\mathbf{s}$ if it has the same dimension as $\mathbf{s}$ and each element y_i of it is one and only one of the elements of $\mathbf{s}$ which is scaled by an arbitrary coefficient. It can be written as

$$\forall\ 1 \leq i \leq n \quad y_i = c_i s_{\tau_i} \tag{2.1}$$

where c_i for $i = 1, \ldots, n$ is a scalar and $(\tau_1, \tau_2, \ldots, \tau_n)$ is a permutation of $(1, 2, \ldots, n)$. □

Many algorithms have been designed based on different approximations of RV independence (refer to Chapter 1), e.g. CoM2 [Comon, 1994], JADE [Cardoso and Souloumiac, 1993], Normalized EASI [Cardoso and Laheld, 1996], HOSVD [De Lathauwer et al., 2000] and FastICA [Hyvärinen, 1999a]. We can also cite AMUSE [Tong et al., 1990, Tong et al., 1991] and SOBI [Belouchrani et al., 1997] which, conversely to previous algorithms assuming iid samples, exploit the assumption that the source samples are not iid, and consider the statistical independence of delayed samples. Afterwards, taking into account any of the mentioned "non-properties", any combination of them, or even some other characteristics such as sparsity, other separation algorithms have been proposed, e.g. INFOMAX [Bell and Sejnowski, 1995] (a thorough study of different methods is provided in [Comon and Jutten, 2010]).

However, the linear instantaneous model is too simple to fit in many practical applications. Therefore, it has to be extended in many directions; e.g.

- considering additive noise

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t), \tag{2.2}$$

- dealing with complex signals (instead of real ones),
- considering the mixture to be convolutive

$$\mathbf{x}(t) = [\mathbf{A}(z)]\mathbf{s}(t) \stackrel{\triangle}{=} \sum_k \mathbf{A}_k \mathbf{s}(t-k), \tag{2.3}$$

- considering over-determined/under-determined cases (when the number of the sources is less/more than the number of the observations)

$$\begin{cases} m > n & \text{over-determined} \\ m = n & \text{determined} \\ m < n & \text{under-determined} \end{cases} . \tag{2.4}$$

There are several works in the literature on each of the mentioned directions: for more details refer to [Comon and Jutten, 2010].

These extensions made BSS applicable to numerous realistic problems, such as LVA (Latent Variable Analysis; e.g. in economics [Kiviluoto and Oja, 1998]), bio-medical signal processing (e.g. separating signals from different sections of the brain in EEG (Electroencephalogram) and MEG (Magnetoencephalography) [Vigário et al., 2000] and extraction of FECG (Fetal Electrocardiogram) [De Lathauwer et al., 1995]), multiple antennas and MIMO (Multiple Input Multiple Output) communications [Li and Liu, 1998], analysis of multi-spectral astronomical images [Nuzillard and Bijaoui, 2000], etc. (more details about all these applications and several other ones can be found in [Hyvärinen et al., 2004]).

2.2 Nonlinear BSS and its applications

In many applications the mixing system of the sources has to be modeled as nonlinear. Hyperspectral imaging [Dobigeon et al., 2014, Golbabaee et al., 2013], remote sensing data [Meganem et al., 2011], determining the concentration of different ions in a combination via smart chemical sensor arrays [Duarte and Jutten, 2014], and removing show-through in scanned doc-

uments [Merrikh-Bayat et al., 2011] are some well-studied examples of such applications.

However, although for linear mixtures, conventional ICA (i.e. based on RV independence) ensures identifiability and separability even for iid sources, it is not sufficient for nonlinear mixtures. In other words, one can find some nonlinear mixtures (with non-diagonal Jacobian) of mutually independent sources which are still mutually independent. In the following, it is shown by counter-examples why RV-based ICA does not work for nonlinear BSS.

As introduced in [Taleb and Jutten, 1999, Hosseini and Jutten, 2003], let us consider two independent iid source signals $s_1(t)$ and $s_2(t)$, whose samples follow the following pdf's

$$\rho_1(s_1(t)) = s_1(t) \times e^{-s_1(t)^2/2} \tag{2.5}$$

$$\rho_2(s_2(t)) = 1/2\pi; \quad 0 \leq s_2(t) < 2\pi \tag{2.6}$$

and the nonlinear transform

$$x_1 = s_1 \times \cos(s_2) \tag{2.7}$$

$$x_2 = s_1 \times \sin(s_2). \tag{2.8}$$

On the other hand, we know that for a bijective and differentiable function $\mathbf{f}$

$$\mathbf{x} = \mathbf{f}(\mathbf{s}) \quad \Rightarrow \quad \rho_{\mathbf{X}}(\mathbf{x}) = \frac{\rho_{\mathbf{S}}(\mathbf{s})}{|\det(\mathbf{J_f}(\mathbf{s}))|} \tag{2.9}$$

where $\det(\mathbf{J_f}(\mathbf{s}))$ is the determinant of the Jacobian matrix of the nonlinear transformation and is defined as

$$\mathbf{J_f}(\mathbf{s}) = \begin{bmatrix} \frac{\partial f_1}{\partial s_1} & \dots & \frac{\partial f_1}{\partial s_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial s_1} & \dots & \frac{\partial f_n}{\partial s_n} \end{bmatrix}. \tag{2.10}$$

After simple calculations, one can easily compute the joint pdf of x_1 and x_2 which factorizes as

$$\rho_{X_1,X_2}(x_1, x_2) = \frac{\rho_{S_1,S_2}(s_1, s_2)}{|\mathbf{J_f}|} = (\frac{1}{\sqrt{2\pi}} e^{-x_1^2/2})(\frac{1}{\sqrt{2\pi}} e^{-x_2^2/2}). \tag{2.11}$$

Therefore, the observations are statistically independent while they are still mixtures of both sources.

Another counter-example was introduced In [Babaie-Zadeh, 2002, Section 3.3], which showed that even for smooth nonlinear mixing functions, source independence (in the sense of random variables) was not a powerful enough criterion for separating the sources. In this example, at each sample t, the sources are mixed nonlinearly as

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \cos\alpha(\mathbf{s}(t)) & -\sin\alpha(\mathbf{s}(t)) \\ \sin\alpha(\mathbf{s}(t)) & \cos\alpha(\mathbf{s}(t)) \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix} \tag{2.12}$$

where $\alpha(\mathbf{s}(t))$ is a differentiable function. In this particular example, the determinant of the Jacobian matrix of the nonlinear transformation is calculated as

$$\mathbf{J_f}(\mathbf{s}) = \begin{bmatrix} \cos\alpha(\mathbf{s}) & -\sin\alpha(\mathbf{s}) \\ \sin\alpha(\mathbf{s}) & \cos\alpha(\mathbf{s}) \end{bmatrix} \begin{bmatrix} 1 - s_2\frac{\partial\alpha(\mathbf{s})}{\partial s_1} & -s_2\frac{\partial\alpha(\mathbf{s})}{\partial s_2} \\ s_1\frac{\partial\alpha(\mathbf{s})}{\partial s_1} & 1 + s_1\frac{\partial\alpha(\mathbf{s})}{\partial s_2} \end{bmatrix} \tag{2.13}$$

$$\Rightarrow \det(\mathbf{J_f}(\mathbf{s})) = 1 + s_1\frac{\partial\alpha(\mathbf{s})}{\partial s_2} - s_2\frac{\partial\alpha(\mathbf{s})}{\partial s_1}. \tag{2.14}$$

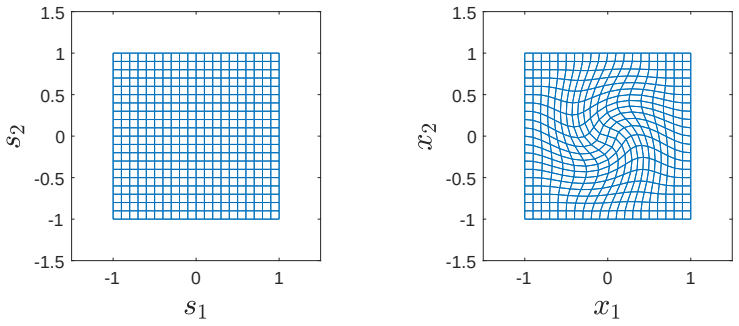
If $\alpha(\mathbf{s}(t))$ is only a function of the norm of the input vector, i.e. $r(t) \stackrel{\triangle}{=} \sqrt{s_1^2(t) + s_2^2(t)}$, (2.14) will be equal to one for any source vector. Consequently

$$\rho_{X_1,X_2}(x_1, x_2) = \frac{1}{|\det(\mathbf{J_f}(\mathbf{s}))|}\rho_{S_1,S_2}(s_1, s_2) = \rho_{S_1,S_2}(s_1, s_2).$$

Particularly, if the source samples are iid and uniformly distributed between -1 and 1, i.e. $\rho_{S_1,S_2}(s_1, s_2) = 0.25$ for $(s_1, s_2) \in [-1, 1] \times [-1, 1]$ and 0 elsewhere, and given

$$\alpha(\mathbf{s}(t)) = \begin{cases} \theta_0(1 - r(t))^{\alpha_0} & \text{if } 0 \le r(t) \le 1 \\ 0 & \text{if } r(t) \ge 1 \end{cases} \tag{2.15}$$

where θ_0 and α_0 are real and integer constants respectively, the observations will also follow a joint uniform distribution as $\rho_{X_1,X_2}(x_1, x_2) = 0.25$ for $(x_1, x_2) \in [-1, 1] \times [-1, 1]$ and 0 elsewhere, which factorizes. Thus the



(a) Source grid-map (b) Mixture grid-map

Figure 2.2: Illustration of the nonlinear mapping of (2.12)

observations are instantaneously mutually independent, even though each of them is a nonlinear mixture of both sources. Fig. 2.2 illustrates this mapping for $\theta_0 = \pi/2$ and $\alpha_0 = 2$.

These counter-examples prove that RV independence is not sufficient for separating nonlinearly mixed signals. As a consequence, except a few dispersed works (e.g. [Blaschke et al., 2007] and [Levin, 2010]), studies in nonlinear BSS were mainly focused on specific mixing models or specific source signals, which were concerned by practical applications and for which RV independence is sufficient for ensuring identifiability and separability.

2.2.1 Specific nonlinear models

There are two main classes of nonlinear models investigated [Deville and Duarte, 2015] and for which ICA leads to source separation under mild conditions.

1. Post-Nonlinear (PNL) [Achard and Jutten, 2005, Taleb and Jutten, 1999, Altmann et al., 2012]: the unknown mixing nonlinear system contains a linear mixture cascaded by component-wise nonlinear distortions as depicted in Fig. 2.3. Convolutive PNL is an extension of this basic model [Babaie-Zadeh, 2002].

2. Bi-Linear (or Linear Quadratic) mixtures [Deville and Hosseini, 2007,

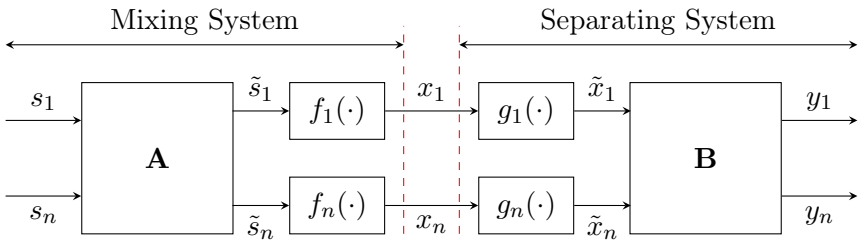


Figure 2.3: PNL problem model

Merrikh-Bayat et al., 2011, Halimi et al., 2011]: each observation is a linear mixture of the sources and their second-order multiplications, i.e. the nonlinear function is linear with respect to each source if the other are constant. This model can be formulated as

$$\forall 1 \leq i \leq n \qquad x_i = \sum_{j=1}^{n} a_{ij} s_j + \sum_{j=1}^{n} \sum_{k=j}^{n} b_{ijk} s_j s_k. \tag{2.16}$$

Conformal mappings [Hyvärinen and Pajunen, 1999] and linear-transformable mappings [Kagan et al., 1973] are two other categories that have been addressed so far and for which RV independence leads to source separation. We have also studied some other specific models (e.g. polynomial mixtures) whose results will come in the following chapters.

2.2.2 General Approach

As stated before, it had been shown by counter-examples that ICA does not work for nonlinear mixtures. So studies on nonlinear mixtures were limited to specific cases where the mixing function and/or the sources are parametrized or they follow an already-known structure.

However, these limitations are mainly due to the fact that the temporal information of the sources is not exploited. For example in [Hosseini and Jutten, 2003] it is shown that even though for each time instant t_0, $x_1(t_0)$ and $x_2(t_0)$ are independent random variables, stochastic processes $x_1(t)$ and $x_2(t)$ might not be independent stochastic processes, and random variables

$x_1(t_0)$ and $x_2(t_0 - 1)$ could be dependent. Taking this fact into account, previous "counter-examples" lose their validity for proving that general nonlinear mixtures are not separable.

Therefore, using a more general definition of independence than RV independence (1.2), but simpler than SP independence (1.3), a more general problem may be addressed without being restricted to any specific mixture or parametric model. Actually there is a similar story in *convolutive* linear mixtures; ICA in the RV sense does not separate the sources, but a more general definition of ICA does. In that case, it is shown that if the signals and their delayed versions are independent, they are separable. Therefore, dealing with nonlinear mixtures, the key idea is that although the mixture is instantaneous, RV independence is not powerful enough to separate the sources. This is the main idea of the Chapter 3.

A similar approach is taken in [Levin, 2010] for "performing BSS for nonlinear mixtures using signal invariants". In that paper, the mixture is modeled as (1.1) where the number of the sources is assumed to be equal to the number of the observations, i.e. $m = n$.

In that work, the source signals are assumed to be independent in the sense of

$$\rho_S(\mathbf{s}, \dot{\mathbf{s}}) = \prod_{k=1}^{n} \rho_k(s_k, \dot{s_k}) \tag{2.17}$$

where " $\dot{}$ " denotes time(or sample)-derivative.

Note that (2.17) is more powerful than RV independence (1.2) used in ICA, but it is simpler than SP independence (1.3). In other words, SP independence results (2.17), and (2.17) results RV independence, but reciprocals do not hold. It is also interesting to note that using time-derivatives implies considering temporal information of signals.

The paper [Levin, 2010], as well as [Levin, 2017], proposes a method for diagonalizing the local correlation matrix of the data's velocity. For this

purpose tensors $C_{kl\ldots}(\mathbf{x})$ are defined as

$$C_{kl\ldots}(\mathbf{x}) \triangleq \frac{\int \rho_X(\mathbf{x},\dot{\mathbf{x}})(\dot{x}_k - \bar{\dot{x}}_k)(\dot{x}_l - \bar{\dot{x}}_l)\ldots d\dot{\mathbf{x}}}{\int \rho_X(\mathbf{x},\dot{\mathbf{x}})d\dot{\mathbf{x}}} \tag{2.18}$$

$$\approx \langle(\dot{x}_k - \bar{\dot{x}}_k)(\dot{x}_l - \bar{\dot{x}}_l)\ldots\rangle_{\mathbf{x}} \tag{2.19}$$

where in (2.19), $\bar{\dot{x}} = \langle\dot{x}\rangle_{\mathbf{x}}$, the bracket denotes the time average over the trajectory's segments in a small neighborhood of $\mathbf{x}$ and "…" denotes possible additional indices on both the left side, and correspondingly, the right side.

For better understanding, let us change the notation of the paper and rewrite (2.18) for second order correlations (only two indices) as

$$\Rightarrow \mathbf{C}^{\mathbf{x}} \triangleq \mathbb{E}\{\dot{\mathbf{x}}\dot{\mathbf{x}}^\dagger\} \tag{2.20}$$

where $\mathbb{E}$ represents the statistical expected value. Note that the superscript $\mathbf{x}$ shows that it is calculated locally.

Now, a linear transformation matrix $\mathbf{M}$

$$\mathbf{y} \triangleq \mathbf{M}\mathbf{x} \quad \Rightarrow \quad \dot{\mathbf{y}} = \mathbf{M}\dot{\mathbf{x}} \tag{2.21}$$

is locally found such that (1) the $\mathbf{M}$-transformed velocity correlations ($\dot{\mathbf{y}}$) are orthonormal and (2) a projection matrix of the forth-order correlation tensor of $\dot{\mathbf{x}}$ is diagonal.

The $\mathbf{M}$-transformed velocity correlations can be calculated as

$$\mathbf{C}^{\mathbf{y}} \triangleq \mathbb{E}\{\dot{\mathbf{y}}\dot{\mathbf{y}}^\dagger\} = \mathbb{E}\{\mathbf{M}\dot{\mathbf{x}}\dot{\mathbf{x}}^\dagger\mathbf{M}^\dagger\} = \mathbf{M}\mathbb{E}\{\dot{\mathbf{x}}\dot{\mathbf{x}}^\dagger\}\mathbf{M}^\dagger = \mathbf{M}\mathbf{C}^{\mathbf{x}}\mathbf{M}^\dagger. \tag{2.22}$$

The first condition imposes that

$$\mathbf{C}^{\mathbf{y}} = \mathbf{M}\mathbf{C}^{\mathbf{x}}\mathbf{M}^\dagger = \mathbf{I}_{n\times n}. \tag{2.23}$$

From Linear Algebra we know that a solution to this equation can be obtained by eigenvalue decomposition (EVD) of $\mathbf{C}^{\mathbf{x}}$

$$\mathbf{E}^\dagger\mathbf{C}^{\mathbf{x}}\mathbf{E} = \mathbf{\Lambda}, \tag{2.24}$$

where $\mathbf{E}$ contains the eigenvectors of the matrix $\mathbf{C}^{\mathbf{x}}$ and $\mathbf{\Lambda}$ is a diagonal matrix of the eigenvalues ordered the same as the corresponding eigenvectors

in $\mathbf{E}$. Thus

$$\Rightarrow \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{E}^{\dagger}\mathbf{C}^{\mathbf{x}}\mathbf{E}\mathbf{\Lambda}^{-\frac{1}{2}} = \mathbf{I}_{n\times n} \Rightarrow \mathbf{M}^{\dagger} = \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{E}^{\dagger} \tag{2.25}$$

and $\mathbf{M}^{\dagger}$ is a transformation which diagonalize the data's velocity correlations.

It is easy to show that the general form of the solution of (2.23) is a rotated version of the calculated $\mathbf{M}^{\dagger}$, i.e.

$$\mathbf{M} = \mathbf{U}\mathbf{M}^{\dagger} \qquad s.t. \quad \mathbf{U}\mathbf{U}^{\dagger} = \mathbf{I}_{n\times n}. \tag{2.26}$$

The second condition proposed in the paper on (2.21) results in the proper rotation and unifies the transformation $\mathbf{M}$. It says

$$\sum_{m} \boldsymbol{\mathcal{C}}^{\mathbf{y}}_{klmm}(\mathbf{x}) = [\mathbf{D}(\mathbf{x})]_{kl} \tag{2.27}$$

where $\boldsymbol{\mathcal{C}}$ is the four-dimension correlation tensor of $\dot{\mathbf{y}}$ defined as

$$[\boldsymbol{\mathcal{C}}]_{ijklmn} \triangleq \mathbb{E}\{\dot{y}_i\dot{y}_j\dot{y}_k\dot{y}_l\}, \tag{2.28}$$

and $\mathbf{D}$ is a diagonal matrix.

The left side of (2.27) can be written as

$$\sum_{m} \boldsymbol{\mathcal{C}}^{\mathbf{y}}_{klmm} = \sum_{m} \mathbb{E}\{\dot{y}_k\dot{y}_l\dot{y}_m^2\} = \mathbb{E}\{\sum_{m} \dot{y}_k\dot{y}_l\dot{y}_m^2\} \tag{2.29}$$

$$= \mathbb{E}\{\dot{y}_k\dot{y}_l \sum_{m} \dot{y}_m^2\} = \mathbb{E}\{\dot{y}_k\dot{y}_l\|\dot{\mathbf{y}}\|^2\} \tag{2.30}$$

Defining an $n \times n$ matrix named $\mathbf{T}_1$, whose entries are equal to $\sum_m \boldsymbol{\mathcal{C}}^{\mathbf{y}}_{klmm}$ with the corresponding index, we will have

$$[\mathbf{T}_1]_{kl} = \sum_{m} \boldsymbol{\mathcal{C}}^{\mathbf{y}}_{klmm} = \mathbb{E}\{\dot{y}_k\dot{y}_l\|\dot{\mathbf{y}}\|^2\} \Rightarrow \mathbf{T}_1 = \mathbb{E}\{\dot{\mathbf{y}}\dot{\mathbf{y}}^{\dagger}\|\dot{\mathbf{y}}\|^2\}. \tag{2.31}$$

Now let us write the above equations for the $\mathbf{M}^{\dagger}$-transformed version of $\dot{\mathbf{x}}$ as

$$\dot{\mathbf{y}}^{\dagger} \triangleq \mathbf{M}^{\dagger}\dot{\mathbf{x}} \;\Rightarrow\; \dot{\mathbf{y}} = \mathbf{U}\dot{\mathbf{y}}^{\dagger} \;\Rightarrow\; \sum_{m}\sum_{m} \boldsymbol{\mathcal{C}}^{\mathbf{y}^{\dagger}}_{klmm} = \mathbb{E}\{\dot{y}_k^{\dagger}\dot{y}_l^{\dagger}\|\dot{\mathbf{y}}^{\dagger}\|^2\}. \tag{2.32}$$

Similarly, $\mathbf{T}_2$ can be defined as

$$[\mathbf{T}_2]_{kl} = \sum_{m} \boldsymbol{\mathcal{C}}^{\mathbf{y}^{\dagger}}_{klmm} \quad \Rightarrow \quad \mathbf{T}_2 = \mathbb{E}\{\dot{\mathbf{y}}^{\dagger}\dot{\mathbf{y}}^{\dagger\dagger}\|\dot{\mathbf{y}}^{\dagger}\|^2\}. \tag{2.33}$$

Considering (2.32) and (2.33), the relation between $\mathbf{T}_1$ and $\mathbf{T}_2$ can be formulated as

$$\mathbf{T}_1 = \mathbb{E}\{\dot{\mathbf{y}}\dot{\mathbf{y}}^\dagger \|\dot{\mathbf{y}}\|^2\} = \mathbb{E}\{\mathbf{U}\dot{\mathbf{y}}^\dagger \dot{\mathbf{y}}^{\dagger\dagger}\mathbf{U}^\dagger \|\mathbf{U}\dot{\mathbf{y}}^\dagger\|^2\} \tag{2.34}$$

$$= \mathbb{E}\{\mathbf{U}\dot{\mathbf{y}}^\dagger \dot{\mathbf{y}}^{\dagger\dagger} \|\dot{\mathbf{y}}^\dagger\|^2 \mathbf{U}^\dagger\} \tag{2.35}$$

$$= \mathbf{U}\mathbb{E}\{\dot{\mathbf{y}}^\dagger \dot{\mathbf{y}}^{\dagger\dagger} \|\dot{\mathbf{y}}^\dagger\|^2\}\mathbf{U}^\dagger \tag{2.36}$$

$$= \mathbf{U}\mathbf{T}_2\mathbf{U}^\dagger, \tag{2.37}$$

where (2.35) comes from the fact that rotations do not change the norm.

Eq. (2.27) says that $\mathbf{T}_1$ needs to be a diagonal matrix. Therefore from (2.37), it is concluded that the eigenvectors of the matrix $\mathbf{T}_2$ should be used as the columns of the rotation matrix $\mathbf{U}$. Therefore, the transformation matrix $\mathbf{M}$ is constructed by multiplying three matrices as

$$\mathbf{M} = \mathbf{U}\mathbf{M}^\dagger = \mathbf{U}\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{E}^\dagger \tag{2.38}$$

where $\mathbf{E}$ and $\mathbf{\Lambda}$ contain the eigenvectors and eigenvalues of the matrix $\mathbf{C}^{\mathbf{x}}$, respectively, and $\mathbf{U}$ contains the eigenvectors of the matrix $\mathbf{T}_2$.

The transformation $\mathbf{M}$ which is constructed according to (2.38), is a unique "whitening" transform that is proposed to be calculated locally. In other words, the probability distribution function of the velocity of the observations is estimated at each point (neighborhood) by time-averaging (considering the observations to be ergodic) and then the whitening transformation is calculated for this specific neighborhood. So it will be more accurate to use the notation $\mathbf{M}(\mathbf{x})$ to show that it depends on the location. The proposed method recalls performing an ICA-like algorithm "locally" and for the "derivatives" of the observations.

It is easy to show that if the data is separable (i.e. if the transformation has whitened the velocity of the observations), the auto-correlation of each of the transformed signals, in any order, is only a function of one of the sources. Thus the paper proposes to calculate data's autocorrelations in the transformed domain in different orders to see if they are whitened or not. For

this purpose, the following triples are plotted in a three dimensional space.

$$\begin{aligned} I_1 &= \{\boldsymbol{\mathcal{C}}^{\mathbf{y}}_{111}(\mathbf{x}), \boldsymbol{\mathcal{C}}^{\mathbf{y}}_{1111}(\mathbf{x}), \boldsymbol{\mathcal{C}}^{\mathbf{y}}_{11111}(\mathbf{x})\} \\ I_2 &= \{\boldsymbol{\mathcal{C}}^{\mathbf{y}}_{222}(\mathbf{x}), \boldsymbol{\mathcal{C}}^{\mathbf{y}}_{2222}(\mathbf{x}), \boldsymbol{\mathcal{C}}^{\mathbf{y}}_{22222}(\mathbf{x})\} \\ &\vdots \\ I_n &= \{\boldsymbol{\mathcal{C}}^{\mathbf{y}}_{nnn}(\mathbf{x}), \boldsymbol{\mathcal{C}}^{\mathbf{y}}_{nnnn}(\mathbf{x}), \boldsymbol{\mathcal{C}}^{\mathbf{y}}_{nnnnn}(\mathbf{x})\} \end{aligned} \tag{2.39}$$

Since each of the triples of (2.39) is supposed to be a function of only one of the source signals, it is expected to lie in a one dimensional sub-space. The paper claims that the data has been separated iff it happens.

There are parts in the paper that are not clear enough, and claims which are neither mathematically proven nor thoroughly explained. Even the above notes on the proposed method and the mathematical expressions do not exist in the paper and come from our understanding of that. However, our interpretation of the main idea of the paper, i.e. using a more general definition of independence (2.17) than RV independence, was inspiring for our approach to general nonlinear BSS problem.

2.3 Conclusion

As stated before, linear BSS problem has been vastly studied in the past 30 years and many algorithms and methods are proposed for that. However, since ICA was shown not to be able to separate general nonlinear mixtures, nonlinear BSS has only been considered for some particular structured models. Nonetheless, considering temporal information of the signals leads to a more general definition of independence than RV independence, which results in separability of more general nonlinear mixtures.

It should be mentioned that this work, as well as other general nonlinear BSS methods [Comon and Jutten, 2010, Jutten and Karhunen, 2003, Jutten and Karhunen, 2004,Ehsandoust et al., 2016], suffers from the ambiguity of a nonlinear transformation that cannot be resolved. However, it is important to differentiate between source *separation* and source *reconstruction*. In fact,

once the sources are *separated*, the task of BSS is done. Source reconstruction is a more general task that is out of the scope of this work.

Although source separation can be sufficient and efficient in the cases where BSS is used as a first step before classification, in practical applications of source reconstruction, the proposed method of this work, as well as most other papers on nonlinear BSS, serves as a first step which *separates* the sources and maybe needs to be followed by a reconstruction method. For this last step, simple and weak priors on a source like sparsity [Duarte et al., 2015], bandwidth [Dogancay, 2005], zero-crossing [Marvasti and Jain, 1986], etc. can be used for reconstructing a signal without knowing the nonlinear distortion. This point is more elaborated in the following chapters.

3 A General Approach to Nonlinear BSS

Contents

In this chapter, the separability of general nonlinear BSS is studied and a basic algorithm for source separation is proposed. The proposed approach is mainly based on using signal derivatives in order to employ temporal information of the signals, as introduced in previous chapters.

Please note that this chapter will provide a method, on which different algorithms can be developed for solving nonlinear BSS problems. It proposes a general approach for performing the separation in nonlinear mixtures as well as the necessary conditions on the model. A separation algorithm is also provided and its efficiency is proved by simulations.

The idea of this chapter is original and has been published in [Ehsandoust et al., 2015] and [Ehsandoust et al., 2017a]. This chapter is organized as follows. The novel approach for solving the nonlinear BSS problem is introduced in Section 3.1. Then a discussion on the separability and the assumptions on the model is provided. Section 3.2 contains the basic algorithms proposed for performing the separation. The algorithms are implemented and tested with examples, the results of which are presented in Section 3.4. Finally, conclusions, remained questions and future works are discussed in the last section.

3.1 The Main Idea

The proposed approach for nonlinear BSS in this chapter is mainly based on local linear approximation of the nonlinear mixture. So, it is applicable to any nonlinear model satisfying the mentioned assumptions. In addition, a discussion is made in Section 3.4 showing how its performance relates with the amount of the nonlinearity of the mixture (supported by simulation results).

3.1.1 An Example of Local Linear Approximation of Nonlinear Functions

In this subsection, we show how the idea of local linear approximation can be applied on nonlinear BSS in hyperspectral images. We have published this concept, with more details and simulation results, in [Drumetz et al., 2017]. In that paper, the locally approximated model is called "space-variant", where its relationship with the original nonlinear mixture is investigated. The theoretical results are employed in the well-known hyperspectral image unmixing application, which confirm the validity of the proposed approach.

3.1.1.1 Hyperspectral Image Unmixing

Hyperspectral image unmixing is a source separation problem whose goal is to identify the spectral signatures of the materials present in the imaged scene (called endmembers), and to estimate their proportions (called abundances) in each pixel. Usually, the contributions of each material are assumed to be perfectly represented by a single spectral signature and to add up in a linear way. However, the main two limitations of this model have been identified as nonlinear mixing phenomena and spectral variability, i.e. the intra-class variability of the materials.

The former limitation has been addressed by designing nonlinear mixture models, while the second can be dealt with by using space varying models (usually keeping the linear mixture assumption). The typical example is a linear mixing model where the sources can vary from one pixel to the other.

A hyperspectral image is represented as a matrix $\mathbf{X} \in \mathbb{R}^{L \times N}$, where L is the number of considered wavelengths, and N is the number of pixels in the image. The endmembers are gathered in the columns of a matrix $\mathbf{S} \in \mathbb{R}^{L \times P}$, where P is the number of considered materials. The abundance coefficients for each pixel and each material are stored in a matrix $\mathbf{A} \in \mathbb{R}^{P \times N}$. Then a

simple linear mixing model (LMM) writes, for a given pixel $\mathbf{x}_k \in \mathbb{R}^L$:

$$\mathbf{x}_k = \sum_{p=1}^{P} a_{pk}\mathbf{s}_p + \mathbf{e}_k \tag{3.1}$$

where $\mathbf{e}_k$ is an additive noise, often assumed to be zero mean Gaussian-distributed, with an isotropic covariance matrix.

The endmembers, being reflectance spectra, are constrained to be non-negative. In addition, the abundances are proportions, so they are usually constrained to be positive, and to sum to one in each pixel. Geometrically, the LMM constrains the data to live in a simplex spanned by the endmembers. In many cases, the LMM is a reasonable approximation of the physics of the mixtures. However, in more complex cases nonlinear mixture models are necessary, for instance when rays of light undergo multiple reflections before reaching the sensor (e.g. in tree canopies) [Heylen et al., 2014, Dobigeon et al., 2014].

This issue fostered research on nonlinear mixing models and the corresponding unmixing algorithms (e.g. [Meganem et al., 2014, Altmann et al., 2014, Févotte and Dobigeon, 2015]). A popular choice for modeling this problem is the class of linear-quadratic models, which take into account second order interactions between materials, under the form of product spectra $\mathbf{s}_p \odot \mathbf{s}_q$, where $\odot$ is the Hadamard (element-wise) product as

$$\mathbf{x}_k = \sum_{p=1}^{P} a_{pk}\mathbf{s}_p + \sum_{p=1}^{P}\sum_{q=p}^{P} b_{pqk}\mathbf{s}_p \odot \mathbf{s}_q + \mathbf{e}_k \tag{3.2}$$

where b_{pqk} are positive quadratic interaction coefficients for each pixel k and each pair of materials (p, q). The higher order interactions are usually omitted, since they are considered to have a low contribution to the final at-sensor reflectance. The data is now bound to lie in a nonlinear manifold which is more complex than a simplex.

The other limitation comes from the representation of a single endmember by a unique spectral signature. This is a very convenient approximation, but an endmember is actually more accurately described by a collection of

signatures, which account for the intra-class variability of that material [Zare and Ho, 2014]. Many physical phenomena can induce variations on the spectra of pure materials, be it a change in their physico-chemical composition, or the topography of the scene, which locally changes the incidence angle of the light and the viewing angle of the sensor. This phenomenon is referred to as *endmember variability* [Thouvenin et al., 2016, Halimi et al., 2015, Henrot et al., 2016]. A physics-inspired model to explain illumination induced variability is the Extended Linear Mixing Model (ELMM) [Drumetz et al., 2016], which writes

$$\mathbf{x}_n = \sum_{p=1}^{P} a_{pn}\psi_{pn}\mathbf{s}_p + \mathbf{e}_n \tag{3.3}$$

where ψ_{pn} is a positive scaling factor whose effect is to rescale locally each endmember, the variations between variants of the same material due to changing illumination conditions being reasonably well explained by a scaling variation. Geometrically, the data may now lie inside a convex cone spanned by the endmembers. More specifically, each pixel belongs to a simplex, whose vertices can slide on lines (passing through the origin) which correspond to the edges of the convex cone.

Spectral variability and nonlinear mixtures are physically very different phenomena. Mathematically, spectral variability essentially amounts to using a space varying (usually linear) mixing model, while a general nonlinear mixing model is spatially invariant. In [Revel et al., 2016], the joint consideration of both nonlinearities (through a linear-quadratic model) and spectral variability was experimentally shown not to give substantially better abundance estimation results than considering endmember variability alone. The dataset considered was acquired over an urban area, where both phenomena were expected to be nonnegligible, which suggests that using a nonlinear model along with a variability model was not necessary, and that the latter can already handle nonlinear effects to some extent.

Nonetheless, following the ideas of [Ehsandoust et al., 2017a], we provide theoretical insight to these results, by showing that there is a mathematical

connection between both approaches. We show that a local Taylor expansion of a generic nonlinear model can be related to a variant of the spatially varying ELMM. This derivation, as well as the experiments, show that the ELMM has the ability to recover abundances from nonlinear mixtures, even though it was derived from physical considerations about endmember variability in linear mixtures.

3.1.1.2 Connection Between Nonlinear Models and Variability Models

A generic (noise free) nonlinear mixing model can be expressed, for a given pixel n and wavelength l, as:

$$x_{ln} = f_n(s_{l1}, s_{l2}, ..., s_{lP}) \tag{3.4}$$

where s_{lp} is the value of endmember p at wavelength l, and $f_n : \mathbb{R}^P \to \mathbb{R}$ is a generic nonlinear function, which does not depend on the considered spectral band. Assuming the nonlinear function f_n is sufficiently regular, and that the sources are allowed to vary, we can perform an M^{th} order Taylor expansion around $(0, 0, ..., 0)$ as

$$x_{ln} = f_n(\mathbf{0}) + \mathbf{s}_{l:}^\top \nabla f_n(\mathbf{0}) + \mathbf{s}_{l:}^\top \nabla^2 f_n(\mathbf{0}) \mathbf{s}_{l:} + \cdots + o(\|\mathbf{s}_{l:}\|^M) \tag{3.5}$$

$$= \sum_{p=1}^{P} \frac{\partial f_n}{\partial s_{lp}}(\mathbf{0}) s_{lp} + \sum_{p=1}^{P} \sum_{q=1}^{P} \frac{\partial^2 f_n}{\partial s_{lp} \partial s_{lq}}(\mathbf{0}) s_{lp} s_{lq} + \cdots + o(\|\mathbf{s}_{l:}\|^M) \tag{3.6}$$

where in 3.6, we have discarded the constant term (if the sources are zero, nothing is observed, i.e. we assume that $f_n(\mathbf{0}) = 0$), and where $\mathbf{s}_{l:} = [s_{l1}, ..., s_{lP}]^\top \in \mathbb{R}^P$. Note that even though this expansion is performed in $\mathbf{0}$, the error term $o(||\mathbf{s}_{l:}||^M)$ is likely to be small around $\mathbf{s}_{l:}$, because linear-quadratic and multilinear mixing models approximate the physics of hyperspectral imaging well, i.e. if the underlying nonlinear function is close to polynomial, we expect the coefficients of the expansion to be very close to the actual coefficients of the polynomial. In addition, even with a more general model, the expansion will also be valid in the neighborhood of $\mathbf{s}_{l:}$ with a high enough order M of the expansion.

We change the notation of the coefficients of the expansion, keeping in mind their dependence with respect to the different variables of the model, and also change the indexing such that the identical second order terms are gathered in only one term as

$$x_{ln} = \sum_{p=1}^{P} \alpha_{pn} s_{lp} + \sum_{p=1}^{P} \sum_{q=p}^{P} \beta_{pqn} s_{lp} s_{lq} + \cdots + o(\|\mathbf{s}_{l:}\|^M). \tag{3.7}$$

There is no dependence of the coefficients on the spectral band since we assumed the nonlinearity affects all spectral bands equally. If, following the physics of the problem, we assume the true nonlinear model is close enough to a multilinear model, that is a generalization of model (3.2) to higher order interaction terms, then considering the uniqueness of Taylor expansion, we can safely assume that $\alpha_{pn} \approx a_{pn}$ and $\beta_{pqn} \approx b_{pqn}$, and then model (3.2) is a truncation at the second order of

$$x_{ln} = \sum_{p=1}^{P} a_{pn} s_{lp} + \sum_{p=1}^{P} \sum_{q=p}^{P} b_{pqn} s_{lp} s_{lq} + \cdots + o(\|\mathbf{s}_{l:}\|^M). \tag{3.8}$$

On the other hand, if we factor the coefficient $\alpha_{pn} s_{lp}$ in terms of Eq. (3.7), we obtain

$$x_{ln} = \sum_{p=1}^{P} \alpha_{pn} \left(1 + \sum_{q=p}^{P} \frac{\beta_{pqn}}{\alpha_{pn}} s_{lq} + \cdots + o(\|\mathbf{s}_{l:}\|^M) \right) s_{lp}. \tag{3.9}$$

In order to make this factorization possible, we had to assume that all materials have a nonzero linear coefficient in pixel n. If the true model is multilinear, then these coefficients correspond to the abundances, and we simply have to remove the endmembers with zero abundance in pixel n from the equation.

By denoting the factor between the parentheses by ψ_{lpn}, and again by assuming the true model is close to multilinear, we obtain

$$x_{ln} = \sum_{p=1}^{P} a_{pn} \psi_{lpn} s_{lp} \tag{3.10}$$

which is formally close to the variability model (3.3), with the notable exception that the scaling factor now depends on the wavelength. The ELMM

is essentially a linear model where each endmember is allowed to vary spatially according to the law $\mathbf{s}_{pn} = \psi_{pn}\mathbf{s}_p$, where $\mathbf{s}_p$ is a reference signature for material p. The scaling factor does not depend on the wavelength here.

Note that model (3.10) is very general and may be too flexible to provide reliable performance without additional well chosen regularizations. Still, this shows that the space invariant (in terms of the endmembers) nonlinear model (3.4) can be locally approximated by a spatially varying linear model.

Model (3.10) is more general than truncating model (3.8) at the second order, since the scaling factor incorporates information about the linear and quadratic terms of the expansion, but also about higher order terms. In [Drumetz et al., 2017], we have also shown experimental evidences of the capability of space-variant models, and in particular ELMM, to extract information related to nonlinear mixtures, confirming that this model can actually obtain better abundance estimations than a linear-quadratic model-based algorithm in several cases, and thus better handles more general nonlinear mixtures than a model which is specifically designed for this purpose.

3.1.2 Signal Derivatives

Our main idea for general nonlinear BSS is based on the fact that *the derivatives of the sources are locally mixed linearly even though the mixture model is nonlinear in general.* Indeed, if the nonlinear mapping $\mathbf{f}$ is differentiable at each point, one can derive a local linear approximation of it involving the derivatives of sources and observations. This is easily seen from

$$x_i(t) = f_i(\mathbf{s}(t)) \quad \Rightarrow \quad \frac{dx_i}{dt} = \sum_{j=1}^{n} \frac{\partial f_i}{\partial s_j}\frac{ds_j}{dt} \tag{3.11}$$

$$\Rightarrow \quad \dot{\mathbf{x}} = \mathbf{J}_{\mathbf{f};t}(\mathbf{s})\dot{\mathbf{s}}, \tag{3.12}$$

where $\mathbf{J}_{\mathbf{f};t}(\mathbf{s})$ is the Jacobian of the mixing function $\mathbf{f}$ defined as (2.10).

It should be noted that the precise definition of the derivative of a random process $p(t)$ is in the mean square sense, i.e. a random process $\dot{p}(t)$ is the time-derivative of a random process $p(t)$ iff $\lim_{\epsilon\to 0} \mathbb{E}[|\frac{p(t+\epsilon)-p(t)}{\epsilon} - \dot{p}(t)|^2] = 0$.

In this sense, it can be shown that if $p(t)$ is stationary, the auto-correlation function $R_p(\tau)$ and its first and second order derivatives, $R'_p(\tau)$ and $R''_p(\tau)$ exist. Nonetheless, in the rest of the chapter, for the reason of simplicity, we use the equality symbol "=" for the equality of random processes in the mean squared sense as well.

It is worth noting that $\mathbf{J}_{\mathbf{f};t}(\mathbf{s})$ is the Jacobian of the nonlinear time-invariant function $\mathbf{f}$ and is a function of the sources $\mathbf{s}$. However, since the source vector is a random process and varies over time, the elements of $\mathbf{J}_{\mathbf{f};t}(\mathbf{s})$ change over time as well. This is why t does not directly appear in (2.10), and in (3.12) is considered as an index of the Jacobian matrix, but not an input argument. Thus, (3.12) is a *local linear* mixture model.

So, one can firstly separate the local linear mixtures of the source derivatives using an adaptive linear BSS technique, and then, use an integration step to reconstruct the source signals themselves. Applying a linear BSS method on *derivatives* of the sources imposes some necessary conditions on them, which will be studied in the following section. Particularly, the DC value of signals is removed in the first step of any classical linear BSS method, hence the derivatives in our framework. Nonetheless, as mentioned earlier, the goal in this work is to reconstruct a "nonlinear copy" of the sources which can still be achieved considering this DC-removal pre-processing (the "nonlinear copy" is mathematically defined in the next section).

This idea can also be understood from a totally different point of view as follows. Considering the general model of Fig. 1.1, let us define the matrix $\boldsymbol{\nabla} \in \mathbb{R}^{n\times n}$ containing the partial derivatives of the output signals with respect to the sources as follows

$$\boldsymbol{\nabla} \triangleq \frac{\partial \mathbf{y}}{\partial \mathbf{s}} = \begin{bmatrix} \frac{\partial y_1}{\partial s_1} & \frac{\partial y_1}{\partial s_2} & \cdots & \frac{\partial y_1}{\partial s_n} \\ \frac{\partial y_2}{\partial s_1} & \frac{\partial y_2}{\partial s_2} & \cdots & \frac{\partial y_n}{\partial s_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial s_1} & \frac{\partial y_n}{\partial s_2} & \cdots & \frac{\partial y_n}{\partial s_n} \end{bmatrix} \tag{3.13}$$

or equivalently

$$[\boldsymbol{\nabla}]_{ij} = \frac{\partial y_i}{\partial s_j}. \tag{3.14}$$

So separation is performed if and only if the matrix $\boldsymbol{\nabla}$ is a so-called "nonlinear copy" or a "trivial mapping" matrix, which contains only one non-zero element at each row and column (this term will be defined more precisely in Section 3.1.3).

One can use the chain rule to expand (3.14) as

$$\mathbf{y} = \mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{f}(\mathbf{s})) \tag{3.15}$$

$$\Rightarrow \quad [\boldsymbol{\nabla}]_{ij} = \frac{\partial y_i}{\partial s_j} = \sum_{k=1}^{m} \frac{\partial y_i}{\partial x_k} \frac{\partial x_k}{\partial s_j} \tag{3.16}$$

where m is the number of the observations (not necessarily equal to the number of the sources). Therefore we have

$$\boldsymbol{\nabla} = \frac{\partial \mathbf{y}}{\partial \mathbf{s}} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \mathbf{s}} = \underbrace{\mathbf{J_g}}_{n \times m} \underbrace{\mathbf{J_f}}_{m \times n} \tag{3.17}$$

where $\mathbf{J_g}$ and $\mathbf{J_f}$ represent the Jacobian matrices of the nonlinear functions $\mathbf{g}$ and $\mathbf{f}$ respectively.

Making $\boldsymbol{\nabla}$ contain only one non-zero element at each row and column, is very similar to linear BSS (Theorem 1) where the goal is to make the matrix $\mathbf{C} = \mathbf{BA}$ contain only one non-zero element at each row and column. The only difference is that in linear BSS, the mixing and separating matrices $\mathbf{A}$ and $\mathbf{B}$ are fixed, while in the nonlinear problem (3.17), the multiplicands $\mathbf{J_g}$ and $\mathbf{J_f}$ vary as the sources take different values along time.

This fact inspires the idea that considering (3.17) *locally* such that the variations of the matrices are negligible, linear BSS methods may be utilized for solving the nonlinear BSS problem. However, it should be mentioned that the local equivalent linear BSS problem to (3.17), would be a linear mixture whose mixing matrix equals to $\mathbf{J_f}$. As proposed in (3.11) and (3.12), it is *derivatives of the sources* that are mixed through the matrix $\mathbf{J_f}$. Thus locally solving (3.12) results in nonlinear BSS.

In the following, the problem of interest is formulated and all the assumptions are mentioned. Then the proposed approach is described and the separability is discussed. The discussion is made from two points of view: mathematical expressions and system analysis.

3.1.3 Problem Definition and Assumptions

Similar to definition 2.1, a "nonlinear copy" is defined as follows.

Definition Given $\mathbf{s}$ an n-dimensional vector, $\mathbf{y} = \mathbf{c}(\mathbf{s})$ is called a "nonlinear copy" of $\mathbf{s}$ if it has the same dimension as $\mathbf{s}$ and each element y_i of it is an invertible nonlinear function of one and only one of the elements of $\mathbf{s}$. It can be written as

$$\forall\ 1 \leq i \leq n \quad y_i = c_i(s_{\tau_i}) \tag{3.18}$$

where c_i for $i = 1, \ldots, n$ is an invertible nonlinear function and $(\tau_1, \tau_2, \ldots, \tau_n)$ is a permutation of $(1, 2, \ldots, n)$. □

The transformation $\mathbf{c}(\cdot)$, which only contains component-wise nonlinear functions and permutations, is called a "nonlinear copy function" or a "trivial nonlinear mapping".

Thus, the general nonlinear BSS problem can be defined as follows. Let an observation vector $\mathbf{x}(t)$ be an unknown nonlinear mixture of an unknown source vector $\mathbf{s}(t)$ as (1.1), or equivalently

$$\forall\ i \quad x_i(t) = f_i(\mathbf{s}(t)). \tag{3.19}$$

Source separation consists of finding a nonlinear mapping $\mathbf{g}$ as

$$\text{find} \quad \mathbf{g} \qquad \text{s.t.} \qquad \mathbf{g} \circ \mathbf{f} = \mathbf{c} \tag{3.20}$$

where $\mathbf{c} = \mathbf{g} \circ \mathbf{f}$ is a "nonlinear copy" function.

According to (3.17) the following basic theorem can be proposed.

Theorem 2. *In the model of Fig. 1.1 the sources $s_1, \ldots, s_n$ are mixed through a nonlinear function $\mathbf{f}$ resulting in the observations $x_1, \ldots, x_m$. A function $\mathbf{g}$ is separating, i.e. $\mathbf{y} = \mathbf{g}(\mathbf{x})$ is a "nonlinear copy" of the source vector $\mathbf{s}$, if and only if $\boldsymbol{\nabla} = \mathbf{J_g J_f}$ is a permuted diagonal matrix of functions, i.e. $\boldsymbol{\nabla} = \boldsymbol{\Lambda}\mathbf{P}$ where $\mathbf{P}$ is a permutation and $\boldsymbol{\Lambda}$ is a matrix whose off-diagonal entries are equal to zero (the diagonal elements are not necessarily constant).*

It should be noted that each element of $\mathbf{J_f}$ is in fact a nonlinear function of the sources. Thus, as sources take different values along time, the value of the elements of $\mathbf{J_f}$ should change such that $\mathbf{J_g J_f}$ always remains a copy function.

As a consequence, "finding a function $\mathbf{g}$ whose Jacobian $\mathbf{J_g}$ makes $\boldsymbol{\nabla} = \mathbf{J_g J_f}$ a permuted diagonal matrix" is equivalent to "finding a matrix of functions $\mathbf{J_g}$ which linearly separates the derivatives of the sources for all values taken by $\mathbf{s}$". This interpretation complies with previous nonlinear BSS results, especially those concerning derivatives of signals, e.g. [Levin, 2010].

Note that an ambiguity of a permutation and a nonlinear function in reconstruction of the sources cannot be resolved. It is evident from the definition of a nonlinear copy function and (3.20). In addition, it can also be understood from another point of view by looking at the Jacobian of the mixing function (see Section 3.1.5).

The above source separation problem is ill-posed without additional assumptions, either on the nonlinear mapping $\mathbf{f}$ or on the sources. In this chapter, we consider the following assumptions:

1. The number of the sources is equal to the number of the observations,
2. $\mathbf{f}$ is invertible,
3. $\mathbf{f}$ is memoryless,
4. $\mathbf{f}$ is time-invariant,
5. $\mathbf{f} \in \mathbb{C}^1$ (i.e. it is differentiable with continuous first-order derivative),
6. Sources $s_1(t), \ldots, s_n(t)$ are differentiable, hence colored (this assumption implies continuity and smoothness),
7. Derivatives of the sources $\{\dot{s}_1(t), \ldots, \dot{s}_n(t)\}$ are mutually independent and
8. At most, one of the derivatives of the sources follows the Gaussian distribution.

These assumptions are satisfied in most practical applications where the signals and the nonlinear mixing model correspond to real physical phenomena. In fact, the assumptions 1 to 4 are classical assumptions of BSS that are assumed even in linear cases. If the source signals have different origins (i.e. their physical origins are independent), then they will also be mutually independent *stochastic* processes, hence assumptions 6 and 7 hold.

As a consequence, all applications introduced in the Section 2.2, including hyperspectral imaging [Golbabaee et al., 2013] and determining the concentration of different ions in a combination via smart chemical sensor arrays [Duarte and Jutten, 2014] satisfy the mentioned assumptions. Therefore, nonlinear BSS problems, which can be treated through the proposed approach in this work, do not belong to a specific set of functions and are quite general.

These assumptions are necessary for the proposed approach which will come in Section 3.1.5. Needless to mention, in this approach, derivatives of the signals should contain some information; in other words, signals with constant time-derivatives cannot be treated through this framework. Nevertheless, it is worth adding some remarks about some of them.

The assumption $\mathbf{f} \in \mathbb{C}^1$ imposes the continuity of $\mathbf{J}_\mathbf{f}$. Moreover, according to the inverse function theorem [Spivak, 1965], if a function $\mathbf{f}$ is invertible on a region in its domain and $\mathbf{f} \in \mathbb{C}^1$, 1) its Jacobian $\mathbf{J}_\mathbf{f}$ will be non-singular on that region and 2) the Jacobian of its inverse is equal to the inverse of its Jacobian ($\mathbf{J}_\mathbf{f}^{-1} = \mathbf{J}_{\mathbf{f}^{-1}}$). Consequently, assumptions 5 and 2 result in continuity and non-singularity of $\mathbf{J}_\mathbf{f}$, which makes the local linear BSS problem (3.12) solvable with ICA. Note that if the function is not invertible, although (3.12) is always true, since the Jacobian matrix would not be full-rank everywhere, it does not lead to a solvable BSS problem.

In addition, $\mathbf{f}$ needs to be memoryless and time-invariant, because otherwise $\mathbf{J}_\mathbf{f}$ in (3.12) would also vary along time, hence the variations of local linear approximation would be too difficult to be followed by a BSS algorithm. This limitation will be better understood after Section 3.2 in which

we utilize it for amending the initially proposed method.

Moreover, assumption 6, in combination with the differentiability and continuity of $\mathbf{f}$, implies the smoothness of the variations of the nonlinear function, hence its Jacobian $\mathbf{J_f}$, along time so that it is tractable by adaptive local BSS algorithms. In other words (as it will be elaborated in Section 3.2 and simulation results), the performance of the proposed method depends on the speed of the variations of $\mathbf{J_f}$ along time, which is due to the spectral colorfulness of the sources and the nonlinearity of $\mathbf{f}$ itself.

As mentioned before, the proposed algorithm in this work is based on the statistical independence of the sources. Therefore, as assumed in ICA-based classical BSS methods, mixed signals in (3.12) need to satisfy certain conditions [Comon, 1994]. This is where the assumptions 7 and 8 come from.

It should be noticed that the assumptions 7 and 8 concern *derivatives* of the sources, because in (3.12), the mixed signals are the derivatives of the sources. The assumption 7 can be expressed as

$$\rho_S(\dot{\mathbf{s}}) = \prod_{k=1}^{n} \rho_k(\dot{s}_k) \tag{3.21}$$

where $\rho_S(\dot{\mathbf{s}})$ and $\rho_k(\dot{s}_k)$ correspond to the joint and marginal pdf's of the derivatives of the sources. It should be noted that a stronger assumption than (3.21) was proposed as a necessary and sufficient condition for separability of nonlinear mixtures in [Levin, 2010] (but without any proof or explanation), which needed the signals and their derivatives to be jointly statistically independent (2.17).

Note that (3.21) is a completely different condition from RV independence of the source signals and is not a result of that. Generally, a signal and its derivative can be instantaneously independent: for instance, given the position of a particle at a time, one cannot say anything about its speed at that time. However, the derivative of a signal contains some information about the variations of it (which can be translated to the bandwidth or the amount of spectral colorfulness).

It should be finally declared that the mentioned assumptions are not

claimed to be necessary for the general separability of nonlinear mixtures. One may suggest other approaches and methods for nonlinear BSS, based on other assumptions. However, in the proposed framework, they should necessarily be satisfied and they are sufficient in the sense that if they hold, it will be possible to separate the sources based on the proposed approach.

3.1.4 A Parametric Model

As a result of the section 3.1.3, the separating matrix $\mathbf{J_g}(\mathbf{x})$ should be found such that $\boldsymbol{\nabla} = \mathbf{J_g}\mathbf{J_f}$ is a permuted diagonal matrix (see Theorem 2). Recalling the basic linear BSS problem, especially Theorem 1 [Comon, 1994], enforcing the independence guarantees the separation. In other words, it would be necessary and sufficient that components of the output vector $\dot{\mathbf{y}}(t) = \mathbf{J_g}(\mathbf{x})\dot{\mathbf{x}}(t)$ be mutually independent for achieving the separation.

For making $\dot{\mathbf{y}}(t)$ signals mutually independent, the well-known classic cost function of their statistical *mutual information*, $\mathrm{I}(\dot{\mathbf{y}})$, is chosen to be minimized. However, the difficulty here is that this cost function should be minimized with respect to the $m \times n$ nonlinear functions of the matrix $\mathbf{J_g}(\mathbf{x})$ (equations are firstly written for the more general case $m \neq n$, then in Section 3.1.4.3 the assumption $m = n$ is exploited). Although one may propose a method for optimizing a cost function with respect to functions, in this section, for simplicity, a parametric model for the separating function is assumed. Consequently, the cost function is minimized with respect to those parameters.

3.1.4.1 The Model

Now, let us model the separating function $\mathbf{g}(\cdot)$ in a parametric manner as

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \\ \vdots \\ g_n(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta_1}^\dagger \\ \boldsymbol{\theta_2}^\dagger \\ \vdots \\ \boldsymbol{\theta_n}^\dagger \end{bmatrix} \mathbf{k}(\mathbf{x}) = \boldsymbol{\Theta}_{n\times P}\mathbf{k}(\mathbf{x}) \tag{3.22}$$

where $\boldsymbol{\theta_i}$ for $i = 1, \ldots, n$ is an $P \times 1$ column vector of the parameters (constant scalars), $\mathbf{k}(\mathbf{x}) \in \mathbb{R}^P$ is the column vector of kernel functions of the parametric model and P is the number of the parameters of each entry $g_i(\cdot)$ which is obviously equal to the number of the kernel functions in $\mathbf{k}(\cdot)$. For example, in order to model an L^{th} order polynomial of n sources, one has to take all monomials of the degree less than or equal to L as the kernel functions. The interesting point of this model is that it is linear with respect to the parameters, which simplifies the algorithm significantly.

It is worth noting that the kernel functions can be chosen according to the application in order to best fit the nonlinearity of the mixture. However, there are two conditions that should be met:

1. Avoiding the redundancy of parameters which causes the degeneration of the matrices, the kernels should be linearly independent functions, i.e. they should really make a P-dimensional sub-space in the infinite-dimensional space of the functions. For example, different monomials are linearly independent functions. Nevertheless, it is better to choose a set of orthonormal functions as the kernels, which for all $1 \leq i, j \leq P$ satisfy

$$< k_i(\mathbf{x}), k_j(\mathbf{x}) >= \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{3.23}$$

   where $< k_i(\mathbf{x}), k_j(\mathbf{x}) > \stackrel{\triangle}{=} \int k_i(\mathbf{x}) k_j(\mathbf{x}) \, \mathrm{d}\,\mathbf{x}$ is the inner product of the functions and the integral is over the domain of the functions. In this case, the kernels make a basis for the P-dimensional subspace.

2. The number of kernel functions should be at least equal to the number of sources, i.e. $P \geq n$, in order to be capable of estimating the n source signals. Note that in linear BSS, $\mathbf{k}(\mathbf{x}) = \mathbf{x}$ and $P = n$, which is the simplest form of the problem.

According to Taylor's theorem, any smooth enough nonlinear function can be approximated by a polynomial with an arbitrary small error (choosing

the order L high enough). Thus normalized monomials up to a high enough order are generally a good choice of kernel functions.

Parametrizing $\mathbf{g}(\mathbf{x})$ as (3.22) with $P \geq n$ can also be interpreted from another point of view: nonlinearly projecting the observations $\mathbf{x}$ to a high dimensional kernel-induced space in order that they are more easily separated. This implies a connection between the proposed method and the well-known kernel method for dealing with nonlinearities [Bach and Jordan, 2002,Muller et al., 2001].

3.1.4.2 Mutual Information Minimization

For minimizing the mutual information between derivatives of the output signals $\mathrm{I}(\dot{\mathbf{y}})$ with respect to the parameters, the Steepest Descent framework can be used as

$$\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} - \mu \frac{\partial \mathrm{I}(\dot{\mathbf{y}})}{\partial \boldsymbol{\Theta}} \tag{3.24}$$

where $\mathrm{I}(\dot{\mathbf{y}}) = (\sum_{i=1}^{n} \mathrm{H}(\dot{y}_i)) - \mathrm{H}(\dot{\mathbf{y}})$ is the mutual information of $\dot{\mathbf{y}}$, $\mathrm{H}(\cdot)$ denotes the statistical Shannon entropy, and μ is the step size of the algorithm. It is worth recalling that for any $1 \leq i \leq n$, the entropy of $\dot{y}_i$ is defined as

$$\mathrm{H}(\dot{y}_i) = -\mathbb{E}\{\ln \rho_{\dot{Y}_i}(\dot{y}_i)\} \tag{3.25}$$

where $\rho_{\dot{Y}_i}(\dot{y}_i)$ is the pdf of $\dot{y}_i$ and $\mathbb{E}$ denotes the expected value.

Taking the derivative of $\mathrm{I}(\dot{\mathbf{y}})$ with respect to any parameter vector $\boldsymbol{\theta_k}$ for $1 \leq k \leq n$ leads to

$$\frac{\partial \mathrm{I}(\dot{\mathbf{y}})}{\partial \boldsymbol{\theta_k}} = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{\partial \mathrm{I}(\dot{\mathbf{y}})}{\partial [\mathbf{J_g}(\mathbf{x})]_{ij}} \frac{\partial [\mathbf{J_g}(\mathbf{x})]_{ij}}{\partial \boldsymbol{\theta_k}} \tag{3.26}$$

where $[\mathbf{J_g}(\mathbf{x})]_{ij}$ represents the $(i,j)^{\text{th}}$ element of the matrix $\mathbf{J_g}(\mathbf{x})$.

The Jacobian of the separating function is formulated as

$$\begin{aligned} \mathbf{J_g}(\mathbf{x}) &= \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}} = [\frac{\partial g_i(\mathbf{x})}{\partial x_j}] = \boldsymbol{\Theta} \frac{\partial \mathbf{k}(\mathbf{x})}{\partial \mathbf{x}} \\ &= \boldsymbol{\Theta} [\frac{\partial k_i(\mathbf{x})}{\partial x_j}] = \boldsymbol{\Theta} \mathbf{K}(\mathbf{x}) \end{aligned} \tag{3.27}$$

where by $[\frac{\partial g_i(\mathbf{x})}{\partial x_j}]$ (respectively $[\frac{\partial k_i(\mathbf{x})}{\partial x_j}]$) we mean the matrix whose $(i,j)^{\text{th}}$ element is $\frac{\partial g_i(\mathbf{x})}{\partial x_j}$ (respectively $\frac{\partial k_i(\mathbf{x})}{\partial x_j}$) and $\mathbf{K}(\mathbf{x}) \in \mathbb{R}^{P\times m}$ is the Jacobian of the vector $\mathbf{k}(\mathbf{x})$ of the kernels. From (3.27) we have

$$[\mathbf{J_g}(\mathbf{x})]_{ij} = \sum_{p=1}^{P} \mathbf{\Theta}_{ip} k_{pj} = \boldsymbol{\theta_i}^{\dagger} \mathbf{k}_j \tag{3.28}$$

where $\mathbf{k}_j$ is the j^{th} column of $\mathbf{K}$. It can be easily seen from (3.28) that for any $1 \leq i \leq n$, $[\mathbf{J_g}(\mathbf{x})]_{ij}$ only depends on $\boldsymbol{\theta_i}$. In other words

$$\frac{\partial [\mathbf{J_g}(\mathbf{x})]_{ij}}{\partial \boldsymbol{\theta_k}} = \begin{cases} \mathbf{k}_j & i = k \\ 0 & i \neq k \end{cases} \tag{3.29}$$

for $k = 1, \ldots, n$.

Therefore (3.26) for any $1 \leq k \leq n$ can be rewritten as

$$\frac{\partial \, \mathrm{I}(\dot{\mathbf{y}})}{\partial \boldsymbol{\theta_k}} = \sum_{j=1}^{m} \frac{\partial \, \mathrm{I}(\dot{\mathbf{y}})}{\partial [\mathbf{J_g}(\mathbf{x})]_{kj}} \mathbf{k}_j = \left[\left[\frac{\partial \, \mathrm{I}(\dot{\mathbf{y}})}{\partial \mathbf{J_g}(\mathbf{x})} \right]_{k^{\text{th}} \text{ row}} \mathbf{K}^{\dagger}(\mathbf{x}) \right]^{\dagger} \tag{3.30}$$

or equivalently

$$\left[\frac{\partial \, \mathrm{I}(\dot{\mathbf{y}})}{\partial \boldsymbol{\theta_k}} \right]^{\dagger} = \frac{\partial \, \mathrm{I}(\dot{\mathbf{y}})}{\partial \boldsymbol{\theta_k}^{\dagger}} = \left[\frac{\partial \, \mathrm{I}(\dot{\mathbf{y}})}{\partial \mathbf{J_g}(\mathbf{x})} \right]_{k^{\text{th}} \text{ row}} \mathbf{K}^{\dagger}(\mathbf{x}). \tag{3.31}$$

Stacking (3.31) for $k = 1, \ldots, n$ on top of each other yields to the derivative of $\mathrm{I}(\dot{\mathbf{y}})$ with respect to the parameters which can be written as

$$\underbrace{\frac{\partial \, \mathrm{I}(\dot{\mathbf{y}})}{\partial \mathbf{\Theta}}}_{n\times P} = \underbrace{\frac{\partial \, \mathrm{I}(\dot{\mathbf{y}})}{\partial \mathbf{J_g}(\mathbf{x})}}_{n\times m} \underbrace{\mathbf{K}^{\dagger}(\mathbf{x})}_{m\times P}. \tag{3.32}$$

In fact, considering (3.27), the last equation (3.32) could also been achieved by directly applying the chain rule for matrices.

3.1.4.3 Final Computations

Keeping (3.32) in mind, let us formulate $\partial \, \mathrm{I}(\dot{\mathbf{y}})/\partial \mathbf{J_g}(\mathbf{x})$. For the rest of the section, we exploit the assumption that the number of the observations is equal to the number of sources, i.e. the determined case ($m = n$).

Note that $\partial \mathrm{I}(\dot{\mathbf{y}})/\partial \mathbf{J_g}(\mathbf{x})$ is formulated similar to the linear BSS case. Considering

$$\mathrm{I}(\dot{\mathbf{y}}) = \left(\sum_{i=1}^{n} \mathrm{H}(\dot{y}_i)\right) - \mathrm{H}(\dot{\mathbf{y}}), \tag{3.33}$$

the partial derivative of both terms should be calculated with respect to the matrix $\mathbf{J_g}(\mathbf{x})$.

Considering (3.11) and (3.12), one can conclude

$$\frac{\partial \mathrm{H}(\dot{y}_i)}{\partial [\mathbf{J_g}]_{kj}} = \begin{cases} 0 & k \neq i \\ -\mathbb{E}\left\{ \frac{\frac{\partial}{\partial [\mathbf{J_g}]_{kj}} \rho_{\dot{Y}_i}(\dot{y}_i)}{\rho_{\dot{Y}_i}(\dot{y}_i)} \right\} & k = i \end{cases}. \tag{3.34}$$

Thus for any $1 \leq k = i \leq n$

$$\frac{\partial \mathrm{H}(\dot{y}_i)}{\partial [\mathbf{J_g}]_{ij}} = -\mathbb{E}\left\{ \frac{\rho'_{\dot{Y}_i}(\dot{y}_i)}{\rho_{\dot{Y}_i}(\dot{y}_i)} \frac{\partial \dot{y}_i}{\partial [\mathbf{J_g}]_{ij}} \right\} = -\mathbb{E}\left\{ \frac{\rho'_{\dot{Y}_i}(\dot{y}_i)}{\rho_{\dot{Y}_i}(\dot{y}_i)} \dot{x}_j \right\} = \mathbb{E}\{\Psi_i(\dot{y}_i)\dot{x}_j\} \tag{3.35}$$

where $(\cdot)'$ denotes the derivative with respect to the input argument, the last equation is a result of (3.12) and for $i = 1, \ldots, n$, $\Psi_i(\dot{y}_i)$ is the score function of $\dot{y}_i$ defined as

$$\Psi_i(\dot{y}_i) \triangleq \frac{-\rho'_{\dot{Y}_i}(\dot{y}_i)}{\rho_{\dot{Y}_i}(\dot{y}_i)} = -\big[\ln(\rho_{\dot{Y}_i}(\dot{y}_i))\big]'. \tag{3.36}$$

Therefore,

$$\sum_{i=1}^{n} \frac{\partial \mathrm{H}(\dot{y}_i)}{\partial \mathbf{J_g}} = \mathbb{E}\{\boldsymbol{\Psi}(\dot{\mathbf{y}})\dot{\mathbf{x}}^{\dagger}\} \tag{3.37}$$

where $\boldsymbol{\Psi}(\cdot)$ represents the component-wise score function defined as

$$\boldsymbol{\Psi}(\dot{\mathbf{y}}) = \begin{bmatrix} \Psi_1(\dot{y}_1) \\ \Psi_2(\dot{y}_2) \\ \vdots \\ \Psi_n(\dot{y}_n) \end{bmatrix}. \tag{3.38}$$

Regarding the second term of the right side of (3.33), we have

$$\begin{aligned} \mathrm{H}(\dot{\mathbf{y}}) &= -\mathbb{E}\{\ln \rho_{\dot{\mathbf{Y}}}(\dot{\mathbf{y}})\} = -\mathbb{E}\{\ln \frac{\rho_{\dot{\mathbf{X}}}(\dot{\mathbf{x}})}{|\det(\mathbf{J_g})|}\} \\ &= \ln|\det(\mathbf{J_g})| - \mathbb{E}\{\ln \rho_{\dot{\mathbf{X}}}(\dot{\mathbf{x}})\} \end{aligned} \tag{3.39}$$

which is concluded from the fact that in (3.12), assuming that the nonlinear mixing function is invertible, $\rho_{\dot{\mathbf{Y}}}(\dot{\mathbf{y}}) = \rho_{\dot{\mathbf{X}}}(\dot{\mathbf{x}})/|\det(\mathbf{J_g})|$. Thus

$$\frac{\partial \mathrm{H}(\dot{\mathbf{y}})}{\partial \mathbf{J_g}} = \frac{\partial \ln|\det(\mathbf{J_g})|}{\partial \mathbf{J_g}} = \mathbf{J_g}^{-\dagger} \tag{3.40}$$

where $\mathbf{J_g}^{-\dagger} = (\mathbf{J_g}^{-1})^{\dagger} = (\mathbf{J_g}^{\dagger})^{-1}$. It should be noted that, as mentioned earlier in Section 3.1.3, according to the inverse function theorem [Spivak, 1965], the invertibility of the function $\mathbf{f}$ ends to the non-singularity of its Jacobian $\mathbf{J_f}$ and the nice result that the Jacobian of its inverse is equal to the inverse of its Jacobian ($\mathbf{J_f}^{-1} = \mathbf{J_{f^{-1}}}$). Eq. (3.40) is a special case of a theorem[1] in linear algebra and matrix calculations [Petersen et al., 2008, Section 2.1.2].

Using (3.37) and (3.40) in (3.33) ends to

$$\frac{\partial \mathrm{I}(\dot{\mathbf{y}})}{\partial \mathbf{J_g}(\mathbf{x})} = \mathbb{E}\{\mathbf{\Psi}(\dot{\mathbf{y}})\dot{\mathbf{x}}^{\dagger}\} - \mathbf{J_g}^{-\dagger}(\mathbf{x}). \tag{3.41}$$

Finally, substituting (3.41) for computing the derivatives of (3.32) leads to

$$\frac{\partial \mathrm{I}(\dot{\mathbf{y}})}{\partial \mathbf{\Theta}} = \Big(\mathbb{E}\{\mathbf{\Psi}(\dot{\mathbf{y}})\dot{\mathbf{x}}^{\dagger}\} - \mathbf{J_g}^{-\dagger}(\mathbf{x})\Big)\mathbf{K}^{\dagger}(\mathbf{x}) \tag{3.42}$$

$$= \Big(\mathbb{E}\{\mathbf{\Psi}(\dot{\mathbf{y}})\dot{\mathbf{x}}^{\dagger}\} - \big(\mathbf{\Theta}\mathbf{K}(\mathbf{x})\big)^{-\dagger}\Big)\mathbf{K}^{\dagger}(\mathbf{x}) \tag{3.43}$$

where the last equation comes from (3.27). It should be noted the $\mathbf{\Theta}\mathbf{K}(\mathbf{x})$ is supposed to be invertible as far as the conditions of choosing kernel functions (orthonormality and $P \geq n$) are satisfied.

Finally, the update rule of the steepest descent algorithm (3.24) will become

$$\mathbf{\Theta} \;\leftarrow\; \mathbf{\Theta} - \mu\Big(\mathbb{E}\{\mathbf{\Psi}(\dot{\mathbf{y}})\dot{\mathbf{x}}^{\dagger}\} - \big(\mathbf{\Theta}\mathbf{K}(\mathbf{x})\big)^{-\dagger}\Big)\mathbf{K}^{\dagger}(\mathbf{x}). \tag{3.44}$$

To conclude, given a parametric model for the separating matrix as (3.22), Eq. (3.44) proposes an update rule for the parameters based on minimizing the mutual information between the derivatives of the outputs.

Nonetheless, in the rest of this chapter we consider general nonlinear functions, hence no parametric model is assumed for either the mixing or the separating functions. As a consequence, a general non-parametric approach will be proposed that is *not* based on mathematical derivations of the current section, but is based on locally solving linear BSS for the derivatives.

[1] $\frac{\partial}{\partial \mathbf{X}} \det(\mathbf{AXB}) = \det(\mathbf{AXB})\mathbf{X}^{-\dagger}$.

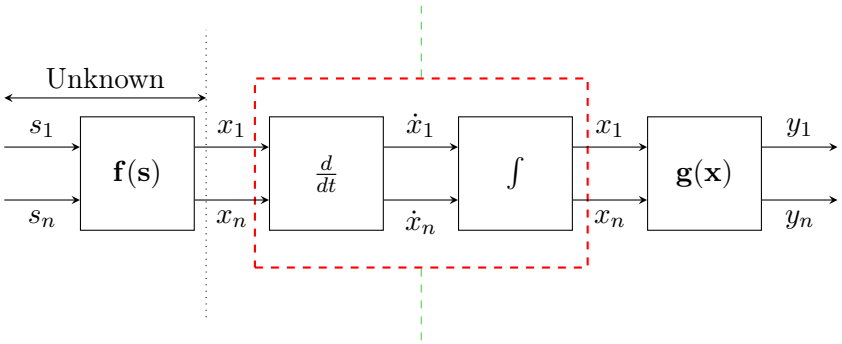


Figure 3.1: Nonlinear BSS problem alternative model

3.1.5 The Proposed General Approach

In order to get $\dot{\mathbf{x}}$, a component-wise derivative operator should be applied on the output of the mixing function $\mathbf{f}(\mathbf{s})$ of Fig. 1.1. Then, in order to cancel the effect of the differentiation operator (so that the separating function $\mathbf{g}(\cdot)$ in Fig. 1.1 remains unchanged), an integration operator needs to be added right after the differentiation operator. This will lead to the system which is depicted in Fig. 3.1.

Therefore, the problem (3.20), i.e. finding a nonlinear mapping $\mathbf{g}$ such that $\mathbf{g} \circ \mathbf{f} = \mathbf{c}$ is a nonlinear copy, can be equivalently written as

$$\text{find} \quad \mathbf{g} \qquad \text{s.t.} \qquad \mathbf{g} \circ \mathbf{d}^{-1} \circ \mathbf{d} \circ \mathbf{f} = \mathbf{c} \tag{3.45}$$

where $\mathbf{c}$ is a nonlinear copy function and $\mathbf{d}$ and $\mathbf{d}^{-1}$ are the component-wise differentiation and integration operators respectively. For the reason of homogeneity in expressions, we use the same notation as functions for operators even though it is not mathematically accurate. In fact, it must be noted that $\mathbf{d}^{-1} \circ \mathbf{d}$ is not necessarily equal to identity function because the result of integration is not unique and it could be added by any constant: $\mathbf{d}^{-1} \circ \mathbf{d} \circ \mathbf{f} = \mathbf{f} + cte$. However, since $\mathbf{d}$ and $\mathbf{d}^{-1}$ operate component-wise, applying them may just add a constant value to each signal, which does not affect the proposed framework.

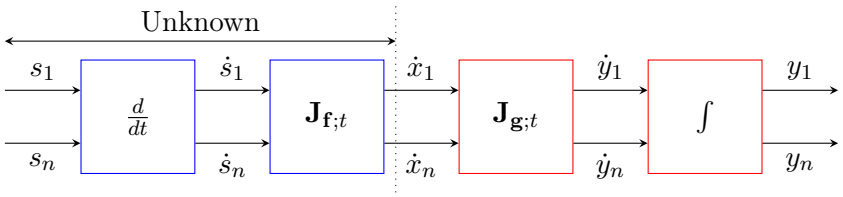


Figure 3.2: Transforming the nonlinear BSS problem model to the linear time-variant one

However, recalling (3.12)

$$\dot{\mathbf{x}} = \mathbf{J}_{\mathbf{f};t}(\mathbf{s})\dot{\mathbf{s}}, \tag{3.46}$$

the derivatives of the observations are locally linear mixtures of the derivatives of the sources, i.e. in each small neighborhood, the derivatives of the sources are linearly mixed through an approximately constant matrix. It means that they can be achieved by mixing the derivatives of sources via the Jacobian matrix of the nonlinear mixing function. In other words, considering (3.12), each half of this new model (which is nonlinear) can be replaced by an equivalent one (which is locally linear) shown in Fig. 3.2.

Mathematically speaking, denoting $\mathbf{J}_{\mathbf{f};t} = \partial\mathbf{f}/\partial\mathbf{s}$ and $\mathbf{J}_{\mathbf{g};t} = \partial\mathbf{g}/\partial\mathbf{x}$ the Jacobian matrices of the mixing function $\mathbf{f}$ and separating function $\mathbf{g}$ respectively, the equivalence of the systems of Fig. 3.1 and Fig. 3.2 can be written as

$$\begin{cases} \mathbf{d} \circ \mathbf{f} \equiv \mathbf{J}_{\mathbf{f};t} \circ \mathbf{d} \\ \mathbf{g} \circ \mathbf{d}^{-1} \equiv \mathbf{d}^{-1} \circ \mathbf{J}_{\mathbf{g};t} \end{cases} . \tag{3.47}$$

This equation says that instead of taking derivatives of a mixture of sources (i.e. $\mathbf{d}\circ\mathbf{f}$), one can equivalently mix derivatives of the sources via the Jacobian of the mixing function (i.e. $\mathbf{J}_{\mathbf{f};t} \circ \mathbf{d}$).

Then, replacing $\mathbf{d}\circ\mathbf{f}$ and $\mathbf{g}\circ\mathbf{d}^{-1}$ in (3.45) with their equivalents in (3.47), the nonlinear BSS problem becomes

$$\forall t, \text{ find } \mathbf{J}_{\mathbf{g};t} \qquad \text{s.t.} \qquad \mathbf{d}^{-1} \circ \mathbf{J}_{\mathbf{g};t} \circ \mathbf{J}_{\mathbf{f};t} \circ \mathbf{d} = \mathbf{c}. \tag{3.48}$$

This new model (depicted in Fig. 3.2) will be used for a discussion on the separability and for proposing an algorithm.

Regarding (3.48) and Fig. 3.2, the goal is to find a linear time-variant system $\mathbf{J}_{\mathbf{g};t}$ such that each of the output signals $y_1(t), \ldots, y_n(t)$ is a function of only one of the sources, hence $\mathbf{y}$ is a nonlinear copy of the sources.

By left-multiplying both sides of (3.48) by $\mathbf{d}$, and right-multiplying them by $\mathbf{d}^{-1}$, we will have

$$\mathbf{d} \circ \mathbf{d}^{-1} \circ \mathbf{J}_{\mathbf{g};t} \circ \mathbf{J}_{\mathbf{f};t} \circ \mathbf{d} \circ \mathbf{d}^{-1} = \mathbf{d} \circ \mathbf{c} \circ \mathbf{d}^{-1} \tag{3.49}$$

$$\Rightarrow \mathbf{J}_{\mathbf{g};t} \circ \mathbf{J}_{\mathbf{f};t} = \mathbf{d} \circ \mathbf{c} \circ \mathbf{d}^{-1} = \mathbf{c}_1 \tag{3.50}$$

where the last equation comes from the fact that $\mathbf{c}$ is a nonlinear copy function and, therefore, in combination with $\mathbf{d}$ and $\mathbf{d}^{-1}$ makes another nonlinear copy function named $\mathbf{c}_1$. As a consequence, the basic problem (3.20) is equivalent to

$$\forall\ t \quad \text{find} \quad \mathbf{J}_{\mathbf{g};t} \qquad \text{s.t.} \qquad \mathbf{J}_{\mathbf{g};t} \circ \mathbf{J}_{\mathbf{f};t} = \mathbf{c}_1 \tag{3.51}$$

where $\mathbf{c}_1$ is a nonlinear copy function. This is a traditional linear BSS problem where the mixing matrix is not constant along time, and can be solved via existing *adaptive* linear BSS methods (probably, with some modifications). As a conclusion, any nonlinear BSS problem is equivalent to a time-varying linear one and if the linear problem is solved correctly, the nonlinear problem will be solved as well.

It is worth adding two remarks which help better understanding the proposed concept:

Firstly, the local linear mixing $\mathbf{J}_{\mathbf{f};t}$ and separating $\mathbf{J}_{\mathbf{g};t}$ matrices are the Jacobian matrices of the nonlinear mixing $\mathbf{f}$ and separating $\mathbf{g}$ functions, respectively. Despite the indeterminacies in reconstructing the sources, it is obvious from Fig. 3.2 that the matrix $\mathbf{J}_{\mathbf{g};t}$ should be the inverse of the matrix $\mathbf{J}_{\mathbf{f};t}$. Actually, as mentioned in Section 3.1.3, the inverse of the Jacobian of a function is the Jacobian of the inverse function [Spivak, 1965]. This can also be easily shown by writing the equivalency equations of the right half of the systems of Figs. 3.1 and 3.2.

Secondly, the Jacobian matrix of a nonlinear function at each point is the best linear approximation of it at that point. Thus, the proposed approach could also be derived by linearly approximating the nonlinear function via Taylor expansion as

$$\begin{aligned} & \mathbf{x}(t) = \mathbf{f}(\mathbf{s}(t)) \Rightarrow \\ \forall t \;\; & \mathbf{x}(t+\epsilon) = \mathbf{x}(t) + \frac{\partial \mathbf{f}}{\partial \mathbf{s}}(\mathbf{s}(t+\epsilon) - \mathbf{s}(t)) + \mathrm{o}(\epsilon) && (3.52) \\ \Rightarrow \;\; & \mathbf{x}(t+\epsilon) - \mathbf{x}(t) \approx \mathbf{J}_{\mathbf{f};t}(\mathbf{s})\Big|_{\mathbf{s}=\mathbf{s}(t)} (\mathbf{s}(t+\epsilon) - \mathbf{s}(t)) && (3.53) \\ \Rightarrow \;\; & \Delta_{\mathbf{x}}(t) \approx \mathbf{J}_{\mathbf{f};t}(\mathbf{s})\Big|_{\mathbf{s}=\mathbf{s}(t)} \Delta_{\mathbf{s}}(t), && (3.54) \end{aligned}$$

where $\mathrm{o}(\epsilon)$ represents higher-order terms and $\Delta_{\mathbf{x}}(t)$ and $\Delta_{\mathbf{s}}(t)$ are the differences (increments) of the observation and source vectors respectively.

Eq. (3.54) can also be considered as a discrete-time approximation of (3.12) using the *difference* instead of the *derivative*. Nevertheless, the proposed framework can also be understood as the (local) linear approximation of the nonlinear mixing function at each point, and trying to separate the sources using adaptive linear BSS methods.

3.2 Proposed Algorithms

It follows from Fig. 3.2 that

$$\dot{\mathbf{y}}(t) = \mathbf{J}_{\mathbf{g};t}(\mathbf{x}(t))\dot{\mathbf{x}}(t) = \mathbf{J}_{\mathbf{g};t}(\mathbf{x}(t))\mathbf{J}_{\mathbf{f};t}(\mathbf{s}(t))\dot{\mathbf{s}}(t). \tag{3.55}$$

Therefore, it is necessary and sufficient for the separation to find a matrix $\mathbf{J}_{\mathbf{g};t}(\mathbf{x}(t))$ such that the off-diagonal elements of $\mathbf{J}_{\mathbf{g};t}(\mathbf{x}(t))\mathbf{J}_{\mathbf{f};t}(\mathbf{s}(t))$ are zero everywhere and its diagonal elements are nonlinear copy functions.

In this section, we are going to propose algorithms in order to perform nonlinear BSS based on the proposed idea. To this end, firstly an adaptive linear BSS method is reviewed in subsection 3.2.1, which plays an important role in the proposed algorithms. In this subsection, the necessity of utilizing an adaptive algorithm is highlighted and its exact formulation is provided.

Then a basic algorithm is proposed in subsection 3.2.2 derived from the sequencing steps of the mentioned approach of Section 3.1. Afterwards, in subsection 3.2.3, the main problem of the proposed preliminary algorithm is discussed and addressed by nonlinear regression of the separating function. Finally, in subsection 3.2.4 a modified algorithm is proposed employing the "Nonlinear Regression" technique.

3.2.1 Adaptive Linear BSS (Normalized EASI)

An adaptive BSS algorithm is an algorithm whose estimation of mixing and/or separating matrix is on-line, i.e. adjusted by observing each new sample. Generally speaking, algorithms of this kind start from an initial estimation (which can be randomly generated) and then update the estimation iteratively by receiving each sample. Normalized EASI (Equivariant Adaptive Separation via Independence) [Cardoso and Laheld, 1996] is an adaptive BSS algorithm that is based on the statistical independence of the sources. This powerful real-time algorithm is used in this work as the adaptive linear BSS method for estimating the $\mathbf{J}_{\mathbf{g};t}$ matrix, which cancels the mixture $\mathbf{J}_{\mathbf{f};t}$. In this purpose, components of $\dot{\mathbf{s}}$ should be statistically independent. In other words, while the independence of the derivatives of the sources (assumption 7) is necessary for the ICA, using other algorithms might impose other assumptions on the sources.

Since the mixing matrix in (3.55) (i.e. $\mathbf{J}_{\mathbf{f};t}$) changes along time, an *adaptive* technique needs to be utilized to perform the linear BSS (so that it can follow the variations of the inverse $\mathbf{J}_{\mathbf{g};t}$). Benefiting from the equivariancy (i.e. its performance does not depend on the condition number of the mixing matrix), good convergence rate and low computational cost of Normalized EASI [Cardoso and Laheld, 1996], it has been used as the adaptive linear BSS algorithm in our work.

The update formula of the separating matrix $\mathbf{J}_{\mathbf{g};t}$ according to this algo-

rithm will be as

$$\mathbf{J}_{\mathbf{g};t+1} = \mathbf{J}_{\mathbf{g};t} - \lambda_t \Big[\frac{\mathbf{y}(t)\mathbf{y}(t)^\dagger - \mathbf{I}}{1 + \lambda_t \mathbf{y}(t)^\dagger \mathbf{y}(t)} + \frac{\mathfrak{h}(\mathbf{y}(t))\mathbf{y}(t)^\dagger - \mathbf{y}(t)\mathfrak{h}(\mathbf{y}(t))^\dagger}{1 + \lambda_t |\mathbf{y}(t)^\dagger \mathfrak{h}(\mathbf{y}(t))|} \Big] \mathbf{J}_{\mathbf{g};t} \quad (3.56)$$

where λ_t is a sequence of positive adaptation steps and $\mathfrak{h}(\cdot)$ is an arbitrary component-wise (n-dimensional) nonlinear function. For a more detailed discussion on the choice of the components $\mathfrak{h}_i(\cdot)$ of $\mathfrak{h}(\cdot)$, the reader is invited to refer to [Cardoso and Laheld, 1996].

Plainly, at each iteration, (3.56) is followed by an update of the output vector as

$$\dot{\mathbf{y}}(t+1) = \mathbf{J}_{\mathbf{g};t+1}\ \dot{\mathbf{x}}(t+1). \quad (3.57)$$

3.2.2 Preliminary Algorithm

As mentioned earlier, assuming $\mathbf{J}_{\mathbf{g};t}(\mathbf{x}(t))$ in (3.55) is varying slowly enough such that it remains almost constant in the temporal neighborhood of each point $\mathbf{x}(t)$, a preliminary algorithm can be suggested simply as locally solving linear BSS problems at all time instants.

Accordingly, the first algorithm, called Adaptive Algorithm for Time-Variant Linear mixtures (AATVL), is sketched in Algorithm 1, where in lines (2) to (9), EASI or any other *adaptive* linear BSS technique can be employed.

The main problem with this algorithm is the issue of convergence: it always needs to be updated at each new sample of observations. In conventional applications of Normalized EASI, where the mixing matrix is assumed to be constant, after a number of iterations the algorithm (hopefully) converges to the exact separating matrix. However, in our case where $\mathbf{J}_{\mathbf{f};t}$ varies from one sample to another, the algorithm should not only estimate the exact separating matrix $\mathbf{J}_{\mathbf{g};t}$ at each sample, but it should also track the variations of $\mathbf{J}_{\mathbf{f};t}$ along time. In linear BSS, Normalized EASI should converge to a steady target (i.e. the exact separating matrix), while in our problem it

Algorithm 1 Adaptive Algorithm for Time-Variant Linear mixtures (AATVL)

$\dot{\mathbf{x}} \leftarrow$ Derivative (difference) of $\mathbf{x}$
procedure ADAPTIVE LINEAR BSS METHOD ($\dot{\mathbf{x}}(t)$)
    $\mathbf{J}_{\mathbf{g};0} \leftarrow$ Random Initialization
    $\dot{\mathbf{y}}(0) = \mathbf{J}_{\mathbf{g};0}\ \dot{\mathbf{x}}(0)$
    for $t = 0, \ldots, T-1$ **do**
        $\mathbf{J}_{\mathbf{g};t+1} \leftarrow$ Update by Eq. (3.56)
        $\dot{\mathbf{y}}(t+1) \leftarrow$ Update by Eq. (3.57)
    end for
end procedure
$\mathbf{y} \leftarrow$ Integral of $\dot{\mathbf{y}}$

needs to converge to a moving one and track it. So the convergence issue is much more severe than the classic linear problem.

It is worth noting that the variations of $\mathbf{J}_{\mathbf{f};t}(\mathbf{s}(t))$ depend on both the nonlinearity of the mixing model $\mathbf{f}(\cdot)$ and the dynamics of the sources $\mathbf{s}(t)$. Thus, even if the nonlinear mixing function $\mathbf{f}(\cdot)$ is smooth, bursty sources may lead to bursty changes in the mixing values, and consequently, the separating matrix cannot be tracked by the separating algorithm. This is the reason why the proposed approach needs both time-invariance of the mixture and coloration of the sources (assumptions 4 and 6) to impose the smoothness on $\mathbf{J}_{\mathbf{f};t}(\mathbf{s}(t))$ along time.

Another issue, which makes the convergence problem even more severe, is that the output of this adaptive linear BSS algorithm is going to be *integrated* through a following step to estimate the separated sources (see Fig. 3.2). This integration will propagate the estimation error to the other samples as well. As a consequence, the AATVL algorithm (algorithm 1) needs to be modified.

3.2.3 Nonlinear Regression

In this subsection, the main problem of the proposed preliminary algorithm (i.e. convergence) is addressed by a nonlinear regression technique. The concept is explained in details providing 2 different methods (subsections 3.2.3.1 and 3.2.3.2). The second method (which is actually used in the modified algorithm 2) is supported by a simulated preliminary example and a discussion on its performance.

The convergence problem of the algorithm 1 is because it does not exploit the time-invariance and smoothness of the mixing function $\mathbf{f}$. In fact, the original nonlinearity $\mathbf{f}$, and its inverse $\mathbf{g}$, are time-invariant. Therefore the dependence of $\mathbf{J}_{\mathbf{f};t}$ (respectively $\mathbf{J}_{\mathbf{g};t}$) on $\mathbf{s}$ (respectively $\mathbf{x}$) is not time-varying.

In other words, $\mathbf{s}$ and $\mathbf{x}$ are themselves time-varying, and $\mathbf{J}_{\mathbf{f};t}$ and $\mathbf{J}_{\mathbf{g};t}$ are evaluated for sources and observations at successive times as

$$\mathbf{J}_{\mathbf{f};t}(\mathbf{s}(t)) = \frac{\partial \mathbf{f}}{\partial \mathbf{s}}(\mathbf{s})\Big|_{\mathbf{s}=\mathbf{s}(t)}, \tag{3.58}$$

$$\mathbf{J}_{\mathbf{g};t}(\mathbf{x}(t)) = \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\mathbf{x})\Big|_{\mathbf{x}=\mathbf{x}(t)}. \tag{3.59}$$

As a result, a modification on the algorithm 1 can be suggested by learning the nonlinear model of $\mathbf{J}_{\mathbf{g};t}(\mathbf{x})$ from its estimations at different samples (say $\hat{\mathbf{J}}_{\mathbf{g}}(\mathbf{x}(t))$ for $t = 1, \ldots, T$, the outputs of the adaptive linear BSS method). It should be noted that $\mathbf{J}_{\mathbf{g};t}(\mathbf{x})$ is an $n \times n$ matrix and contains n^2 nonlinear functions that should be learned in this approach.

For example, let $[\mathbf{J}_{\mathbf{g};t}(\mathbf{x})]_{i,j}$ denote the $(i,j)^{\text{th}}$ element of the separating matrix. In the "nonlinear regression" stage, we aim at estimating the nonlinear function $[\mathbf{J}_{\mathbf{g};t}(\mathbf{x})]_{i,j}$ from $[\hat{\mathbf{J}}_{\mathbf{g}}(\mathbf{x}(t))]_{i,j}$ for $t = 1, \ldots, T$. In the simplest case, it can be mathematically expressed as for all $1 \leq i, j \leq n$

$$\underset{[\mathbf{J}_{\mathbf{g};t}(\mathbf{x})]_{i,j}}{\text{minimize}} \sum_{t=1}^{T} \Big(\mathrm{d}_w^2([\hat{\mathbf{J}}_{\mathbf{g}}(\mathbf{x}(t))]_{i,j}, [\mathbf{J}_{\mathbf{g};t}(\mathbf{x})]_{i,j}) \Big) \tag{3.60}$$

where d_w^2 represents a weighted squared distance of a point and a manifold

defined as

$$\mathrm{d}_w^2(\cdot,\cdot) = \mathrm{d}^2(\cdot,\cdot) \times w(\mathrm{d}^2(\cdot,\cdot)) \tag{3.61}$$

and

$$\mathrm{d}^2([\hat{\mathbf{J}}_{\mathbf{g}}(\mathbf{x}(t))]_{i,j}, [\mathbf{J}_{\mathbf{g};t}(\mathbf{x})]_{i,j}) = \Big([\mathbf{J}_{\mathbf{g};t}(\mathbf{x})]_{i,j}\big|_{\mathbf{x}=\mathbf{x}(t)} - [\hat{\mathbf{J}}_{\mathbf{g}}(\mathbf{x}(t))]_{i,j}\Big)^2, \tag{3.62}$$

is the vertical distance of a sample point $[\hat{\mathbf{J}}_{\mathbf{g}}(\mathbf{x}(t))]_{i,j}$ from its corresponding point on the estimated nonlinear function $[\mathbf{J}_{\mathbf{g};t}(\mathbf{x})]_{i,j}\big|_{\mathbf{x}=\mathbf{x}(t)}$.

Since the error in the estimation $[\hat{\mathbf{J}}_{\mathbf{g}}(\mathbf{x}(t))]_{i,j}$ might be large for some samples (especially due to the convergence issue), there might be some *outliers* in the data. Although the outliers are supposed to be rare, due to the power of 2 in (3.62), they can highly affect the result of the manifold learning process. Consequently, using a *weighted* distance in (3.60) is essential in order to reduce the effect of the estimations that are too far from the learned manifold.

Using a weighted squared distance, long distances of outliers will be less weighted and their effect on the learned manifold will be limited. The weighting function is designed such that it is close to 1 for short distances and it tends to zero as the distance increases. As an example, Gaussian weighting function can be defined as

$$w(d^2) = e^{-\frac{d^2}{2\zeta^2}} \tag{3.63}$$

where ζ is a parameter which can be adjusted according to the data.

The optimization (3.60), where the cost function should be minimized with respect to a nonlinear manifold, can be performed using either a parametric model (when the nonlinear function is assumed to belong to a specific set of functions, e.g. polynomials) or a non-parametric one (utilizing an interpolation method like smoothing splines). One may also modify a dimension reduction technique (e.g. ISOMAP [Tenenbaum et al., 2000]) in order to solve (3.60).

3.2.3.1 Parametric Approach

In this approach, a parametric model for each $[\mathbf{J}_{\mathbf{g};t}(\mathbf{x})]_{i,j}$ is assumed and then the minimization of (3.60) is performed with respect to those parameters. In other words, we assume that each manifold is formulated as

$$[\mathbf{J}_{\mathbf{g};t}(\mathbf{x})]_{i,j} = Q_{i,j}(\mathbf{x}; \boldsymbol{\theta}_{i,j}) \tag{3.64}$$

where $\boldsymbol{\theta}_{i,j}$ is a vector of the parameters in nonlinear model of $[\mathbf{J}_{\mathbf{g};t}(\mathbf{x})]_{i,j}$.

For example, a second-order polynomial modeling can be assumed as

$$[\mathbf{J}_{\mathbf{g};t}(\mathbf{x})]_{i,j} = Q_{i,j}(\mathbf{x}; \boldsymbol{\theta}_{i,j}) = \mathbf{x}^{\dagger} A_{i,j} \mathbf{x} + \mathbf{b}_{i,j}^{\dagger} \mathbf{x} + c_{i,j} \tag{3.65}$$

where the vector of the parameters $\boldsymbol{\theta}_{i,j}$ consists of the $n \times n$ matrix $\mathbf{A}_{i,j}$, the n-dimensional vector $\mathbf{b}_{i,j}$, and the scalar $c_{i,j}$ (for each $[\mathbf{J}_{\mathbf{g};t}(\mathbf{x})]_{i,j}$ there are n^2+n+1 parameters in this model). One may suggest any other parametric model depending on either prior information about the mixing model (if it exists) or a general form which is able to model a wide range of nonlinear functions.

As a consequence, with this parametric model, (3.60) becomes

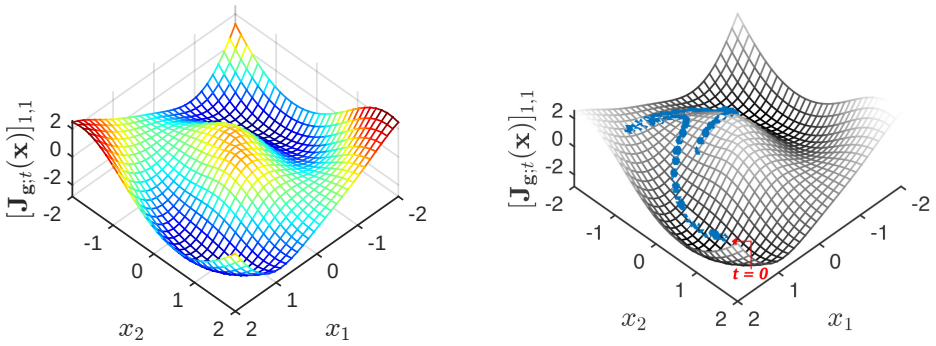
$$\underset{\boldsymbol{\theta}_{i,j}}{\text{minimize}} \sum_{t=1}^{T} \Big(\mathrm{d}_w^2([\hat{\mathbf{J}}_{\mathbf{g}}(\mathbf{x}(t))]_{i,j}, [\mathbf{J}_{\mathbf{g};t}(\mathbf{x})]_{i,j}) \Big) \tag{3.66}$$

where $\mathrm{d}_w^2([\hat{\mathbf{J}}_{\mathbf{g}}(\mathbf{x}(t))]_{i,j}, [\mathbf{J}_{\mathbf{g};t}(\mathbf{x})]_{i,j})$ can be calculated as a function of the parameters according to (3.61) and (3.62). Thus it can be solved, and the optimal parameter vectors $\boldsymbol{\theta}_{i,j}^{*}$ will let us formulate the $[\mathbf{J}_{\mathbf{g};t}(\mathbf{x})]_{i,j}$'s.

3.2.3.2 Non-Parametric Approach

The other approach proposed for nonlinear regression is non-parametric where no model for the nonlinearity is assumed. To this end, the nonlinear functions are learned by fitting curves using a smoothing method (e.g. smoothing splines [Reinsch, 1967]) to the estimations $[\hat{\mathbf{J}}_{\mathbf{g}}(\mathbf{x}(t))]_{i,j}$ for $t = 1, \ldots, T$.

In this work, *smoothing spline* [De Boor, 1978] is utilized as the smoothing method, for which the second order derivative of $[\mathbf{J}_{\mathbf{g};t}(\mathbf{x})]_{i,j}$ is added to the



(a) Pure function (b) 300 sample evaluations

Figure 3.3: The nonlinear function of $[\mathbf{J}_{\mathbf{g}}(\mathbf{x})]_{1,1}$ of (3.69) with respect to the observations

cost function (3.60) as a penalty term to impose the smoothness. In this method, there is a smoothing parameter, controlling the trade-off between fidelity to the data and roughness of the function estimate.

This method is explained via studying its performance on an example with a mixing function $\mathbf{f}$ as (2.12). This model is a rotation with the angle which depends to the norm of the source vector. So the inverse function $\mathbf{g}$ can be easily achieved by another rotation with the negative angle as

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} \cos\alpha(\mathbf{x}(t)) & \sin\alpha(\mathbf{x}(t)) \\ -\sin\alpha(\mathbf{x}(t)) & \cos\alpha(\mathbf{x}(t)) \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \tag{3.67}$$

where

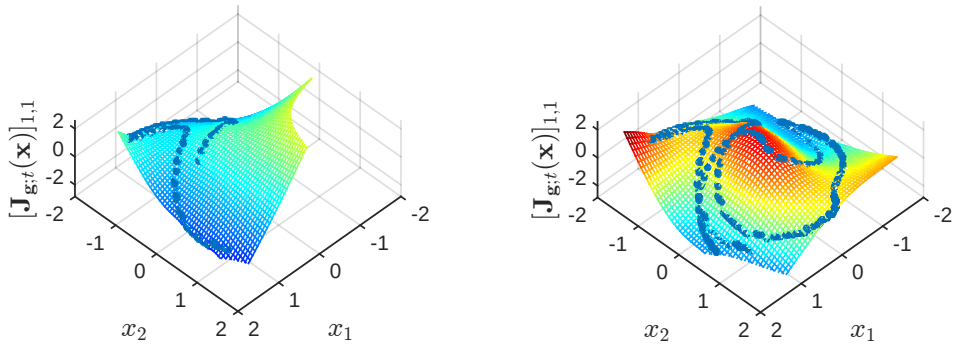
$$\alpha(\mathbf{x}(t)) = \alpha_0 + \gamma \times \sqrt{x_1^2(t) + x_2^2(t)}. \tag{3.68}$$

Therefore, the exact Jacobian $\mathbf{J}_{\mathbf{g}}(\mathbf{x})$ is calculated as

$$\mathbf{J}_{\mathbf{g}}(\mathbf{x}) = \begin{bmatrix} \cos\alpha(\mathbf{x}) & \sin\alpha(\mathbf{x}) \\ -\sin\alpha(\mathbf{x}) & \cos\alpha(\mathbf{x}) \end{bmatrix} \begin{bmatrix} 1 + x_2 \frac{\partial\alpha(\mathbf{x})}{\partial x_1} & x_2 \frac{\partial\alpha(\mathbf{x})}{\partial x_2} \\ -x_1 \frac{\partial\alpha(\mathbf{x})}{\partial x_1} & 1 - x_1 \frac{\partial\alpha(\mathbf{x})}{\partial x_2} \end{bmatrix}. \tag{3.69}$$

Now consider one of the elements of $\mathbf{J}_{\mathbf{g}}(\mathbf{x})$, say $[\mathbf{J}_{\mathbf{g}}(\mathbf{x})]_{1,1}$. In this example, $n = 2$ and the 2-dimensional nonlinear function $[\mathbf{J}_{\mathbf{g}}(\mathbf{x})]_{1,1}$ with respect to x_1 and x_2 (calculated in (3.69)) is depicted in Fig. 3.3a.

As an example, suppose that the sources $s_1(t)$ and $s_2(t)$ are integrals of a triangle signal (with the amplitude of 6 and the primitive period of 200π



(a) 300 samples (b) 700 samples

Figure 3.4: The estimated (learned) nonlinear model of $[\mathbf{J}_{\mathbf{g}}(\mathbf{x})]_{1,1}$ from 300 (Fig. 3.4a) and 700 (Fig. 3.4b) samples of observations. The circles are the outputs of the adaptive linear BSS method $[\hat{\mathbf{J}}_{\mathbf{g}}(\mathbf{x}(t))]_{1,1}$, and hyper-surface is the learned manifold using the introduced smoothing spline technique.

samples) and a sinusoidal signal (with the amplitude of 6 and the frequency of $\sqrt{3}/200\pi$ samples), respectively. The trajectory of the observation vector along time projected onto the 2-dimensional manifold of $[\mathbf{J}_{\mathbf{g}}(\mathbf{x})]_{1,1}$ for 300 time instants is plotted in Fig. 3.3b. It illustrates the changes in the value of $[\mathbf{J}_{\mathbf{g};t}(\mathbf{x})]_{1,1}$ over time. It is nice to note that as time passes, the observation vector will take different values, hence the whole range will be spanned by the time trajectory, which will result in having enough samples for learning the entire shape of the nonlinear function.

Fig. 3.4 shows the learned nonlinear model (the hyper-surface) given 300 and 700 samples of $[\hat{\mathbf{J}}_{\mathbf{g}}(\mathbf{x}(t))]_{1,1}$ using the smoothing spline technique. It can be seen that the learned nonlinear model from 700 samples based on smoothing spline is quite accurate in the region of interest, i.e. where samples are available.

The normalized Root Mean Squared (RMS) error in reconstruction of $[\mathbf{J}_{\mathbf{g}}(\mathbf{x})]_{1,1}$ in (3.69) with respect to the number of observation samples, i.e. the

error E_{nrms}, can be defined as

$$E_{nrms} = \frac{\left(\iint\limits_{|x_1|,|x_2| \leq M} \left([\mathbf{J_g}(\mathbf{x})]_{1,1} - [\hat{\mathbf{J}}_\mathbf{g}(\mathbf{x})]_{1,1} \right)^2 \right)^{\frac{1}{2}}}{\left(\iint\limits_{|x_1|,|x_2| \leq M} \left([\hat{\mathbf{J}}_\mathbf{g}(\mathbf{x})]_{1,1} \right)^2 \right)^{\frac{1}{2}}} \tag{3.70}$$

where $M = \max\big(\max(|x_1(t)|), \max(|x_2(t)|)\big)$ is the maximum range of variations of the observations. However, a more meaningful definition of the N-RMS error, say empirical error, is when it is calculated over the region of interest (where the observation vector spans) as

$$\tilde{E}_{nrms} = \frac{\left(\sum\limits_{t=1,\ldots,T} \left([\mathbf{J_g}(\mathbf{x}(t))]_{1,1} - [\hat{\mathbf{J}}_\mathbf{g}(\mathbf{x}(t))]_{1,1} \right)^2 \right)^{\frac{1}{2}}}{\left(\sum\limits_{t=1,\ldots,T} \left([\hat{\mathbf{J}}_\mathbf{g}(\mathbf{x}(t))]_{1,1} \right)^2 \right)^{\frac{1}{2}}}. \tag{3.71}$$

Fig. 3.5 shows how both $\tilde{E}_{nrms}$ and E_{nrms} decrease as the number of given samples increases. As it can be seen in this figure, the accuracy of the estimated model improves as the number of input samples grows, until a certain number at which the estimation is close enough to the correct model and the error does not decrease anymore. As expected, although the error on the region of interest is larger that the one on the whole region (which does not mathematically mean and depends on the simulation), it tends to zero after enough iterations when the nonlinearity is learned.

It should be added that the utilized algorithm in this example (smoothing splines) does not force the model to pass the input points. Nevertheless, depending on the application, other smoothing algorithms with different properties: more robust to noise, forcing to pass the points, etc., may be exploited. Such algorithms can include Kalman filter, kernel smoother, Laplacian smoothing, exponential smoothing, and so on.

3.2.4 Modified Algorithm

Employing the nonlinear regression idea introduced in Section 3.2.3 in combination with algorithm 1 leads to a second algorithm which outperforms

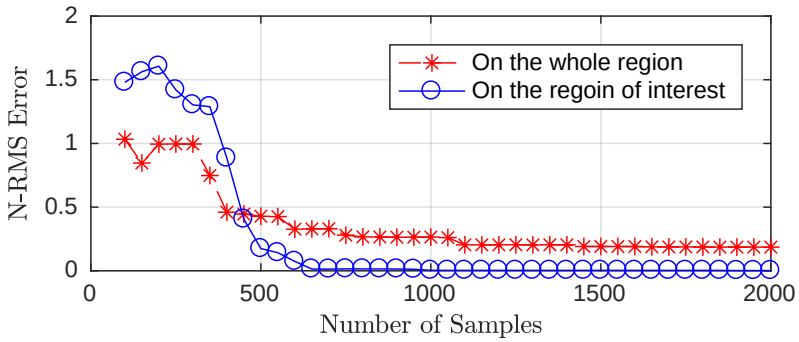


Figure 3.5: The N-RMS error of the estimation of the nonlinear model of $[\mathbf{J}_{\mathbf{g}}(\mathbf{x})]]_{1,1}$ with respect to the number of samples over 1) an $M \times M$ square (the dashed line) and 2) the region of interest in which the samples exist (the solid line)

the first one. This algorithm includes 2 steps: 1) an "Adaptive linear BSS" method for estimating $\hat{\mathbf{J}}_{\mathbf{g}}(\mathbf{x}(t))$ for $t = 1, \ldots, T$ and 2) a "Nonlinear Separation" process through which the nonlinear functions $\mathbf{J}_{\mathbf{g};t}(\mathbf{x})$ are learned by the proposed smoothing spline method and are used to separate the sources. Once the nonlinear functions $\mathbf{J}_{\mathbf{g};t}(\mathbf{x})$ are estimated, they are used for separating the derivatives of the sources.

The Batch Algorithm for Time-Invariant Nonlinear mixtures (BATIN) can thus be proposed as algorithm 2.

It should be finally noted that the Normalized EASI and the smoothing spline algorithms that are used in algorithm 2, could be replaced by other equivalent algorithms depending on the application.

3.3 Reconstruction Indeterminacies

Linear BSS methods generally suffer from ambiguities both in the order of the sources and their scales. On the other hand, as pointed earlier and will be explained in the following, the proposed framework in this chapter is based on the local linear approximation of the nonlinear mixture. So it is important

Algorithm 2 Batch Algorithm for Time-Invariant Nonlinear mixtures (BATIN)

$\dot{\mathbf{x}} \leftarrow$ Derivative (difference) of $\mathbf{x}$

Step Adaptive linear BSS:

procedure ADAPTIVE LINEAR BSS METHOD ($\dot{\mathbf{x}}(t)$)

  $\hat{\mathbf{J}}_{\mathbf{g}}(\mathbf{x}(0)) \leftarrow$ Random Initialization

  $\dot{\mathbf{y}}(0) = \hat{\mathbf{J}}_{\mathbf{g}}(\mathbf{x}(0))\ \dot{\mathbf{x}}(0)$

  for $t = 0, \ldots, T-1$ **do**

    $\hat{\mathbf{J}}_{\mathbf{g}}(\mathbf{x}(t+1)) \leftarrow$ Update by Eq. (3.56)

    $\dot{\mathbf{y}}(t+1) \leftarrow$ Update by Eq. (3.57)

  end for

end procedure

Step Nonlinear Separation:

procedure NONLINEAR REGRESSION ($\hat{\mathbf{J}}_{\mathbf{g}}(\mathbf{x}(t)), \mathbf{x}(t)$)

  $\mathbf{J}_{\mathbf{g};t}(\mathbf{x}) \leftarrow$ Smoothing Spline of $\hat{\mathbf{J}}_{\mathbf{g}}(\mathbf{x}(t))$

end procedure

for $t = 1, \ldots, T$ **do**

  $\dot{\mathbf{y}}(t) \leftarrow \mathbf{J}_{\mathbf{g};t}(\mathbf{x})\ \dot{\mathbf{x}}(t)$

end for

$\mathbf{y} \leftarrow$ Integral of $\dot{\mathbf{y}}$

to understand how these *local* permutation and scaling ambiguities perform *globally*.

3.3.1 PERMUTATION

Let us consider Theorem 1 related to ICA identifiability and the problem (3.51), i.e. finding $\mathbf{J}_{\mathbf{g};t}$ such that $\mathbf{J}_{\mathbf{g};t} \circ \mathbf{J}_{\mathbf{f};t} = \mathbf{c}_1$ is a nonlinear copy function. Thus ICA for linear BSS problem guarantees $\mathbf{J}_{\mathbf{g};t}\mathbf{J}_{\mathbf{f};t} = \mathbf{\Lambda}_t\mathbf{P}_t$ where $\mathbf{\Lambda}_t$ is a diagonal matrix, and $\mathbf{P}_t$ is a permutation matrix. So one could say that the permutation would potentially change as the algorithm progresses; i.e. linear ICA algorithm could converge to several instances of

this form, depending on the initialization. This seems to bring an issue about the alignment of permutations at different times. Considering the fact that the proposed algorithm is in the form of tracking algorithms, is there still a danger of permutation pattern change during the algorithm's course?

Mathematically speaking, the permutation matrix $\mathbf{P}_t$ might change over time which might cause the alignment issue. However, any change in the permutation matrix results in discontinuity (because of the structure of a permutation matrix which can take exactly one "1" value at each row and each column and the rest of the entries should be equal to zero). Consequently, if the derivatives of the observations are continuous functions of the state space coordinate (which is true in most realistic applications), $\mathbf{J}_{\mathbf{f};t}$ will be continuous, thus the continuity of $\mathbf{J}_{\mathbf{g};t}$ imposes the continuity (in the time/sample domain) of $\mathbf{P}_t$.

The only exception of the above argument is the case when two entries of the Jacobian matrix have simultaneous zero-crossings. At such instances, the two corresponding entries of the permutation matrix $\mathbf{P}_t$ may also swap values without violating the continuity of $\mathbf{J}_{\mathbf{g};t}$. This phenomenon has also been experienced in linear time-varying ICA (for instance using EASI with slowly moving or rotating sources), where the estimated sources smoothly approach to zero and instantaneous permutations occur between the extracted sources. In this work, we have ignored this very special case, for which a more detailed study on the alignment of local permutations can be proposed as a future study.

Therefore, as long as the local separating matrix $\mathbf{J}_{\mathbf{g};t}$ is estimated adaptively and continuously, the local permutation matrix $\mathbf{P}_t$ should also be continuous, hence constant along time. Therefore, in any neighbourhood of observation state space, there will always be a continuous separating solution which is unique, up to an arbitrary *global* permutation.

It is worth noting that a similar result is obtained for different frequencies in IVA (independent vector analysis) for convolutive mixtures, by considering joint source separation in different frequency bands, with continuity between

successive bands [Lee et al., 2007].

To summarize, since the local separating matrix $\mathbf{J}_{\mathbf{g};t}$ is estimated adaptively and continuously, the local permutation matrix should also change continuously. Therefore, *local* permutations in any neighbourhood of observations result in an arbitrary *global* permutation, and do not cause any issue about the alignment of permutations at successive time instants.

3.3.2 Scaling

A similar argument may also be asked regarding the scaling indeterminacy; even though two different time instants may correspond to the same input (i.e. $\mathbf{x}(t_1) = \mathbf{x}(t_2)$), may the corresponding linear algorithm convergence points, $\mathbf{J}_{\mathbf{g},t_1}$ and $\mathbf{J}_{\mathbf{g},t_2}$, be different according to initialization of the algorithm? In this regard, two issues should be distinguished:

1. The final estimated separating matrix at each observation point, i.e. "algorithm convergence point", is continuous. Since the "convergence points" of $\mathbf{J}_{\mathbf{g},t}(\mathbf{x})$ for different time instants corresponding to the same observation value make a *continuous* function with the estimations for neighbouring observation vectors, they cannot be different.

   Mathematically speaking, if for two time instants t_1 and t_2, $\mathbf{x}(t_1) = \mathbf{x}(t_2)$ and $\mathbf{J}_{\mathbf{g},t_1} \neq \mathbf{J}_{\mathbf{g},t_2}$, then there would be two different trajectories in the $\mathbf{x}$ domain by which we could approach to $\mathbf{J}_{\mathbf{g},t}(\mathbf{x}(t_1)) = \mathbf{J}_{\mathbf{g},t}(\mathbf{x}(t_2))$ getting different values

   $$\lim_{t \to t_1} \mathbf{J}_{\mathbf{g},t}(\mathbf{x}) \neq \lim_{t \to t_2} \mathbf{J}_{\mathbf{g},t}(\mathbf{x}). \tag{3.72}$$

   Thus the continuity of $\mathbf{J}_{\mathbf{g},t}(\mathbf{x})$ unifies the convergence points.

2. However, since the algorithm is adaptive and it performs just one iteration at each time instant, it does not necessarily results the "convergence point" depending on the initialization. This issue relates to the performance of the adaptive linear BSS method (not the proposed

framework) and depends on the smoothness and bandwidth of the nonlinear model and sources, respectively. Nevertheless, assuming that there are a sufficiently high number of observation samples, the small estimation errors in adaptive BSS process will be cancelled through the nonlinear regression procedure proposed in BATIN.

This could also be understood from another point of view. In a small neighbourhood of any particular value of the observation, the problem can be well approximated and can be exactly solved via a linear BSS technique with a scaling ambiguity (without the indeterminacy of the convergence point for different time instants). Combining this result throughout the observation domain and imposing the continuity and smoothness assumption leads to a global separating function which contains a global permutation and a smooth component-wise nonlinear function (because of the smoothly varying scaling) on each source.

Nevertheless, the amplitude-varying values of the scaling ambiguity on the whole domain of the signals cause a *component-wise* nonlinearity which cannot be resolved by the proposed algorithm, i.e. each output of the algorithm does depend on only one of the sources but with a time-varying scaling factor (i.e. a nonlinear function).

This indeterminacy in reconstructing the sources could also be seen from another point of view. Assume $\mathbf{u}(\cdot)$ is a component-wise nonlinear function as

$$\tilde{\mathbf{y}}(t) = \mathbf{u}(\mathbf{y}(t)) \tag{3.73}$$

such that

$$\forall\ 1 \leq k \leq n \quad \tilde{y}_k(t) = u_k(\mathbf{y}(t)) = \hat{u}_k(y_k(t)) \tag{3.74}$$

where $\hat{u}_k(\cdot)$ for $k = 1, \ldots, n$ are 1-dimensional $\mathbb{R} \to \mathbb{R}$ nonlinear functions.

Obviously, the Jacobian of a component-wise function is diagonal. As a consequence, if $\mathbf{J}_{\mathbf{g};t}$ satisfies (3.51), i.e. $\mathbf{J}_{\mathbf{g};t} \circ \mathbf{J}_{\mathbf{f};t} = \mathbf{c}_1$ is a nonlinear copy function, $\mathbf{J}_{\mathbf{u}\circ\mathbf{g}} = \mathbf{J}_{\mathbf{u}}\mathbf{J}_{\mathbf{g}}$ will satisfy (3.51) as well. Indeed, if a function $\mathbf{g}$ (resulting in $\mathbf{y}$ as the separated sources) is a separating function, the function

$\mathbf{u} \circ \mathbf{g}$ (resulting in $\tilde{\mathbf{y}}(t) = \mathbf{u}(\mathbf{y}(t))$) will also separate the sources. Therefore, the proposed approach may result in any component-wise nonlinear function of the sources.

3.4 Simulations

In this section, simulation results of both proposed algorithms for two different nonlinear functions are shown as a proof of concept. The data model, nonlinear functions, the parameters and the details of the simulations come in Section 3.4.1. Afterwards, the results of the simulations and their performance evaluations are reported in Section 3.4.2.

3.4.1 Simulated Data and Mixture Models

In the first example, consider the two-input two-output system of (2.12) as

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \cos\alpha(\mathbf{s}(t)) & -\sin\alpha(\mathbf{s}(t)) \\ \sin\alpha(\mathbf{s}(t)) & \cos\alpha(\mathbf{s}(t)) \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix} \tag{3.75}$$

where $\alpha(\mathbf{s}(t))$ is defined by the parametric model

$$\alpha(\mathbf{s}(t)) = \alpha_0 + \gamma \times \sqrt{s_1^2(t) + s_2^2(t)} \tag{3.76}$$

where α_0 and γ are some parameters.

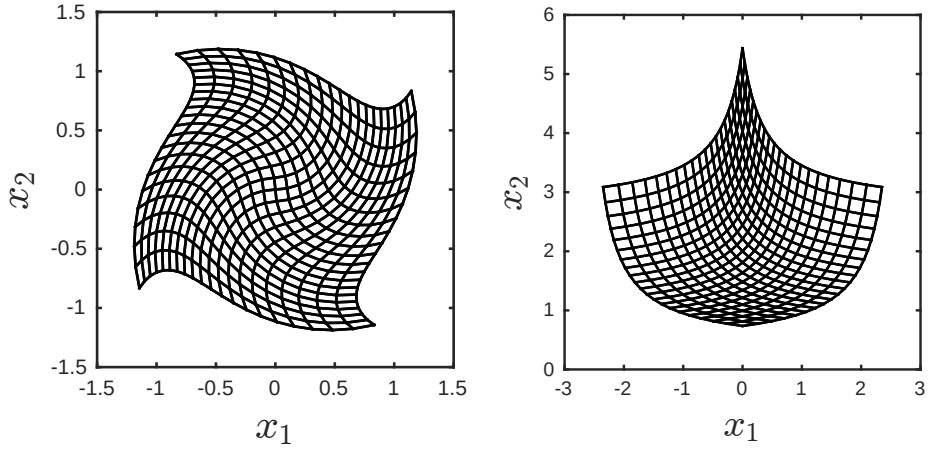
In our first simulation, (3.76) is considered for $\alpha_0 = 0$ and $\gamma = 1$.

Secondly, the proposed method is applied to another mixing model defined as

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \mathbf{f}(\mathbf{s}(t)) = \begin{bmatrix} e^{s_1(t)} - e^{s_2(t)} \\ e^{-s_1(t)} + e^{-s_2(t)} \end{bmatrix} \tag{3.77}$$

which is a nonlinear but invertible mixing model, as well as the first one.

The function mappings of the two simulated models are illustrated in Fig. 3.6: the figure shows how a regular grid in the input domain is transformed through the functions. As it can be understood from this figure as well as (2.12) and (3.77), both models are nonlinear but bijective (one-to-one) in the input range.



(a) Simulation 1 (b) Simulation 2

Figure 3.6: Illustration of the nonlinear mappings. a) the mapping follows model (3.75) and (3.76) for $\alpha_0 = 0$ and $\gamma = 1$ and b) the mapping follows model (3.77). In both figures, we represent the grid obtained by applying the nonlinear mapping (3.75) or (3.77) to the regular grid in the domain $[-1, +1] \times [-1, +1]$, and the input domain is mapped to nonlinear grids in the output domain which are shown.

In both simulations, the two sources that are mixed are the integrals of a sine wave

$$\dot{s}_1(t) = \sin(\sqrt{3}t/100) \;\Rightarrow\; s_1(t) \propto \int \sin(\sqrt{3}t/100)\,\mathrm{d}\,t \tag{3.78}$$

and a triangle wave

$$\dot{s}_2(t) = \mathrm{saw}(t/100) \;\Rightarrow\; s_2(t) \propto \int \mathrm{saw}(t/100)\,\mathrm{d}\,t \tag{3.79}$$

where $\mathrm{saw}(t)$ is defined as a sawtooth wave with period 2π passing through the points $(0, 0)$, $(\pi/2, 1)$, $(3\pi/2, -1)$ and $(2\pi, 0)$ (see Fig. 3.7).

The sources are chosen well-known simple signals with non-harmonically related frequencies to avoid any coherence, and satisfying assumptions on $\mathbf{s}$, and especially independence of the derivatives (assumption 7).

It should be noted that the integral can be practically approximated by either a recursive summation $s(t) = \Delta t\ \dot{s}(t) + s(t-1)$ or a continuous function estimation based on an interpolation method. Simulations (not

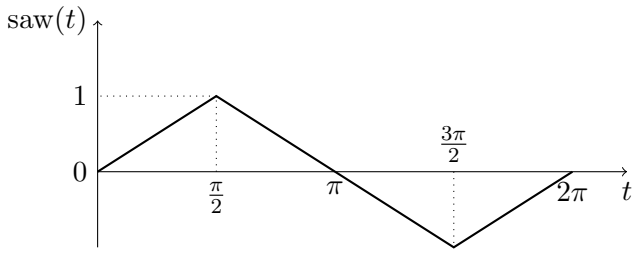


Figure 3.7: Illustration of a sawtooth signal saw(t)

presented in this work) show that these two approaches result in almost the same estimation. Thus the summation is used as an approximation of the integral everywhere.

The observations are then calculated by (3.75) and (3.77), and are depicted in Fig. 3.8 as well as the sources themselves.

In order to see the time-variations of the mixing matrix, each element of the Jacobian matrix of the first simulation (3.75) for $\alpha_0 = 0$ and $\gamma = 0.1$ is plotted separately in Fig. 3.9. It can be seen that their variations along time is periodic (because of the dynamics of the source). As mentioned earlier, variations of the value of the Jacobian are due to both time-variations of the sources and nonlinearity of the mixing function (which make the Jacobian dependent on the value of the sources).

AATVL and BATIN algorithms are applied on the observations of Fig. 3.8 to separate the sources. As mentioned earlier, smoothing spline is the algorithm that is utilized for the nonlinear regression step of algorithm 2. Note that the smoothing parameter, which determines the smoothness of the learned manifold in smoothing spline method, is adjusted heuristically in this work. It should be noted that similarly with the integral, the *difference between two successive time samples* is used as an approximation of the time-derivative everywhere in this work.

In the implementation of Normalized EASI (3.56) in this work, $\mathfrak{h}(\cdot)$ is chosen as $\mathfrak{h}(\mathbf{y}) = \mathbf{y}^3$. In addition, the adaptation step λ_t in (3.56) is chosen

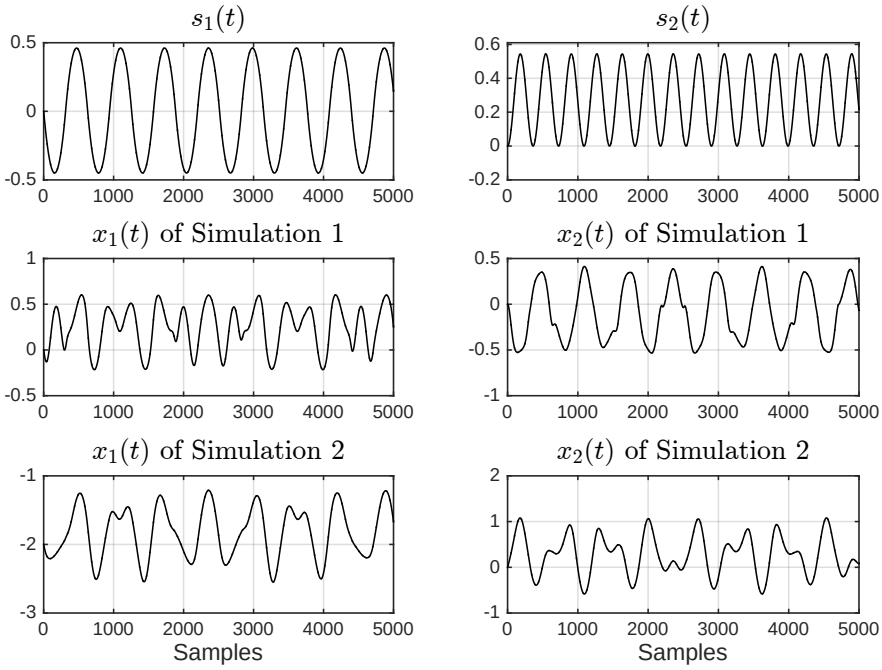


Figure 3.8: The sources $s_1(t)$ and $s_2(t)$ (the integral of a sine and a triangle wave) in the top row, and the observations $x_1(t)$ and $x_2(t)$ for the two simulations with the nonlinear model (2.12) in the middle and with the nonlinear model (3.77) in the bottom.

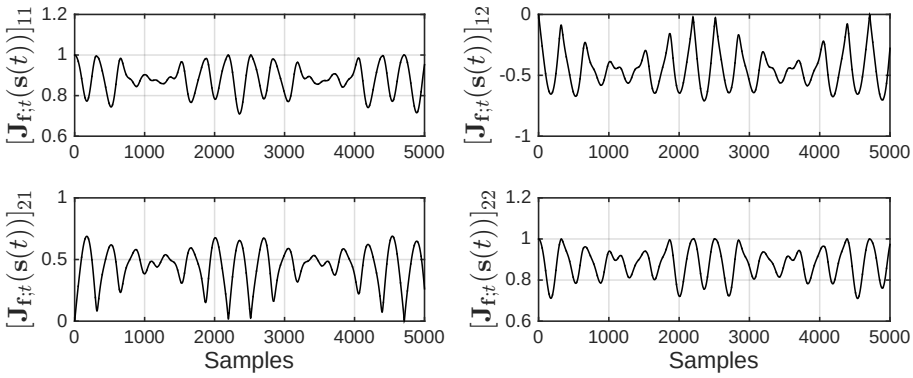
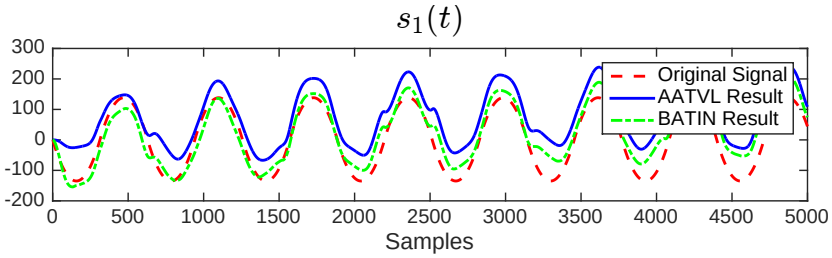
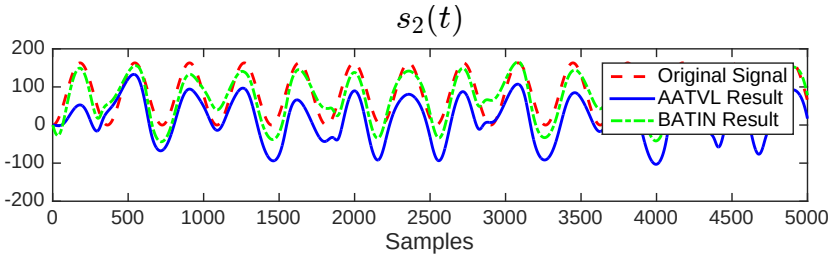


Figure 3.9: Variations of the elements of the Jacobian matrix of (3.75) along the samples



(a) The first source



(b) The second source

Figure 3.10: The results of AATVL and BATIN algorithm in the mixture (3.75)

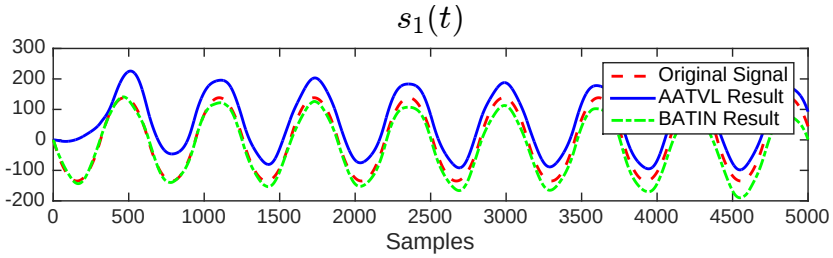
as

$$\lambda_t = \begin{cases} 1/t, & 1 \leq t \leq 1000 \\ 1/1000, & 1000 < t \end{cases}. \tag{3.80}$$

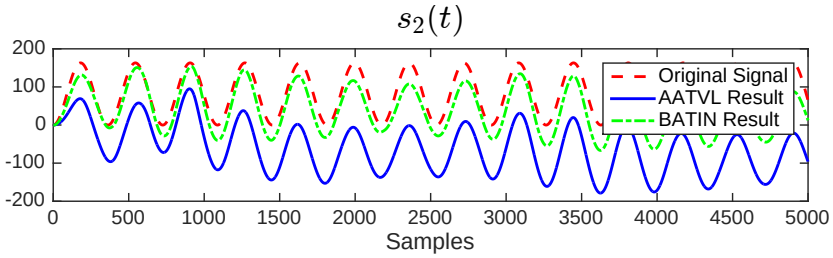
Even though a decreasing adaptation step (tending to zero as t moves forward) is traditionally taken in order to stabilize the algorithm after the convergence [Cardoso and Laheld, 1996], in this case it does not go below a threshold. This is because the mixing matrix $\mathbf{J}_{\mathbf{x};t}$ is not constant and should be followed by the algorithm.

3.4.2 Simulation Results

Applying AATVL and BATIN algorithms on the observations, we get the results shown in Fig. 3.10 for the first simulation (mapping of Eq. (3.75)), and Fig. 3.11 for the second one (mapping of Eq. (3.77)). As expected, BATIN outperforms AATVL in estimating the separated sources in both simulations. Especially, the late convergence problem with AATVL has been almost completely resolved by BATIN.



(a) The first source



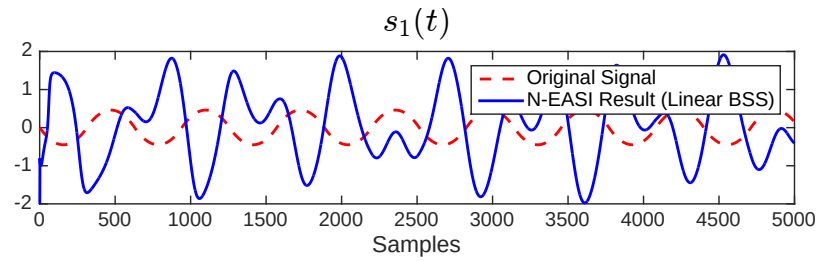
(b) The second source

Figure 3.11: The results of AATVL and BATIN algorithm in the mixture (3.77)

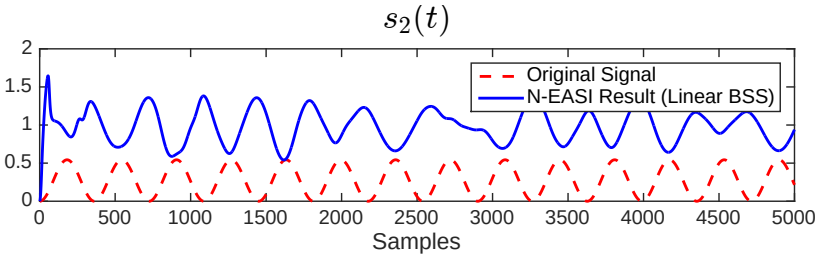
It should be noted that since in nonlinear BSS, sources can be reconstructed up to a nonlinear function which remains as an ambiguity, the performance of the algorithms may not be evaluated by looking at the waveform of the signals. For this reason, we have proposed a novel performance index for nonlinear BSS which will be introduced in Section 3.4.3.

In our simulations, in order to reduce the computational cost, the nonlinear regression is performed based on the result of the "Adaptive Linear BSS" procedure on *down-sampled* signals. However, one can utilize a smarter method (than a uniform down-sampling) for picking some points in order to estimate the nonlinear functions, which may highly affect the performance of the algorithm.

Additionally, in order to see that adaptive linear BSS algorithms are not able to separate the sources (since the mixture is nonlinear), we have also implemented the same algorithm Normalized EASI for separating the mixture (3.77). It can be seen from Fig. 3.12 that the nonlinear mixture is not separated at all since EASI never converges.



(a) The first source



(b) The second source

Figure 3.12: The result of performing adaptive linear BSS (Normalized EASI method) on the sources which are mixed through (3.77)

3.4.3 Performance Evaluation

As mentioned earlier in Section 3.1.3 and Section 3.3, unlike linear BSS where the sources may be estimated up to a scaling (and a permutation), in nonlinear problem, they can be estimated up to a *nonlinear transformation* (and a permutation). Depending on the application, there should be some known characteristics of the sources (e.g. band-limited, sparse in some domain, bounded amplitude, and so forth) allowing the exact reconstruction of the sources. As a consequence, traditional performance index (e.g. normalized RMS error) cannot be applied in nonlinear BSS.

Without loosing generality, assume that the sources are separated as $y_i(t) = c_i(s_i(t))$ for $i = 1, \ldots, n$ where c_i's are nonlinear functions. Therefore, in noiseless problems, the pairs $(s_i(t), y_i(t))$ for $t = 1 \ldots, T$ lie on a 1-dimensional manifold in a 2-dimensional space. However, if y_i depended on another source s_j $(i \neq j)$, it would not be a mathematical function of s_i which would make the scatter plot of $(s_i(t), y_i(t))$ thick instead of a 1-

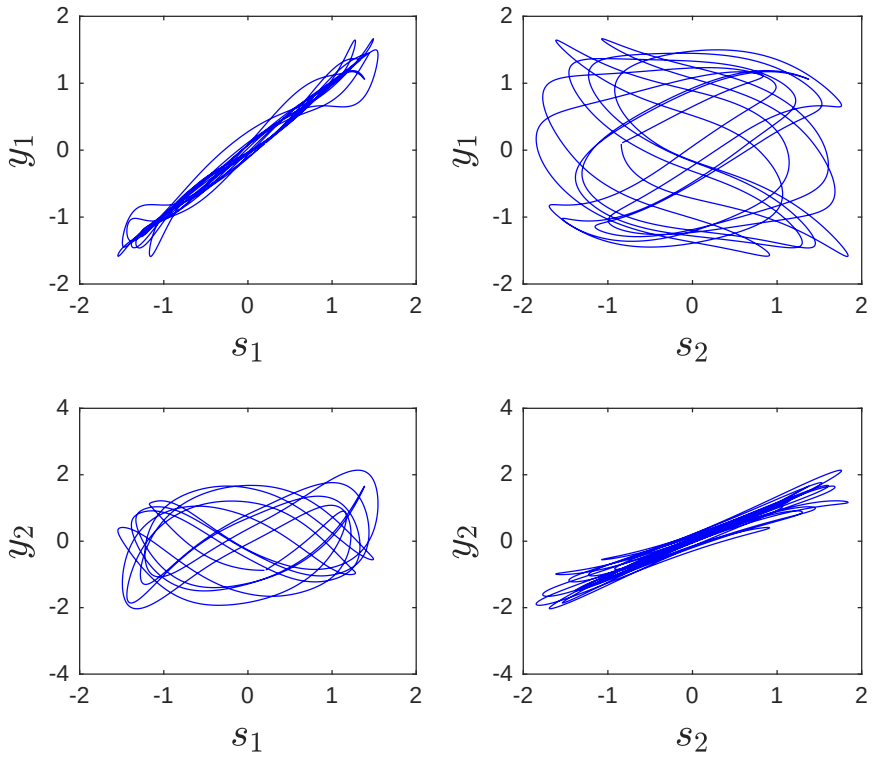


Figure 3.13: The estimated sources $y_1(t)$ and $y_2(t)$ against the actual sources $s_1(t)$ and $s_2(t)$, where the thickness of a plot indicates how much the estimated signal (vertical axis) depends on the other source

dimensional manifold. This fact is also illustrated in Fig. 3.13. Since the pairs $(s_1(t), y_1(t))$ (similarly $(s_2(t), y_2(t))$) approximately lie on a 1-dimensional manifold, one concludes that y_1 (y_2) is only a function of s_1 (s_2).

If the separation is perfect, y_1 (y_2) will be exactly just a function of s_1 (s_2), hence the pairs $(s_1(t), y_1(t))$ (similarly $(s_2(t), y_2(t))$) exactly make a 1-dimensional manifold. The thicker the plot of the pairs $(s_i(t), y_i(t))$ is, the more separation error we have. So the *thickness* of the scatter plot indicates whether there is a dependence to another signal or not.

We thus propose this error as a general index for evaluating the performance of nonlinear BSS methods. It can also be understood by modeling

each output $y_i(t)$ as $y_i(t) = h_i(s_i(t)) + \textit{interference}$. This model highlights that the proposed index approximates the normalized interference to signal ratio of the output.

Although the thickness of data in linear 2-dimensional cases can be easily represented by the second eigenvalue of the auto-correlation matrix, it is not trivial in nonlinear problems. Two estimations for this index in nonlinear frameworks can be proposed:

3.4.3.0.1 Local Approximation Since nonlinear manifolds can be approximated linearly in small neighborhoods, an estimation of the index can be made by summing local linear thickness errors over the whole domain. In other words, the data should be split into small bins, the second eigenvalue of the auto-correlation matrix of the data in each bin should be calculated as the local linear RMS error, and then the summation of the local errors is proposed as the estimation of the global thickness index.

3.4.3.0.2 Curve Fitting Another approach for estimating the evaluating index, which is used in our simulations, is based on the error in fitting a nonlinear curve onto the data points. For this purpose, firstly a nonlinear curve is fitted onto the data and then the RMS error of this fitting (similar to (3.70) but for a 1-dimensional manifold fitting) is introduced as the performance indicator (named as *normalized Error of Nonlinear Fit (N-ENF)*). Normalized ENF of the i^{th} source separation can be formulated as

$$\tilde{E}_{nenf} = \frac{\left(\sum\limits_{t=1,\dots,T} \left(\hat{c}_i(s_i(t)) - y_i(t) \right)^2 \right)^{\frac{1}{2}}}{\left(\sum\limits_{t=1,\dots,T} \left(\hat{c}_i(s_i(t)) \right)^2 \right)^{\frac{1}{2}}}, \tag{3.81}$$

where $\hat{c}_i(s_i(t))$ is the best nonlinear curve which can be fitted onto the pairs $(s_i(t), y_i(t))$. In this work, the curve is fitted using smoothing splines [Reinsch, 1967] as

$$\underset{\hat{c}_i}{\text{minimize}} \sum_{t=1}^{T} \left(y_i(t) - \hat{c}_i(s_i(t)) \right)^2 + \delta \sum_{t=1,\dots,T} \left(\hat{c}_i''(s_i(t)) \right)^2 \tag{3.82}$$

Table 3.1: N-ENF Error for AATVL and BATIN in the simulations

	AATVL	**BATIN**
N-ENF for the **Source 1** in the mixture (3.75) & (3.76)	0.0030	0.0019
N-ENF for the **Source 2** in the mixture (3.75) & (3.76)	0.0084	0.0031
N-ENF for the **Source 1** in the mixture (3.77)	0.0025	0.0023
N-ENF for the **Source 2** in the mixture (3.77)	0.0064	0.0040

where $\hat{c}_i''(s_i(t))$ is the second-order time-derivative of $\hat{c}_i(s_i(t))$ and δ is a smoothing parameter.

Simulation results of the algorithms are also compared in terms of normalized ENF error and can be found in table 3.1.

These results show that the proposed idea is able to separate the sources that are mixed nonlinearly, which proves the proposed concept. However, as mentioned earlier, the performance of the proposed approach depends on the amount of the nonlinearity of the mixing function, i.e. as the mixing model gets distant from a linear mixture, the performance of the algorithm decreases. In order to show how the performance changes according to the nonlinearity level, a 3^{rd} experiment is provided as follows.

Recall example (3.75) with $\alpha(\mathbf{s}(t))$ defined as (3.76), when $\alpha_0 = \pi/6$ and parameter γ varies. In this example, if $\gamma = 0$, the mixture will be linear (a $\pi/6$ rotation). But as γ grows, the mixture will become "more" nonlinear. Thus γ can be considered as a level of nonlinearity of this parametric model.

Finally, the algorithm BATIN is employed for separating two sources of (3.78) and (3.79) mixed by (3.75), for different values of γ in (3.76). The normalized ENF error of BATIN for both sources is calculated and plotted in Fig. 3.14. Evidently, the more the mixture is nonlinear, the less efficient the proposed method is in separating the sources.

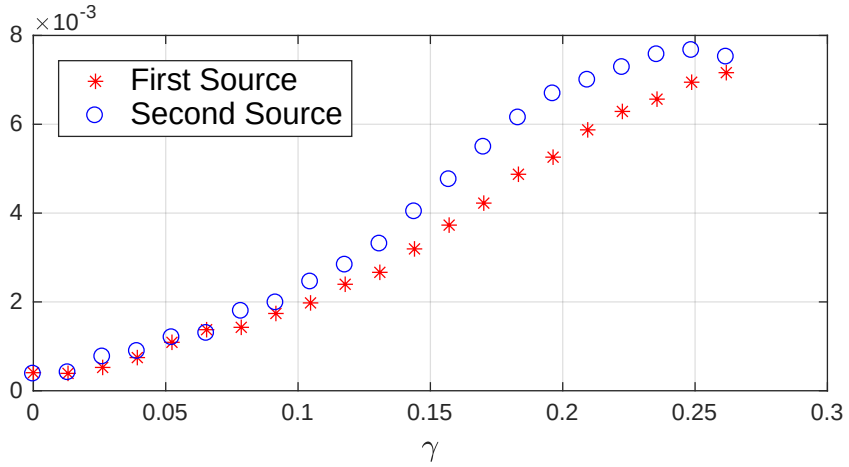


Figure 3.14: The normalized ENF error in separating the mixture (3.75) for different levels of nonlinearity (represented by γ in (3.76)) using BATIN algorithm

3.5 Conclusions and Perspectives

In this chapter, a novel approach for performing nonlinear BSS is proposed. Through this approach, it is shown that nonlinear mixtures are generally separable under a few assumptions (see subsection 3.1.3). So the counter-examples provided in the literature to show that nonlinear mixtures are not separable, are not valid any more.

The key idea is to consider the time-derivative of the observed signals as a time-varying linear mixture of the (mutually independent) time derivatives of the sources. As a consequence, the model (3.12) will be obtained, where the mixing matrix is a function of the sources (not to be confused with a time-variant mixing matrix which is a function of time).

Assuming both sources as functions of the time and nonlinear mapping as a function of the sources to be smooth enough yields a sufficiently smooth mixing matrix which can be considered as a time-variant model (AATVL algorithm). However, the model (3.12) being a function of the *sources* instead of conventional time-variant mixing models, enables performing the nonlinear regression (as explained in Section 3.2.3) and dramatically improves

the performance of the separation, which resulted in proposing the second algorithm (BATIN).

Once the sources are *separated*, BSS has been performed. However, aiming at exactly recovering the sources (not only separating them), the problem reduces to compensating an unknown nonlinear distortion. In other words, in order to precisely estimating the source signals (compensating the nonlinear function), each of the separated signals should be considered separately.

Numerous algorithms have been proposed for blind restoration of nonlinearly distorted signals (e.g. [Marvasti and Jain, 1986, Dogancay, 2005]). The proposed methods are fundamentally based on retrieving some characteristics of the signal which are affected by nonlinear distortions. For example, nonlinear functions generally widen the bandwidth of signals. Thus, given a distorted band-limited signal, one may recover the original signal by trying to minimize its bandwidth via a nonlinear (compensating) function.

Moreover, assuming that the nonlinearly distorted signal is sparse in some domain, it can be blindly reconstructed [Malek, 2013, Duarte et al., 2015]. Since nonlinear distortions generally tend to reduce the sparsity, the proposed algorithms compensate the distortion via a sparse recovery procedure.

Nonetheless, depending on the application, there should be some known characteristics of the sources (e.g. band-limited, sparse in some domain, bounded amplitude, and so forth) allowing the exact reconstruction of the sources. However, being focused on source separation, source *reconstruction* is out of the scope of our work and is suggested as a direction for future studies.

The basic idea proposed in this chapter for nonlinear BSS is to utilize time-derivatives of the signals. Working with time-derivatives implicitly utilizes temporal information in the signals. This fact also supports the proposition in [Hosseini and Jutten, 2003], which says that although we may mix two sources such that the mixtures are instantaneously independent of each other, it is highly probable that their delayed versions are not mutually independent when each of them is temporally correlated. In other words, it was

implied in that paper that utilizing the temporal information of the sources might lead to solve nonlinear BSS problems.

It is worth noting that the proposed idea is quite different with respect to the previous works in the literature on nonlinear mixtures; it is more theoretic and general and does not assume any specific mixing model or source signals. Two basic methods, AATVL and BATIN were provided in this chapter to show how the idea is to be employed. Nevertheless, many different separation algorithms can be suggested based on the proposed approach and they can be optimized to deal with more complex signals/mixtures of practical applications.

However, there are several issues to be considered in the future. Firstly, the statistical characteristics of the derivative of a signal with respect to those of the signal, itself, should be investigated. This might be the key to better understanding of the key feature of derivatives that lets perform the separation, and accordingly, it may lead to new algorithms of nonlinear BSS.

Secondly, the "Nonlinear Regression" used in the proposed algorithm should be improved. The main objective of this step is to accumulate the information of the separation at each sample. For example, if at two different times, the source vector takes the same value, the mixing matrix will remain the same as well.

Moreover, assuming a parametric model like Section 3.1.4 would be very interesting in cases where such information exists. So simulating the mathematical derivations of that section, particularly Eq. (3.44), can be a short-term perspective which will validate the theoretical results.

In this thesis, the problem is considered in the simplest form where there is no noise added to the signals. Since all the signals in practical applications are noisy, and considering the fact that taking the derivatives may dramatically amplifies the noise, new methods should be developed which are more robust to noise. It may also enforce some modifications on "Adaptive Linear BSS" procedure of the algorithms as well.

Last but not least, finding out the relations between autocorrelation func-

tions of the sources (i.e. how much colored they are) and the performance of the proposed approach and trying to quantify it is also an interest for future studies.

4 Blind Linearization of Nonlinear Mixtures

Contents

As hypothesized in previous chapters, given sources having temporal correlation, nonlinear mixtures may be blindly separated by retrieving the independence. This information was utilized in the proposed approach of Chapter 3 by assuming that the variations of the Jacobian of the mixing function is smooth enough, and by locally linear approximating the mixture. In this chapter, instead, sources are assumed to be modeled by Gaussian processes. Our proposed approach for this purpose is to linearize the nonlinear mixture

such that it can then be separated via a linear BSS technique. This is the reason why the chapter is named "Blind Linearization of Nonlinear Mixtures" which has applications not only in nonlinear BSS, but also in other signal processing domains which are briefly introduced in the following.

A notable practical application for blind linearization of nonlinear mixtures is for electroencephalogram (EEG) and electromagnetogram (EMG) signals, for which the sources can be very well approximated by a Gaussian distribution (according to the central limit theorem and numerous neural activities), but may have passed a nonlinear transformation before being recorded on the body surface.

In this chapter, a mathematical proof is provided to show that Gaussian signals will lose their Gaussianity if they are passed through a polynomial of an order greater than 1. This can help in blind compensation of polynomial nonlinearities on Gaussian sources by forcing the output to follow a Gaussian distribution, as done for post-nonlinear mixtures [Larue et al., 2004].

The idea of this chapter is original and has been partially published in [Ehsandoust et al., 2017b] and [Fantinato et al., 2017]. The chapter is organized as follows. Firstly, the assumed model is presented and some of its applications are introduced. Then in Section 4.2 the linearization of nonlinear mixtures of Gaussian sources is mathematically investigated and theoretical results are derived. In Section 4.3 a simple algorithm is proposed based on the results of Section 4.2, which is then supported by simulations in Section 4.4. Finally, the results are discussed in Section 4.5, where directions for future works are also suggested.

4.1 Introduction

As mentioned before, in signal processing applications, including BSS, it is usual to have a number of signals measured by some sensors, while each of them might be a mixture of a number of source signals. Even though this problem is relatively easy to solve when the mixture is linear (more generally

linear time-invariant), it becomes mostly too difficult for general nonlinear functions. Thus it is often wanted to transform the nonlinear system to a linear one in order that it can be processed by already established signal processing methods for linear mixtures. So the problem would be decomposed into two consecutive steps: 1) the problem is linearized and 2) the transformed linear problem is processed using traditional linear approaches. In this framework, the current chapter is only focused on the first step and can be used in different applications.

Linearizing a nonlinear problem generally needs some prior knowledge about the input signals and/or the nonlinear mapping. Mainly, the source signals are assumed to have some known characteristics. Trying to retrieve these specifications in the output may result in linearizing the problem in some cases. For example, sparsity in [Duarte et al., 2015], bandwidth in [Dogancay, 2005], zero-crossing in [Marvasti and Jain, 1986], etc. are shown to be useful for not only linearizing the mixture, but also reconstructing the sources without knowing the nonlinear distortion.

In this chapter, in order to employ the spectral colorfulness of the sources, they are assumed to follow Gaussian (normal) distribution. Thus the question in this case is whether retrieving normality in the output results in compensating the unknown nonlinearity or not.

On the other hand, in many applications of signal processing, signals are modeled as stochastic processes. In this sense, Gaussian random variables and Gaussian Processes (GPs) are interesting models because of their simplicity, generality and nice characteristics.

A GP [Rasmussen and Williams, 2006] is a collection of random variables, any finite subset of which has a multivariate Gaussian distribution as $\mathbf{t} \sim \mathcal{N}(\mathbf{m}, \mathbf{K})$, where $\mathcal{N}(\mathbf{m}, \mathbf{K})$ is a Gaussian distribution with the mean vector $\mathbf{m} = [m_1, \ldots, m_n]^\dagger$ and the $n \times n$ covariance matrix $\mathbf{K}$. However, the probability distribution can be specified not only over random variables, but also over functions with an infinite-size domain.

A GP is completely determined by only its mean and covariance func-

tions. This property facilitates model fitting as only the first- and second-order moments of the process require specification. Thus, a random function $s(t)$, as a statistical process, can be fully described at the second order by its mean function $m(t)$, and its covariance function $k(t,t')$ defined as

$$m(t) = \mathbb{E}[s(t)], \tag{4.1}$$

$$k(t,t') = \mathbb{E}[\big(s(t) - m(t)\big)\big(s(t') - m(t')\big)]. \tag{4.2}$$

The set of real valued functions $s(t) \in \mathbb{R}$, can then be described as a Gaussian process as

$$s(t) \sim \mathcal{GP}\big(m(t;\boldsymbol{\theta}), k(t,t';\boldsymbol{\theta})\big). \tag{4.3}$$

By choosing particular mean and covariance functions for the GP, we can introduce some hyperparameters, notated as the set $\boldsymbol{\theta}$, to the prior of the GP. These hyperparameters control the behavior of the functions over which the GP is defined. Now, considering (4.3), it can be said that a collection of random variables $s(t)_{t\in\boldsymbol{t}}$ is drawn from a GP with mean function $m(t;\boldsymbol{\theta})$, and covariance function $k(t,t';\boldsymbol{\theta})$, if the associated finite set of $\{s(t_1),\cdots,s(t_n)\}$ indexed by the inputs $\{t_1,\cdots,t_n\} \in \boldsymbol{t}$ has a distribution as

$$\begin{bmatrix} s(t_1) \\ \vdots \\ s(t_n) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(t_1) \\ \vdots \\ m(t_n) \end{bmatrix} , \begin{bmatrix} k(t_1,t_1) & \cdots & k(t_1,t_n) \\ \vdots & \ddots & \vdots \\ k(t_n,t_1) & \cdots & k(t_n,t_n) \end{bmatrix} \right). \tag{4.4}$$

GPs can be used to track nonlinear communication channels or for probabilistic channel equalization [Pérez-Cruz et al., 2013], for classification [Rasmussen and Williams, 2006] or to perform linear source separation [Rivet et al., 2012, Noorzadeh et al., 2015a].

In this chapter, the goal is to blindly transform a nonlinear system to a linear one under the assumption that the sources are normally distributed. As a result of this work, problems in any domain of signals processing which satisfy the mentioned assumptions, can be pre-processed in order to be transformed to linear ones and then treated linearly.

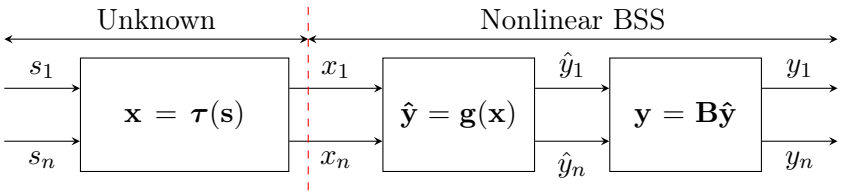


Figure 4.1: Nonlinear BSS can be decomposed to a blind linearization step cascaded by a conventional linear BSS method

It will be shown that if the unknown mixing function is an invertible *polynomial* mapping of the sources, the Gaussianity property is sufficient for inverting the nonlinear mapping and reducing the problem to a source separation problem with linear mixtures.

4.1.1 Application to Nonlinear BSS

Accordingly, in nonlinear BSS problems, blind linearization can be utilized as a pre-processing step before linear BSS (see Fig. 4.1). In this approach, a linearizing function $\mathbf{g}$, cascaded by a linear BSS technique, would be capable of separating nonlinear mixtures.

It is worth noting that there are numerous practical applications of BSS where the sources are modeled by GPs, e.g. [Liutkus et al., 2011, Rivet et al., 2012, Noorzadeh et al., 2015a, Noorzadeh et al., 2015b]. In these applications, utilizing the proposed method leads to a set of signals that are linear mixtures of mutually independent sources.

Therefore, in order to reconstruct the sources after linearizing the mixture, one can use either conventional linear BSS techniques that use the temporal correlation of the sources (like SOBI [Belouchrani et al., 1997]) or recent specific methods for GPs (like [Noorzadeh et al., 2014]).

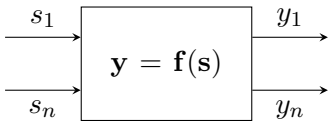


Figure 4.2: Unknown mapping $\mathbf{f}$ preserving normality

4.2 Theory

In this section, we aim at studying functions preserving normality to see whether they are limited to be linear or not. It will be seen that even in one dimension, this hypothesis is not valid, i.e. there are normality-preserving nonlinear mapping.

As mentioned earlier, the current chapter only focuses on the linearization of nonlinear mixtures ($\mathbf{g}$ in Fig. 4.1). So, for the sake of simplicity of notations, let us slightly change the notation of Fig. 4.1, after removing the last block, as follows.

Let n sources $s_1, \cdots, s_n$ be jointly normally distributed and mixed via an invertible nonlinear mapping $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ providing n outputs $y_1, \cdots, y_n$ (Fig. 4.2). The main question is the following: if the outputs $y_1, \cdots, y_n$ also follow a joint normal distribution, is the mapping $\mathbf{f}$ limited to have any specific structure? In other words, which class of functions $\mathbf{f}$ can result in normal outputs?

4.2.1 One-Dimensional Functions

It is tempting to think that in one-dimensional space, if a Gaussian input is transformed into a Gaussian output, then the transformation has to be linear. However, it is not difficult to find counter-examples. For example, one can consider

$$f(x) = \begin{cases} -x & a \leq |x| < b \\ x & \text{otherwise} \end{cases} \tag{4.5}$$

where $0 \leq a < b$ are positive real numbers. As long as the input comes from a symmetric distribution (including Gaussian), function (4.5) preserves it in its output.

Since function (4.5) has discontinuities, it is natural to conjecture that under continuity assumption the only possible transforms are linear. However, that is not the case again. This is shown in [Wesołowski, 1997] by introducing a one-dimensional continuous mapping which preserves normality as follows (refer to [Wesołowski, 1997] for the proof). Define

$$b(x) \triangleq \Phi(\frac{x-3}{4}) + \Phi(\frac{x+3}{4}) - \Phi(x), \tag{4.6}$$

where Φ is the cumulative distribution function (cdf) of the standard normal distribution $\mathcal{N}(0,1)$, defined as

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} \, \mathrm{d}\, t. \tag{4.7}$$

Then the normality-preserving function is constructed as

$$f(x) = \begin{cases} x, & |x| \geq 1, \\ 4x+3, & -1 < x < -0.5, \\ B(x), & -0.5 \leq x \leq 0.5, \\ 4x-3, & 0.5 < x < 1, \end{cases} \tag{4.8}$$

where $B = b^{-1} \circ \Phi$.

4.2.2 High Dimensional Functions

Considering Section 4.2.1, it is evident that there are N-dimensional continuous functions preserving normality. For example, if f is a continuous one-dimensional function which preserves normality, let us define the function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ as $\mathbf{f}(\mathbf{x}) = (f(x_1), ..., f(x_n))^\dagger$ where $\mathbf{x} = (x_1, \ldots, x_n)^\dagger \in \mathbb{R}^n$, which is a continuous nonlinear function as well. It can be shown that if $\mathbf{x}$ is an n-variate random vector with independent and identically distributed (iid) standard normal components then, obviously, $\mathbf{f}(\mathbf{x}) \stackrel{d}{=} \mathbf{x}$, where $\stackrel{d}{=}$ stands for equality of probability density functions.

Nonetheless, it will be interesting to study functions preserving normality in more details aiming at being able of restricting such functions to some specific classes which may be useful in practical applications.

4.2.3 Polynomial Functions

This section aims at mathematically understanding what happens to the GPs passing through a polynomial. As follows, it is proved that polynomials distort the Gaussianity characteristic of their inputs, except linear ones. It should be noted that this result cannot be directly extended for any kind of nonlinear functions (it will be elaborated in the following).

Definition An n-dimensional mapping $\mathbf{p}: \mathbb{R}^n \to \mathbb{R}^n$, defined as $\mathbf{p}(\mathbf{s}) = (p_1(\mathbf{s}), \ldots, p_n(\mathbf{s}))^\dagger$ (where $\mathbf{s}$ is an $n \times 1$ vector of variables) is called an n-dimensional *polynomial mapping* if each p_i is a polynomial (of order O_i of n variables $s_1, \ldots, s_n$). □

Thus we propose a theorem concerning polynomial mappings as follows.

Theorem 3. *Let n sources $s_1, \cdots, s_n$ be jointly normally distributed and mixed via an invertible polynomial mapping $\mathbf{p}: \mathbb{R}^n \to \mathbb{R}^n$ providing n outputs $y_1, \cdots, y_n$. If the outputs $y_1, \ldots, y_n$ also follow a joint Gaussian distribution, the polynomial $\mathbf{p}$ is limited to be linear as*

$$\mathbf{y} = \mathbf{p}(\mathbf{s}) = \mathbf{A}\mathbf{s} + \mathbf{b} \tag{4.9}$$

where $\mathbf{A}$ and $\mathbf{b}$ are an $n \times n$ matrix and an $n \times 1$ vector of constant numbers respectively.

In other words, the Theorem 3 says that the only polynomial which preserves the Gaussianity is the linear one. It is worth noting that the reverse is a well known result: a linear mixture of Gaussian processes (random variables) leads to Gaussian processes (random variables).

Proof. Let us assume $(s_1, \ldots, s_n)$ has a mean vector $\boldsymbol{\mu}_\mathbf{s} = \mathbb{E}[\mathbf{s}]$ and a covariance matrix $\mathbf{K}_\mathbf{s} = \mathbb{E}[(\mathbf{s} - \boldsymbol{\mu}_\mathbf{s})(\mathbf{s} - \boldsymbol{\mu}_\mathbf{s})^\dagger]$. Similarly, the output vector $(y_1, \ldots, y_n)$ has a mean vector $\boldsymbol{\mu}_\mathbf{y} = \mathbb{E}[\mathbf{y}]$ and a covariance matrix

$\mathbf{K_y} = \mathbb{E}[(\mathbf{y} - \boldsymbol{\mu_y})(\mathbf{y} - \boldsymbol{\mu_y})^\dagger]$. Considering the Gaussianity, the probability density function (pdf) of the vectors $\mathbf{s}$ and $\mathbf{y}$, denoted by $\rho_S(\mathbf{s})$ and $\rho_Y(\mathbf{y})$ respectively, can be expressed as

$$\rho_S(\mathbf{s}) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{K_s}|}}\, e^{\frac{-1}{2}(\mathbf{s}-\boldsymbol{\mu_s})^\dagger \mathbf{K_s}^{-1}(\mathbf{s}-\boldsymbol{\mu_s})} \tag{4.10}$$

$$\rho_Y(\mathbf{y}) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{K_y}|}}\, e^{\frac{-1}{2}(\mathbf{y}-\boldsymbol{\mu_y})^\dagger \mathbf{K_y}^{-1}(\mathbf{y}-\boldsymbol{\mu_y})} \tag{4.11}$$

where $|\mathbf{K_s}|$ and $|\mathbf{K_y}|$ are the determinants of $\mathbf{K_s}$ and $\mathbf{K_y}$ respectively. On the other hand, according to (2.9), the pdf of $\mathbf{y}$ follows

$$\rho_Y(\mathbf{y}) = \frac{\rho_S(\mathbf{s})}{|\mathbf{J_p}|} \tag{4.12}$$

where $|\mathbf{J_p}|$ is the determinant of the Jacobian of $\mathbf{p}$.

By definition, it is easy to see that all elements of the Jacobian of a polynomial mapping are polynomials. As a consequence, the determinant of the Jacobian is the absolute value of a polynomial. Let $q(\mathbf{s})$ be a polynomial such that

$$\det(\mathbf{J_p}) = |q(\mathbf{s})|. \tag{4.13}$$

Thus (4.12) can be rewritten as

$$|q(\mathbf{s})| \times \rho_Y(\mathbf{y}) = \rho_S(\mathbf{s}) \tag{4.14}$$

$$\Rightarrow \ln|q(\mathbf{s})| + \ln \rho_Y(\mathbf{y}) = \ln \rho_S(\mathbf{s}). \tag{4.15}$$

Using (4.10) and (4.11) to explicit (4.15) leads to

$$\ln|q(\mathbf{s})| - \frac{1}{2}\ln|\mathbf{K_y}| - \frac{1}{2}(\mathbf{y}-\boldsymbol{\mu_y})^\dagger \mathbf{K_y}^{-1}(\mathbf{y}-\boldsymbol{\mu_y}) = - \frac{1}{2}\ln|\mathbf{K_s}| - \frac{1}{2}(\mathbf{s}-\boldsymbol{\mu_s})^\dagger \mathbf{K_s}^{-1}(\mathbf{s}-\boldsymbol{\mu_s}), \tag{4.16}$$

that is

$$c + (\mathbf{y}-\boldsymbol{\mu_y})^\dagger \mathbf{K_y}^{-1}(\mathbf{y}-\boldsymbol{\mu_y}) = (\mathbf{s}-\boldsymbol{\mu_s})^\dagger \mathbf{K_s}^{-1}(\mathbf{s}-\boldsymbol{\mu_s}) + 2\ln|q(\mathbf{s})| \tag{4.17}$$

where $c = \ln(|\mathbf{K_y}|/|\mathbf{K_s}|) = \ln|\mathbf{K_y}\mathbf{K_s}^{-1}|$ is a constant independent of $\mathbf{s}$ and $\mathbf{y}$. Now it should be proved that since (4.17) holds for all $\mathbf{s} \in \mathbb{R}^n$, $\mathbf{y}$ must be a linear function of $\mathbf{s}$.

In particular, (4.17) holds for any vector $\mathbf{s}$ that lies on the line where all entries of the vector take the same value, i.e. $\mathbf{s} = (s, \ldots, s)^\dagger_{n\times 1} = s\mathbf{1}_{n\times 1}$. In this case, $\mathbf{y} = \mathbf{p}(\mathbf{s}) = \tilde{\mathbf{p}}(s)$ and $q(\mathbf{s})$ become single variable polynomials of s as follows

$$\mathbf{s} = s\mathbf{1}_{n\times 1} \Rightarrow \mathbf{y} = \tilde{\mathbf{p}}(s) = [\ \tilde{p}_1(s), \ldots, \tilde{p}_n(s)\ ]^\dagger \tag{4.18}$$

and $q(\mathbf{s}) = \tilde{q}(s)$ where

$$\forall 1 \leq k \leq n \quad \tilde{p}_k(s) = p_k(\mathbf{s})\big|_{\mathbf{s}=s\mathbf{1}_{n\times 1}} = \sum_{j=0}^{d_k} a_{kj} s^j \tag{4.19}$$

$$\tilde{q}(s) = q(\mathbf{s})\big|_{\mathbf{s}=s\mathbf{1}_{n\times 1}} = \sum_{i=0}^{d_\mathbf{J}} b_i s^i. \tag{4.20}$$

Replacing (4.19) and (4.20) in (4.17) results in

$$\begin{aligned} c + (\tilde{\mathbf{p}}(s) - \boldsymbol{\mu}_\mathbf{y})^\dagger \mathbf{K}_\mathbf{y}^{-1} (\tilde{\mathbf{p}}(s) - \boldsymbol{\mu}_\mathbf{y}) & \\ = (s\mathbf{1}_{n\times 1} - \boldsymbol{\mu}_\mathbf{s})^\dagger \mathbf{K}_\mathbf{s}^{-1} (s\mathbf{1}_{n\times 1} - \boldsymbol{\mu}_\mathbf{s}) + 2\ln|\tilde{q}(s)| & \\ = \alpha s^2 + \beta s + \gamma + 2\ln|\tilde{q}(s)| & \end{aligned} \tag{4.21}$$

where $\alpha = \mathbf{1}^\dagger_{n\times 1}\mathbf{K}_\mathbf{s}^{-1}\mathbf{1}_{n\times 1}$, $\beta = -2\mathbf{1}^\dagger_{n\times 1}\mathbf{K}_\mathbf{s}^{-1}\boldsymbol{\mu}_\mathbf{s}$ and $\gamma = \boldsymbol{\mu}_\mathbf{s}^\dagger\mathbf{K}_\mathbf{s}^{-1}\boldsymbol{\mu}_\mathbf{s}$ are constant scalars.

Particularly, it is interesting to study the equality (4.21) when s tends to infinity. From (4.19) it can be seen that for large s, the right side behaves as αs^2 (considering the fact that the asymptotic growth of s^2 is faster than both s and logarithms), so the left side should also behave as a second order polynomial. In other words, all monomials in $\tilde{\mathbf{p}}(s)$, hence $\mathbf{p}(\mathbf{s})$, have a degree at most 1 and $\mathbf{p}(\mathbf{s})$ is limited to be linear as (4.9). □

Thus, being restricted to invertible polynomials, normality-preserving functions are necessarily linear. The statistics of $\mathbf{y}$ can be easily expressed with respect to the ones of $\mathbf{s}$ as

$$\boldsymbol{\mu}_\mathbf{y} = \mathbb{E}[\mathbf{y}] = \mathbb{E}[\mathbf{A}\mathbf{s} + \mathbf{b}] = \mathbf{A}\boldsymbol{\mu}_\mathbf{s} + \mathbf{b} \tag{4.22}$$

and

$$\mathbf{K_y} = \mathbb{E}[(\mathbf{y}-\boldsymbol{\mu_y})(\mathbf{y}-\boldsymbol{\mu_y})^\dagger] \\ = \mathbb{E}[(\mathbf{A}(\mathbf{s}-\boldsymbol{\mu_s}))(\mathbf{A}(\mathbf{s}-\boldsymbol{\mu_s}))^\dagger] = \mathbf{A}\mathbf{K_s}\mathbf{A}^\dagger. \quad (4.23)$$

Corollary 1. *In the model of Fig. 4.2, assuming that* $\mathbf{f}: \mathbb{R}^n \to \mathbb{R}^n$ *is an invertible polynomial, and given Gaussian Processes as the sources, if we find a polynomial mapping* $\mathbf{g}(\mathbf{x})$ *such that the outputs* $y_1(t), y_2(t), \ldots, y_n(t)$ *are Gaussian Processes, the whole function* $\mathbf{h} = \mathbf{g} \circ \mathbf{f}$ *will be a linear mixture, i.e.* $\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) = \mathbf{h}(\mathbf{s}(t)) = \mathbf{A}\mathbf{s}(t)$. *It should be noted that the constant vector* $\mathbf{b}$ *is dropped because it would affect the mean of the signals, while in the proposed framework, they are assumed to be zero-mean.*

It should be noted that although Theorem 3 holds for a more general class of signals than GPs, they are very useful and flexible in modeling many practical signals (as introduced in Section 4.1).

Proof. By definition, $s_i(t)$ is a Gaussian Process if for any set of M_i time instants $\xi_i = \{t_{i1}, \ldots, t_{iM_i}\}$, the vector $\mathbf{s_{i}}_{\xi_i} = (s_i(t_{i1}), \ldots, s_i(t_{iM_i}))^\dagger$ follows a Gaussian pdf with a mean $\boldsymbol{\mu}_{\mathbf{s_i}}(\xi_i)$ and and a covariance matrix $\mathbf{K}_{\mathbf{s_i}}(\xi_i)$. Consequently, the vector $\mathbf{s}_\xi = (\mathbf{s_1}_{\xi_1}^\dagger, \ldots, \mathbf{s_n}_{\xi_n}^\dagger)^\dagger$ is normally distributed.

On the other hand, since both $\mathbf{f}$ and $\mathbf{g}$ are polynomial mappings, their composition $\mathbf{h} = \mathbf{g} \circ \mathbf{f}$ will be a polynomial mapping as well. According to Theorem 3, since the output vector $\mathbf{y}_\xi = (\mathbf{y_1}_{\xi_1}^\dagger, \ldots, \mathbf{y_n}_{\xi_n}^\dagger)^\dagger = \mathbf{h}(\mathbf{s}_\xi)$ is also a normally distributed vector, $\mathbf{h}$ is limited to be linear. □

4.2.4 Algebraic Functions

According to Taylor expansion theorem, smooth-enough nonlinear functions can be approximated by polynomials. However, Theorem 3 is not shown to hold for polynomials of infinite order. Thus, studying other sets of nonlinear functions would be of interest.

Algebraic functions can be seen as a generalization of polynomials. Thus one may initially hypothesize that they may not preserve Gaussianity either.

In this subsection, we investigate whether the Gaussianity can survive passing through either algebraic or transcendental functions. The short answer is "no"; Theorem 3 cannot even be generalized to algebraic functions in general.

Let us firstly define algebraic and transcendental functions.

Definition In mathematics, an algebraic function is a function that can be defined as the root of a polynomial equation. Quite often algebraic functions can be expressed using a finite number of terms, involving only the algebraic operations addition, subtraction, multiplication, division, and raising to a fractional power. In more precise terms, an algebraic function of degree d in one variable x is a function $y = \boldsymbol{\mathcal{A}}(x)$ that satisfies a polynomial equation

$$a_d(x)y^d + a_{d-1}(x)y^{d-1} + \cdots + a_0(x) = 0 \tag{4.24}$$

where the coefficients $a_i(x)$ are polynomial functions of x. A function which is not algebraic is called a transcendental function, as it is for example the case of $\exp(x)$, $\tan(x)$, $\ln(x)$ and $\Gamma(x)$. □

Remark 1. *To gain an intuitive understanding, algebraic functions mainly comprise polynomials, rational functions and roots of natural orders.*

Remark 2. *A composition of transcendental functions can give an algebraic function, e.g.* $\boldsymbol{\mathcal{A}}(x) = \cos(\arcsin(x)) = \sqrt{1-x^2}$.

As declared before, Theorem 3, which concerned *polynomial* mappings, cannot even be generalized to algebraic functions. It is shown through theorems in the literature as follows.

1. [Baringhaus et al., 1988, Quine, 1994]: if X_1 and X_2 are independent normal random variables (rv's) with zero means and variances σ_1^2 and σ_2^2, then $Y = X_1X_2/\sqrt{X_1^2 + X_2^2}$ is normal with zero mean and variance σ_3^2, where $1/\sigma_3 = 1/\sigma_1^2 + 1/\sigma_2^2$.

2. [Reid, 1987]: let $\mathbf{X} = [X_1, X_2]^\dagger \sim \mathcal{N}_2(\mathbf{0}, \boldsymbol{\Sigma})$ where

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} \tag{4.25}$$

and define $\mathbf{Y}$ as

$$\mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} (\sigma_1^{-1} + \sigma_2^{-1}) X_1 X_2 / \|\mathbf{X}\| \\ \text{sign}(X_1)(\sigma_1^{-1} X_1^2 - \sigma_2^{-1} X_2^2)/\|\mathbf{X}\| \end{bmatrix}. \tag{4.26}$$

Then $\mathbf{Y} \sim \mathcal{N}_2(\mathbf{0}, \mathbf{I})$.

4.2.5 Generalized Rotations

It is evident that there are also other nonlinearities which preserve normality. For example, inspired by [Babaie-Zadeh, 2002], we propose the following theorem.

Theorem 4. *In Fig. 4.2, suppose* $\mathbf{s}$ *is an* $n \times 1$ *standard normal vector and* $\mathbf{f}$ *is a differentiable one-to-one mapping with a continuous derivative. If* $\mathbf{f}$ *satisfies the following two properties, it will preserve normality:*

1. $\mathbf{f}$ *should be norm-preserving; hyper-spheres in the* $\mathbf{s}$*-space will be mapped into hyper-circles in the* $\mathbf{y}$*-space (*$\mathbf{y} = \mathbf{f}(\mathbf{s})$*).*
2. $|\mathbf{J}_\mathbf{f}(\mathbf{s})|$ *should be constant equal to* 1*.*

It is worth noting that $|\mathbf{J}_\mathbf{f}(\mathbf{s})| = 1$ geometrically means that the transformation $\mathbf{f}$ does not change the *volume* of differential elements (it is proved that the determinant of the Jacobian of a function is the proportion of the change in the differential volumes). Therefore, Theorem 4 claims that functions $\mathbf{f}$ preserving both the differential volume and the norm, preserve Gaussianity.

Proof. Let us firstly recall (2.9), i.e. the relationship between the pdf of the input and the output of a differentiable function $\mathbf{f}$, as

$$\mathbf{y} = \mathbf{f}(\mathbf{s}) \quad \Rightarrow \quad \rho_\mathbf{Y}(\mathbf{y}) = \frac{\rho_\mathbf{S}(\mathbf{s})}{|\det(\mathbf{J}_\mathbf{f}(\mathbf{s}))|}. \tag{4.27}$$

Therefore, considering the standard normal pdf, for normal input and output vectors, we will have

$$\frac{1}{\sqrt{2\pi^n}} e^{\frac{-1}{2}(r^\mathbf{y})^2} = \frac{1}{|\det(\mathbf{J}_\mathbf{f}(\mathbf{s}))|} \frac{1}{\sqrt{2\pi^n}} e^{\frac{-1}{2}(r^\mathbf{s})^2} \tag{4.28}$$

$$\Rightarrow (r^\mathbf{y})^2 = (r^\mathbf{s})^2 + 2\ln|\mathbf{J}_\mathbf{f}|, \tag{4.29}$$

where $r^{\mathbf{s}} = \sqrt{\mathbf{s}^\dagger \mathbf{s}}$ and $r^{\mathbf{y}} = \sqrt{\mathbf{y}^\dagger \mathbf{y}}$. Obviously, (4.29) holds for functions satisfying $|\mathbf{J_f}| = 1$ and $r^{\mathbf{y}} = r^{\mathbf{s}}$. □

In the following, generalized rotations are defined and shown to be nonlinear mappings preserving normality. They can also be continuous and differentiable everywhere. Let us firstly define n-dimensional spherical coordinate systems, based on [Vilenkin, 1978, p. 435], as follows.

Definition An n-dimensional spherical coordinate system (analogous to the one defined in 3-dimensional space) consists of a radial coordinate, r and $n-1$ angular coordinates $\theta_1, \theta_2, \ldots, \theta_{n-1}$ where θ_{n-1} ranges over $[0, \pi)$ and the other angles range over $[0, 2\pi)$ radians. If $x_1, x_2, \ldots, x_n$ are the Cartesian coordinates as $\mathbf{x} = (x_1, x_2, \ldots, x_n)^\dagger$, the coordinates transformation can be expressed as

$$\begin{aligned}
x_1 &= r\cos(\theta_1) \\
x_2 &= r\sin(\theta_1)\cos(\theta_2) \\
&\vdots \\
x_{n-1} &= r\sin(\theta_1)\ldots\sin(\theta_{n-2})\cos(\theta_{n-1}) \\
x_n &= r\sin(\theta_1)\ldots\sin(\theta_{n-2})\sin(\theta_{n-1}).
\end{aligned} \tag{4.30}$$

Reciprocally (refer to [Vilenkin, 1978, p. 436])

$$\begin{aligned}
r &= \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} \\
\theta_1 &= \operatorname{arccot} \frac{x_1}{\sqrt{x_2^2 + \cdots + x_n^2}} \\
\theta_2 &= \operatorname{arccot} \frac{x_2}{\sqrt{x_3^2 + \cdots + x_n^2}} \\
&\vdots \\
\theta_{n-2} &= \operatorname{arccot} \frac{x_{n-2}}{\sqrt{x_{n-1}^2 + x_n^2}} \\
\theta_{n-1} &= 2\operatorname{arccot} \frac{x_{n-1} + \sqrt{x_{n-1}^2 + x_n^2}}{x_n}.
\end{aligned} \tag{4.31}$$

□

Therefore, we define generalized rotations as follows.

Definition An n-dimensional mapping $\mathbf{\Phi} : \mathbb{R}^n \to \mathbb{R}^n$ is called an n-dimensional *generalized rotation* iff it preserves the norm and the angle of rotation may depend on the norm. It can be formulated in n-dimensional spherical coordinate system as

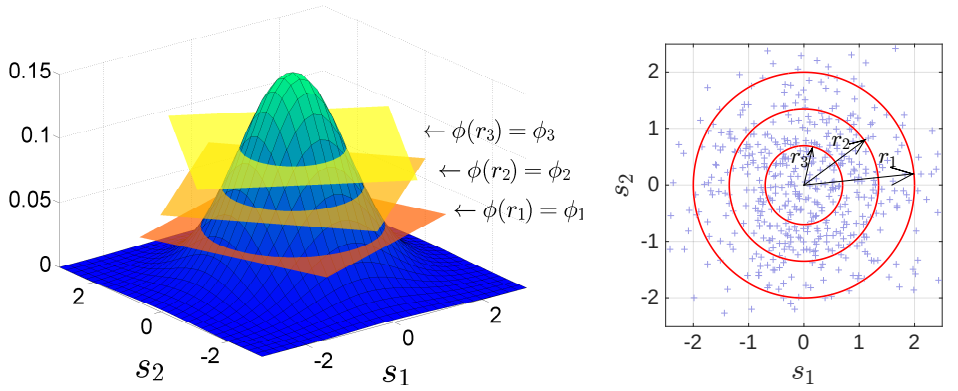
$$\mathbf{y} = \begin{bmatrix} r^{\mathbf{y}} \\ \theta_1^{\mathbf{y}} \\ \theta_2^{\mathbf{y}} \\ \vdots \\ \theta_{n-1}^{\mathbf{y}} \end{bmatrix} = \mathbf{\Phi}(\mathbf{x}) = \begin{bmatrix} r^{\mathbf{x}} \\ \theta_1^{\mathbf{x}} + \phi_1(r^{\mathbf{x}}) \\ \theta_2^{\mathbf{x}} + \phi_2(r^{\mathbf{x}}) \\ \vdots \\ \theta_{n-1}^{\mathbf{x}} + \phi_{n-1}(r^{\mathbf{x}}) \end{bmatrix}. \tag{4.32}$$

where $\phi_i(r^{\mathbf{x}})$ for $i = 1, \ldots, n-1$ is an arbitrary function of the norm of $\mathbf{x}$. □

A figurative illustration of a 2-dimensional generalized rotation applied on a 2-dimensional standard normally distributed vector is depicted in Fig. 4.3. As shown in this figure, a generalized rotation is a rotation whose angle may vary depending on the norm of the input vector. Fig. 4.3a contains the pdf of a joint normal 2-dimensional vector (s_1, s_2). It shows that performing rotations with different angles ϕ_1, ϕ_2 and ϕ_3, for different norms r_1, r_2 and r_3, respectively, twists the pdf but does not affect its bell shape. Fig. 4.3b shows the scatter plot of the $\mathbf{s}$ vector. As it can be seen from the figure, since the standard normal distribution is spherically symmetric, hence rotation invariant, a rotation of the points on a circle with specific radius, does not have any statistical effect.

Remark 3. *It can be easily shown that generalized rotations are invertible and their inverse is another generalized rotation.*

As it can be guessed from Fig. 4.3, generalized rotations do not affect jointly standard normal pdf's. In fact, this intuition is also supported by the following mathematical theorem.



(a) Probability density function (b) Scatter plot

Figure 4.3: Illustration of a generalized rotation; a rotation whose angle may vary depending on the norm of the input

Theorem 5. *If the inputs $x_1, x_2, \ldots, x_n$ of an n-dimensional generalized rotation $\mathbf{\Phi}$ are jointly normally distributed and mutually uncorrelated, hence mutually independent, as*

$$\rho_X(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n}}\, e^{\frac{-1}{2}(\mathbf{x}^\dagger \mathbf{x})} = \frac{1}{\sqrt{(2\pi)^n}}\, e^{\frac{-1}{2}(r^{\mathbf{x}})^2}, \tag{4.33}$$

then the outputs $\mathbf{y} = \mathbf{\Phi}(\mathbf{x})$ will be jointly normally distributed and mutually independent as well

$$\mathbf{y} = \mathbf{\Phi}(\mathbf{x}) \quad \Rightarrow \quad \rho_Y(\mathbf{y}) = \frac{1}{\sqrt{(2\pi)^n}}\, e^{\frac{-1}{2}(r^{\mathbf{y}})^2}. \tag{4.34}$$

In other words, according to the above theorem, n-dimensional generalized rotations preserve the normality characteristic of mutually independent signals.

Proof. According to (2.9), for any invertible function $\mathbf{\Phi}$, the pdf of $\mathbf{y}$ follows

$$\rho_Y(\mathbf{y}) = \frac{\rho_X(\mathbf{x})}{|\mathbf{J}_{\mathbf{\Phi}}|}. \tag{4.35}$$

According to (4.32) and considering (4.35) in the spherical coordinate system,

$|\mathbf{J}_{\mathbf{\Phi}}|$ can be calculated as

$$|\mathbf{J}_{\mathbf{\Phi}}| = \begin{vmatrix} 1 & 0 & \dots & 0 \\ \phi_1'(r^{\mathbf{x}}) & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{n-1}'(r^{\mathbf{x}}) & 0 & \dots & 1 \end{vmatrix} = 1 \tag{4.36}$$

where " $'$ " denotes derivative with respect to the input argument. This result also complies with the fact that the volume element changes by the absolute value of the Jacobian determinant of the transformation and that we expect (generalized) rotations not to change it.

Consequently, (4.35) can be calculated in the spherical coordinate system as

$$\rho_Y(\mathbf{y}) = \frac{\rho_X(\mathbf{x})}{1} = \frac{1}{\sqrt{(2\pi)^n}}\, e^{\frac{-1}{2}(r^{\mathbf{x}})^2} = \frac{1}{\sqrt{(2\pi)^n}}\, e^{\frac{-1}{2}(r^{\mathbf{y}})^2} \tag{4.37}$$

where the last equation comes from the face that $r^{\mathbf{y}} = r^{\mathbf{x}}$ according to the definition of a generalized rotation (4.32). □

It is also interesting to recall that the counter-example firstly introduced in [Babaie-Zadeh, 2002], showing that ICA fails in separating nonlinear mixtures, is a particular 2-dimensional generalized rotation (2.12),. The angle of the rotation in that example is designed such that it maps the square of $[-1, 1] \times [-1, 1]$ to itself, hence also preserves the uniform distribution on $[-1, 1] \times [-1, 1]$.

Finally, the following interesting Theorem 6, proposed in [Hamedani and Volkmer, 2001], claims that normality-preserving algebraic functions also preserve Euclidean norm. In fact, the authors of [Hamedani and Volkmer, 2001] have claimed to be informed by A. M. Kagan that V. L. Eidlin had passed away before publishing his proof and no one possessed a proof of this theorem. Therefore, it would be more accurate if Theorem 6 had been presented as a conjecture.

Theorem 6. *Let $\sigma > 0$ be a given number. Consider a random vector $\mathbf{x} = (x_1, x_2, \dots, x_n)^\dagger$ with every $x_j \sim \mathcal{N}(0, \sigma^2)$. Every algebraic transformation*

$\boldsymbol{\mathcal{A}}$ preserving normality of such a vector also preserves spheres. In other words, if $\mathbf{y} = \boldsymbol{\mathcal{A}}(\mathbf{x})$ is normally distributed, then

$$r^{\mathbf{y}} = \|\mathbf{y}\| = r^{\mathbf{x}} = \|\mathbf{x}\|, \tag{4.38}$$

where $\|\cdot\|$ represents Euclidean norm.

Given Theorem (conjecture) 6, it is straightforward to prove the following corollary.

Corollary 2. *If $\boldsymbol{\mathcal{A}}$ is an invertible algebraic function preserving normality, then the determinant of its Jacobian will be constant and equal to 1 everywhere, i.e.*

$$\det(\mathbf{J}_{\boldsymbol{\mathcal{A}}}) = |\mathbf{J}_{\boldsymbol{\mathcal{A}}}| = 1. \tag{4.39}$$

Proof. Since $\boldsymbol{\mathcal{A}}$ is assumed to be normality-preserving, it maps a random vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^\dagger$ with joint normal distribution into $\mathbf{y} = \boldsymbol{\mathcal{A}}(\mathbf{x})$ which will be normally distributed as well. Therefore, according to (2.9),

$$\rho_X(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n}}\, e^{\frac{-1}{2}(r^{\mathbf{x}})^2} = \rho_Y(\mathbf{y}) \times |\mathbf{J}_{\boldsymbol{\mathcal{A}}}| = \frac{1}{\sqrt{(2\pi)^n}}\, e^{\frac{-1}{2}(r^{\mathbf{y}})^2} \times |\mathbf{J}_{\boldsymbol{\mathcal{A}}}|, \tag{4.40}$$

where $\rho(\cdot)$ represents the pdf.

From theorem (conjecture) 6, we know that $\boldsymbol{\mathcal{A}}$ should preserve the norm, i.e. $r^{\mathbf{y}} = r^{\mathbf{x}}$. Consequently, (4.40) results in (4.39). □

It is interesting to note that although Theorem 3 was precisely proved, it could have also been easily proved using theorem (conjecture) 6.

4.3 Proposed Algorithm

In this section, we aim at proposing an algorithm for blind linearizing an invertible polynomial based on Theorem 3. Although, based on Section 4.2, this theorem holds for any invertible polynomial mapping, our proposed algorithm particularly focuses on polynomials, the inverse of which are also

polynomials. An example of this kind of polynomials is provided in Section 4.4.

In this case, it is necessary and sufficient for linearizing the mixture to estimate a polynomial $\mathbf{g}$ such that $\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t))$ is a vector with Gaussian distribution (see corollary 1 for the proof). Consequently, one can propose an algorithm which takes a cost function of "non-Gaussianity" and minimizes it with respect to the polynomial $\mathbf{g}$.

Here we assume a parametric model for the inverse polynomial $\mathbf{g}$ and then the optimization is done with respect to the parameters of our model. The parametric model of an L^{th} order polynomial of n signals is chosen as

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \\ \vdots \\ g_n(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta_1}^\dagger \\ \boldsymbol{\theta_2}^\dagger \\ \vdots \\ \boldsymbol{\theta_n}^\dagger \end{bmatrix} \mathbf{k}(\mathbf{x}) = \boldsymbol{\Theta}\mathbf{k}(\mathbf{x}) \tag{4.41}$$

where $\boldsymbol{\theta_i}$ for $i = 1, \ldots, n$ is a P-dimensional column vector of the parameters (constant scalars), $\mathbf{k}(\mathbf{x}) \in \mathbb{R}^{P\times 1}$ is the column vector containing all monomials with degree less than or equal to L and $P = \binom{n+L}{L} = \frac{(n+L)!}{n!L!}$ is the number of the parameters of each entry $g_i(\cdot)$ which is equal to the number of monomials with degree at most L. Note that this model is linear with respect to the parameters, which simplifies the algorithm significantly.

For any $1 \leq i \leq n$, the entropy of y_i is defined as

$$\mathrm{H}(y_i) = -\mathbb{E}\{\ln \rho_{Y_i}(y_i)\} \tag{4.42}$$

where $\rho_{Y_i}(y_i)$ is the pdf of the i^{th} output signal y_i. Consequently, the negentropy [Comon, 1994, Hyvärinen, 1999b] is calculated as

$$\boldsymbol{\mathcal{J}}(\mathbf{y}) = \mathbf{H}(\tilde{\mathbf{y}}) - \mathbf{H}(\mathbf{y}) \tag{4.43}$$

where $\tilde{\mathbf{y}}$ is a Gaussian random variable with the same co-variance matrix as $\mathbf{y}$'s.

It can be easily shown that among all distributions with a given mean and variance, Gaussian pdf is the one with the highest entropy; the value of the

Algorithm 3 Calculation of the Neg-Entropy

procedure NEG-ENTROPY ($\mathbf{y}(t)$)
  for $i = 1, \ldots, n$ **do**
    $\rho_{Y_i}(y_i) \leftarrow$ The estimated pdf of y_i based on the histogram of $y_i(t)$ for $t = 1, \ldots, T$
    $\sigma_i \leftarrow$ The variance of y_i
    $\mathrm{H}(y_i) \leftarrow -\mathbb{E}\{\ln \rho_{Y_i}(y_i)\}$
  end for
  $\boldsymbol{\mathcal{J}}(\mathbf{y}(t)) \leftarrow (1 + \ln(2\pi\boldsymbol{\sigma}))/2 - \mathbf{H}(\mathbf{y})$
end procedure

entropy of a random variable $\nu \sim \mathcal{N}(\mu, \sigma)$ is calculated as $\mathrm{H}(\nu) = \ln(\sigma\sqrt{2\pi e})$. Thus, neg-entropy is always nonnegative and invariant by any linear invertible transformation, and vanishes iff the signal is Gaussian. Therefore, as well as some previous works on BSS (e.g. [Girolami and Fyfe, 1996, Hyvärinen, 1999a]), we also use neg-entropy as a measure of Gaussianity. It should be emphasized that in this work, neg-entropy is the cost function that is *minimized*, because we need to recover the Gaussianity of the sources. While in classical BSS methods, it is maximized in order to retrieve non-Gaussianity.

The pseudo-code for calculating the proposed cost function is provided in Algorithm 3. In this algorithm the order of the inverse polynomial is assumed to be known (L).

Thus the algorithm should optimize

$$\underset{\boldsymbol{\Theta}}{\text{minimize}} \; \|\boldsymbol{\mathcal{J}}(\boldsymbol{\Theta}\mathbf{k}(\mathbf{x}))\|_2^2, \tag{4.44}$$

where $\|\cdot\|_2$ represents the ℓ_2 norm, i.e. Euclidean norm defined as $\|\mathbf{v}\|_2 = \sqrt[2]{v_1^2 + v_2^2 + \cdots + v_N^2}$ where $\mathbf{v} = [v_1, v_2, \ldots, v_N]$ is either a column or a row vector. Considering the fact that each entry of $\boldsymbol{\mathcal{J}}(\boldsymbol{\Theta}\mathbf{k}(\mathbf{x}))$ depends only on one row of $\boldsymbol{\Theta}$, minimizing all the entries of $\boldsymbol{\mathcal{J}}(\boldsymbol{\Theta}\mathbf{k}(\mathbf{x}))$ will be equivalent to minimizing its norm.

This cost function is not convex or even close to convex, hence seems to have too many local minima. In our simulations, classical optimization

Algorithm 4 Blind Linearization of Polynomial Mixtures of Gaussian Sources

1: $\mathbf{k}(\mathbf{x}) \leftarrow$ All monomials with degree less than or equal to L
2: $P \leftarrow \binom{n+L}{L} = \frac{(n+L)!}{n!L!}$
3: $\mathbf{\Theta}$: An $n \times P$ matrix of unknown parameters
4: $\mathbf{y}(t) \leftarrow \mathbf{g}(\mathbf{x}(t)) = \mathbf{\Theta}\mathbf{k}(\mathbf{x})$
5: **procedure** SIMULATED ANNEALING ($\|\text{Neg-Entropy}(\mathbf{y}(t))\|_2^2, \mathbf{\Theta}$)
⋮
6: **return** $\mathbf{\Theta}$
7: **end procedure**

methods like steepest descent and Newton always trapped in a local minimum (even for thousands of simulations with different random initialization). Therefore we had to implement a probabilistic method, e.g. particle swarm optimization [Kennedy, 2011] and simulated annealing [Hwang, 1988]. Finally we achieved the best performance by taking the minimum cost function among several runs of simulated annealing [Hwang, 1988] algorithm with different random initializations. It should be noted that even with simulated annealing, hundreds of simulations were needed to finally achieve the global minimum.

The pseudo-code of our proposed algorithm is provided in Algorithm 4. In this algorithm, the order of the inverse polynomial is assumed to be known (L), and the procedure *Simulated Annealing* of lines 5 to 7 corresponds to the traditional well-known simulated annealing algorithm [Hwang, 1988], which take some parameters and a cost function as inputs, and returns the optimal parameters.

Finally it should be noted that when the order of the inverse polynomial is not known, one can start from a linear polynomial, and gradually increase the order until getting a low enough cost function. In addition, considering the fact that the polynomial function is assumed to be invertible, one might confine the search over odd-valued polynomial functions or even monotonic

Algorithm 5 Iterative Blind Linearization of Polynomial Mixtures of Gaussian Sources

$L_0 \leftarrow 0$
repeat
  $L \leftarrow L_0 + 1$
  $L_0 \leftarrow L$
  $\mathbf{k}(\mathbf{x}) \leftarrow$ All monomials with degree less than or equal to L
  $P \leftarrow \binom{n+L}{L} = \frac{(n+L)!}{n!L!}$
  $\mathbf{\Theta}$: An $n \times P$ matrix of unknown parameters
  $\mathbf{y}(t) \leftarrow \mathbf{g}(\mathbf{x}(t)) = \mathbf{\Theta}\mathbf{k}(\mathbf{x})$
  procedure SIMULATED ANNEALING ($\|\text{Neg-Entropy}(\mathbf{y}(t))\|_2^2, \mathbf{\Theta}$)
  ⋮
    return $\mathbf{\Theta}$
  end procedure
until $\|\text{Neg-Entropy}(\mathbf{\Theta}\mathbf{k}(\mathbf{x}(t)))\|_2^2 > \epsilon$

functions. This idea can be implemented as Algorithm 5

4.4 Simulation Results

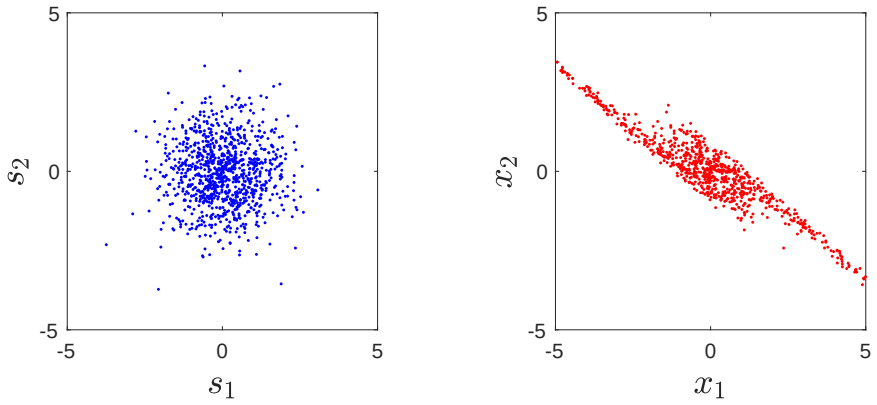
The main theorem proposed in this work is supported by a simple 2-by-2 simulated example as follows. The two sources s_1 and s_2 are randomly chosen as $\mathcal{N}(0,1)$ and are mixed through a 2-dimensional polynomial mapping as

$$\begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} s_1 + (s_1 + s_2)^3 \\ s_2 - (s_1 + s_2)^3 \end{bmatrix}. \tag{4.45}$$

The function (4.45) can be exactly inverted as

$$\begin{bmatrix} \hat{s}_1 \\ \hat{s}_2 \end{bmatrix} = \begin{bmatrix} x_1 - (x_1 + x_2)^3 \\ x_2 + (x_1 + x_2)^3 \end{bmatrix} \leftarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \tag{4.46}$$

From the scatter plot of the sources (Fig. 4.4a) and the observations (Fig. 4.4b), it is obvious that the observations (x_1, x_2) do not follow a Gaussian distribution.



(a) The sources (b) The observations

Figure 4.4: The scatter plot of the sources and the observations of (4.45) for 1000 samples. The neg-entropy for s_1, s_2, x_1 and x_2 are calculated 0.0524, 0.0476, 0.8664 and 1.1073 respectively.

Now, we want to retrieve the Gaussianity by applying a polynomial on the observations. In this experiment, given a cubic model with respect to the two signals x_1 and x_2 (i.e. with 10 parameters), we are looking for the parameters $\boldsymbol{\theta_1}^\dagger = [\theta_{10}, \ldots, \theta_{19}]$ in

$$y_1 = \theta_{10}x_1^3 + \theta_{11}x_1^2x_2 + \theta_{12}x_1^2 + \theta_{13}x_1x_2^2 + \theta_{14}x_1x_2 + \theta_{15}x_1 + \theta_{16}x_2^3 + \theta_{17}x_2^2 + \theta_{18}x_2 + \theta_{19} \quad (4.47)$$

such that y_1 follows a Gaussian distribution. To this end, as proposed in the previous section, the neg-entropy (4.43) of y_1 should be minimized with respect to the parameters $\boldsymbol{\theta_1}$ which leads to a linear mixture of s_1 and s_2.

Our simulations show that the 10-dimensional minimization of $\boldsymbol{\theta_1}$ is quite difficult mainly because of 1) too many local minima and non-convexity and 2) the high dimension of the space and the computational cost of the minimization. Thus, practically, numerous runs of the algorithm, each of which taking a long time to converge, were needed for in order to reach to a global minima. However, the following simulation results validate the proposed method by showing how the cost function behaves around its theoretical

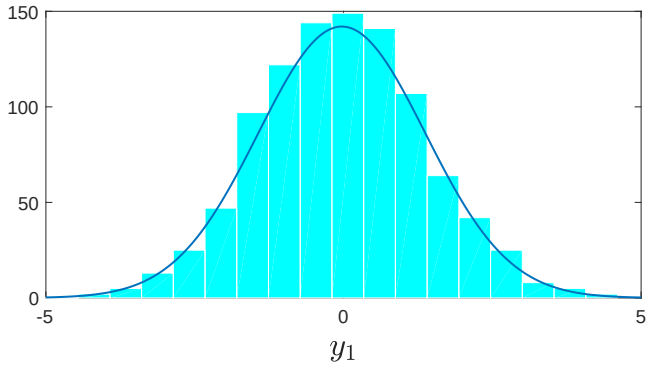


Figure 4.5: The histogram of $y_1 = x_1 + x_2 = s_1 + s_2$ from (4.45) for 1000 samples. The neg-entropy for y_1 is equal to 0.0535

global minima.

It should be noted that, since s_1 and s_2 are assumed to be mutually independent normal signals, any linear mixture of them, particularly the sum of them $s_1 + s_2$ also follows the normal distribution [Eisenberg and Sullivan, 2008], hence is a global minimizer of the neg-entropy. Therefore, linearizing algorithm does not necessarily converge to the exact inverse; evidently a scaled sum of the sources, i.e. $c(s_1 + s_2)$ where c is a constant coefficient, can be a convergence point for the proposed method. Particularly, it is interesting to see the behavior of the cost function (4.43) around $\boldsymbol{\theta_1} = [0, 0, 0, 0, 0, 1, 0, 0, 1, 0]$, where $y_1 = x_1 + x_2 = s_1 + s_2$ is expected to be a global minimizer. Fig. 4.5 shows how the histogram of the first output y_1 fits a Gaussian function.

Fig. 4.6 illustrates the partial variation of the neg-entropy with respect to any of entries of $\boldsymbol{\theta_1}$ around its optimal value $[0, 0, 0, 0, 0, 1, 0, 0, 1, 0]$. As declared in Algorithm 3, the neg-entropy is calculated through estimating the pdf of y_1 via the histogram technique. It is evident that although the neg-entropy is relatively far from zero in the neighborhood, it rapidly tends to zero (global minimum) for the exact optimal value. It should also be noted that changing θ_{19} does not affect the linearity of the mixture y_1 with

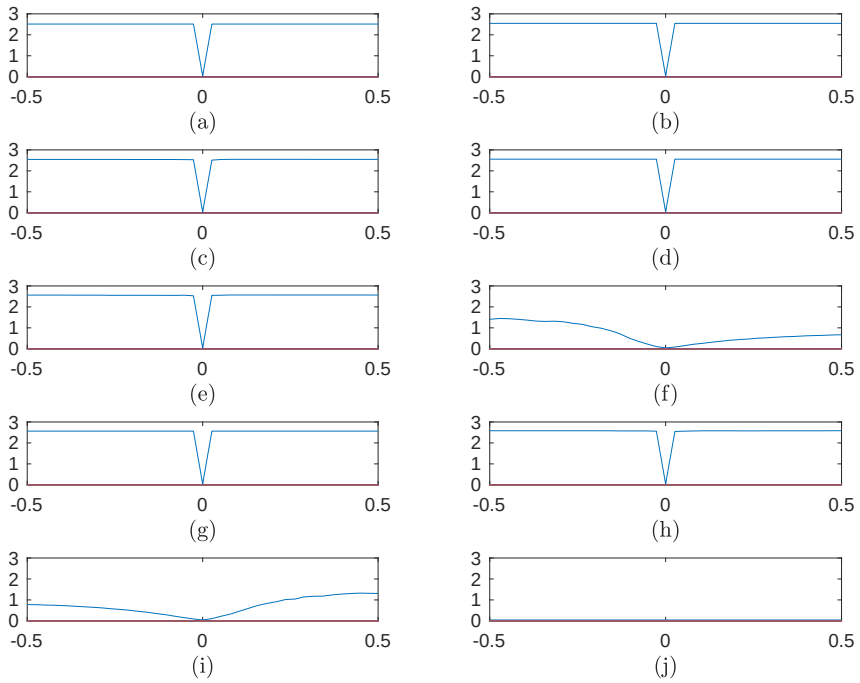
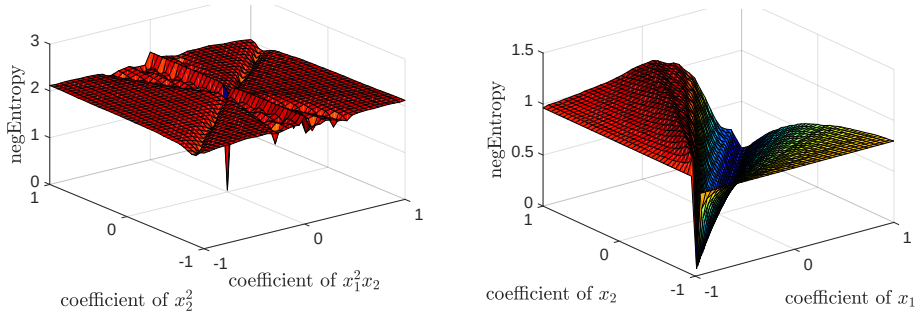


Figure 4.6: The neg-entropy of y_1 in (4.47) with respect to the entries of $\boldsymbol{\theta_1}$ centered around their optimal value $[0, 0, 0, 0, 0, 1, 0, 0, 1, 0]$ (from θ_{10} to θ_{19} in figures (a) to (j) respectively). Plotting with respect to each entry, the other parameters are kept constant.

respect to s_1 and s_2, hence does not change the neg-entropy.

Moreover, the value of the neg-entropy while simultaneously changing θ_{11} and θ_{17} around zero is plotted in Fig. 4.7a. As it can be seen in this figure, although the global minimum is in the origin, there are too many other local minima that may trap the minimizing algorithm. Fig. 4.7b also shows that the value of the neg-entropy is minimized with respect to the coefficients of x_1 and x_2 (while not changing the other parameters) as long as we stay on the line $\theta_{15} = \theta_{18}$ where the two coefficients are equal. This can also be mathematically seen that at any point of the line $\theta_{15} = \theta_{18}$, y_1 is a linear mixture of s_1 and s_2, hence follows a Gaussian pdf.



(a) The coefficients of $x_1^2x_2$ and x_2^2 (b) The coefficients of x_1 and x_2

Figure 4.7: The value of the neg-entropy of y_1 in (4.47) with respect to 2 coefficients of the parametric model, while the other parameters are kept constant and equal to their optimal value in $[0, 0, 0, 0, 0, 1, 0, 0, 1, 0]$.

4.5 Discussion and Future Works

In this chapter, nonlinear mappings preserving normality were studied. Although the only invertible polynomial which preserves the normality is a linear function, there are other normality-preserving nonlinear mappings including algebraic functions.

These theoretical results, as suggested in Section 4.3, can be used for blindly linearizing unknown nonlinear mixtures where the input follows normal distribution. In this application, the goal is to blindly transform a nonlinear system to a linear one, under the assumption that the sources are normally distributed. As a result of this approach, the nonlinear problem can be initially transformed to a linear one through a linearization pre-processing phase, and then be treated linearly.

Our proposed blind linearization approach could be used in some applications dealing with unknown polynomial nonlinearities. The idea of linearization has also been proposed in [Kagan et al., 1973] under the name NL model satisfying the addition theorem, where instead of normality, independence-preserving functions are considered. As an example, in nonlinear BSS problems, in order to transform the problem to a linear one, one may propose a

two-step separating scheme (see Fig. 4.1), where at the first step the mixture is linearized based on the result of this work, and the second step is a linear BSS method which can separate normal sources based on nonstationnarity [Pham, 2000] or correlation [Belouchrani et al., 1997]. However, this is a preliminary result and is to be extended and generalized in both theoretic and algorithmic aspects.

4.5.1 Theoretic Development

It would be interesting to discover other structured models of nonlinear functions that cannot preserve normality. For example, the simplest generalization of Theorem 3 might be reciprocal polynomials, i.e. polynomials with negative powers, or combinations of positive and negative powers for different sources.

Moreover, not all polynomials can be inverted by polynomials. So it is important to study the problem when the parametric model of the inverse function is not polynomial. Again, in this case, some special cases like reciprocals and rational function are of more interest.

In addition, in many practical applications, nonlinear mixtures are not exactly polynomials, but they can be approximated by polynomials. Thus it is interesting to see how a similar result can be achieved in those cases when the equations are not exact. Especially, it can be speculated that *normality-preserving functions that can be well approximated by polynomials are limited to be linear*, where by "well approximated" we mean with arbitrarily small error, i.e. functions that the coefficients of their Taylor expansion tend to zero as the order tends to infinity.

4.5.2 Algorithmic Development

In Section 4.3, neg-entropy is introduced as the cost function to be optimized. However, the minimization of the neg-entropy is too difficult because of the local minima and the computational cost. Thus it would be interesting

to develop algorithms based on other cost functions which may be more convex and simpler to calculate. Approximations of the neg-entropy (similar to [Hyvärinen, 1999a]) and cost functions based on higher order statistics are two examples that are suggested for future studies.

5 Nonlinear Mixtures of Sparse Sources

Contents

BSS problem for *linear* mixtures of sparse sources has already been studied e.g. in [Babaie-Zadeh et al., 2006, Bofill and Zibulevsky, 2001] (a very nice comprehensive survey on sparse component analysis for blind source separation is provided in [Gribonval and Lesage, 2006]). Results for separating nonlinear mixtures of sparse sources are limited to specific models, e.g. post-nonlinear mixtures [Van Vaerenbergh and Santamaría, 2006] and smart ion-selective electrode arrays [Duarte et al., 2009]. However, up to our best knowledge, it has not been considered for *general nonlinear mixtures* so far.

Our contribution in this chapter is performing nonlinear BSS for spatially sparse sources. Although our proposed separation algorithm in this chapter concerns determined cases, it can be shown that in this case, sources are separable even if the problem is under-determined, i.e. the number of observations is less than the number of source signals (see (2.4)). However, similar to the results of Chapter 3, an unknown nonlinear transformation of each source is reconstructed.

The idea of this chapter is original and has been partially published in [Ehsandoust et al., 2016]. The chapter is organized as follows. The problem model and the main idea for solving it is introduced in Section 5.1. The proposed approach and the algorithm for performing the separation are then proposed in Section 5.2. In this section, related background on both linear and nonlinear manifold learning and clustering is also reviewed. Simulation results are finally shown in Section 5.3, which is followed in Section 5.4 by a comprehensive discussion on the performance of the proposed approach and how to develop it for the future works.

5.1 Introduction

As mentioned earlier, in this chapter we are going to investigate the separability of nonlinearly mixed spatially sparse sources and mathematically formulate the proposed approach. For performing the separation, we have the following four assumptions on the sources:

1. Sources are instantaneously mutually independent,
2. Source signals are sparse in the space domain, i.e. they rarely take non-zero values at the same time,
3. The number of sources is equal to the number of observations,
4. The nonlinear mixing function $\mathbf{f}(\cdot)$ (see Fig. 1.1) is time-invariant.

A signal $s(t)$ is sparse, if it takes zero value with high probability. A

sparse signal is said to be "active" at time t_0, if its corresponding value is non-zero, i.e. $s(t_0) \neq 0$.

Definition A signal $s(t)$ is said to be κ-sparse if the fraction of the number of non-zero samples of a it over the total number of the samples is κ. Similarly, *activity rate* can be defined as the chance of the sparse signal being active. □

While being sparse refers to the time samples of the signal, unless otherwise stated, the sparsity can also be defined in other domains. For example, a signal may be sparse in the frequency domain, meaning that it has few frequency components.

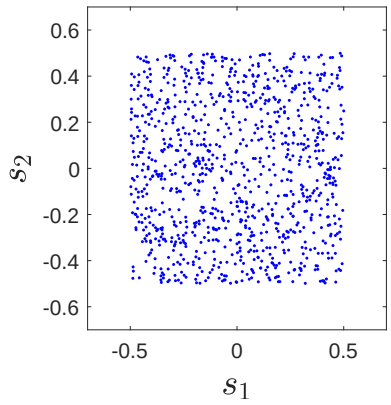
Similarly, *spatially* sparse signals can be defined as follows. Signals $s_1(t), \ldots, s_n(t)$ are said to be *spatially* sparse if it is quite rare that all of them are simultaneously active. In other words, if the signals are *spatially* sparse, the signal vector $\mathbf{s}(t) = [s_1(t), s_2(t), \ldots, s_n(t)]^\dagger$ has few non-zero entries at all time instants t. For example, people's talks in a meeting make a set of spatially sparse signals, because usually people do not talk at the same time, i.e. when someone speaks (is active), the others are silent.

Lemma 1. *If sources are individually sparse and mutually independent, they also make a spatially sparse set.*

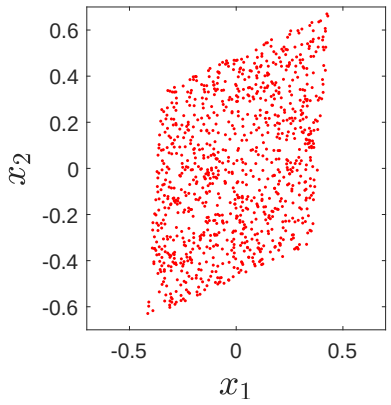
Proof. Let us assume that each source $s_i(t)$, $1 \leq i \leq n$ is κ_i-sparse, $0 < \kappa_i \ll 1$, and is ergodic. By definition, if a signal is ergodic, its statistical properties can be calculated from its time samples. Thus the probability of all sources being simultaneously active is equal to $\prod_{i=1}^{n} \kappa_i \approx 0$. □

If all spatially sparse signals have the same sparsity, i.e. $\kappa_i = \kappa_0$ for $1 \leq i \leq n$, most probably $\eta \approx n\kappa_0$ signals will be simultaneously active, thus the data mostly lies on η-dimensional manifolds in the n-dimensional space [Naini et al., 2008].

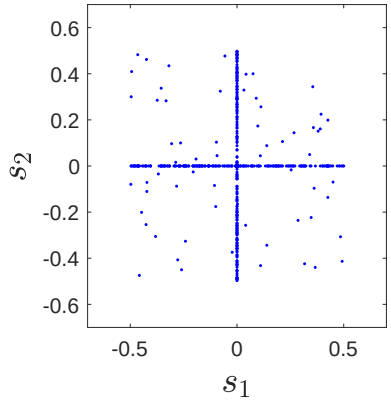
For investigating what would happen in the case of sparse sources from a geometrical point of view, the scatter plots of the observations for the two



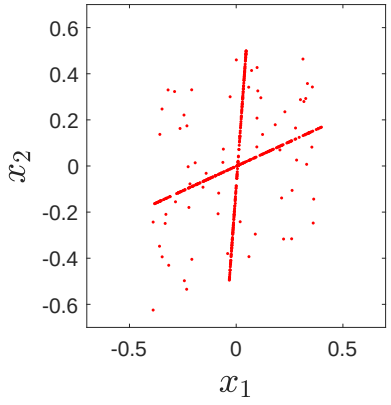
(a) Uniformly distributed sources



(b) A linear mixture of the uniformly distributed sources



(c) Sparse sources



(d) The linear mixture of the sparse sources

Figure 5.1: Comparing scatter plots of the source and observation vectors of a linear mixture, whether the sources are sparse or not

cases whether the sources are sparse or not, are compared in Figs. 5.1 (for a linear mixture) and 5.2 (for a nonlinear mixture).

The linear mixture is made by random 2×2 mixing matrix $\mathbf{A}$ as

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \mathbf{A} \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix}. \tag{5.1}$$

Fig. 5.2 is plotted for a nonlinear 2×2 mixing system of

$$x_1(t) = e^{s_1(t)} - e^{s_2(t)} \tag{5.2}$$

$$x_2(t) = e^{-s_1(t)} + e^{-s_2(t)} \tag{5.3}$$

where the observations $x_1(t)$ and $x_2(t)$ are centered before being plotted.

As it can be seen from the figures, when the sources are sparse (Figs. 5.2c and 5.1c), the samples of the source vector are mainly concentrated around the axes because it is quite rare that both of the sources take a non-zero value at the same time. So in this case, the scatter plot of the observations (Figs. 5.2d and 5.1d) contains two manifolds each of which is the result of the transformation of one of the axes in the source space.

In this work, we mainly consider that the signals are enough sparse so that the samples corresponding to more than one active source are very rare, i.e. $\eta = 1$. Thus, most of the samples lie on 1-dimensional manifolds corresponding to data where only one source is active. However, as it will be discussed in Section 5.4, the proposed approach does not fundamentally depend on this assumption, could be easily generalized for less sparse sources by minor modifications in the proposed method.

In order to better explain the idea which is proposed, let us start from separating linear mixtures. Then the proposed method can be generalized to the nonlinear mixtures which will be studied in the following.

5.1.1 Linear Mixtures

When the mixture is linear, the relationship between the sources and the observations can be written as

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \tag{5.4}$$

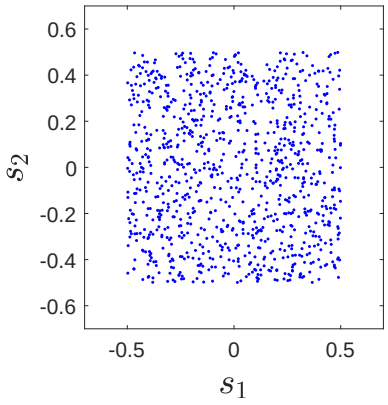
where $\mathbf{A}$ is an invertible $n \times n$ mixing matrix. As a consequence, axes in $\mathbf{s}$ domain will be transformed to direct lines in $\mathbf{x}$ space as

$$\mathbf{x} = \sum_{i=1}^{n} \mathbf{a}_i s_i, \tag{5.5}$$

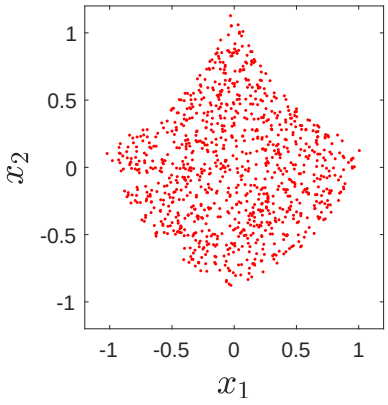
where $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$.

It can be shown through the following equations. From (5.4) we have

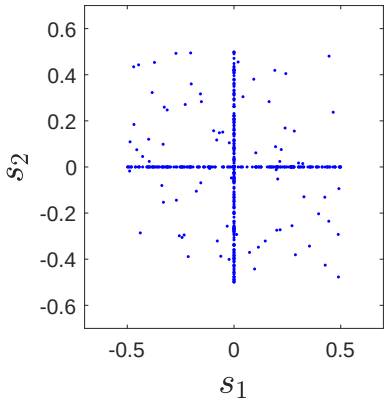
$$\mathbf{s}(t) = \mathbf{A}^{-1}\mathbf{x}(t) = \mathbf{B}\mathbf{x}(t) \tag{5.6}$$



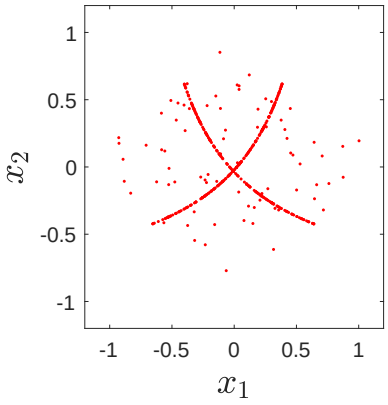
(a) Uniformly distributed sources



(b) Centered observations of (5.2) and (5.3) for uniformly distributed sources



(c) Sparse sources



(d) Centered observations of (5.2) and (5.3) for sparse sources

Figure 5.2: Comparing scatter plots of the source and observation vectors of the nonlinear mixture (5.2) and (5.3), whether the sources are sparse or not

where $\mathbf{B} = \mathbf{A}^{-1}$ is the inverse of the mixing matrix $\mathbf{A}$. So when only one of the sources, s_k, is active, (5.6) leads to

$$\begin{bmatrix} 0 \\ \vdots \\ 0 \\ s_k(t) \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{B} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} \tag{5.7}$$

and then

$$\Rightarrow \ \forall 1 \leq i \neq k \leq n \qquad \mathbf{b}_i^{\dagger}\mathbf{x}(t) = 0 \tag{5.8}$$

where $\mathbf{b}_i^{\dagger}$ is the i^{th} row of $\mathbf{B}$. Therefore, when s_k is the only active source, the observation vector satisfies (5.8) which determines a line set in n-dimensional space.

The separability of this model is proven in [Babaie-Zadeh et al., 2006, Bofill and Zibulevsky, 2001], and the separation algorithms are also provided. Nonetheless, it would be interesting to see what happens when the n sources are less sparse such that most probably $n-1$ of them are simultaneously active [Rivet, 2006, Rivet et al., 2010]. This case is studied in Appendix A, where the separability of the mixture is proved in Theorem 7.

To conclude, it is shown that when n spatially sparse mutually independent sources are mixed linearly, the scatter plot of the observation vector consists of low dimensional subspaces. Learning these subspaces leads to construct the separating matrix. This idea can be generalized for the nonlinear case which is elaborated in the following.

5.1.2 Nonlinear Mixtures

As mentioned in the beginning of Section 5.1 and shown in Fig. 5.2, and similar to the linear model, nonlinear mixtures of n spatially sparse mutually independent sources, with high probability lie on η-dimensional manifold in

n-dimensional space ($\eta < n$). As stated before, for the rest of this chapter, it is assumed that the sources are enough sparse such that rarely more that one of them are simultaneously active, i.e. $\eta = 1$. Given this assumption, the n-dimensional observation space comprises n 1-dimensional manifolds, each of which corresponds to exactly one of the sources.

Mathematically speaking, using the same notation as in Chapter 3, the nonlinear mixture is modeled as $\mathbf{x}(t) = \mathbf{f}(\mathbf{s}(t))$. The model can be inverted as $\mathbf{s}(t) = \mathbf{g}(\mathbf{x}(t))$ where $\mathbf{g} = \mathbf{f}^{-1}$ is the inverse function.

If only one source is active, i.e. $s_k(t) \neq 0$ and for all $1 \leq i \neq k \leq n$, $s_k(t) = 0$, we will have

$$\begin{bmatrix} 0 \\ \vdots \\ 0 \\ s_k(t) \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{g} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} \tag{5.9}$$

and then

$$\Rightarrow \ \forall 1 \leq i \neq k \leq n \qquad \mathbf{g}_i(\mathbf{x}(t)) = 0 \tag{5.10}$$

where $\mathbf{g}_i$ is the i^{th} component of the n-dimensional nonlinear function $\mathbf{g}$. Consequently, 1-dimensional manifolds Γ_k can be defined as

$$\Gamma_k : \quad \forall 1 \leq i \neq k \leq n \qquad \mathbf{g}_i(\mathbf{x}(t)) = 0, \tag{5.11}$$

which determine intersections of $n-1$ $(n-1)$-dimensional manifolds $\mathbf{g}_i(\mathbf{x}(t)) = 0$ in the n-dimensional $\mathbf{x}$ space.

This is the main idea for performing nonlinear BSS for spatially sparse sources, which is elaborated in Section 5.2. So the mixing model that is concerned in this work is not restricted to a specific kind and can be any invertible function. The idea in [Babaie-Zadeh et al., 2002] for separating post-nonlinear mixtures of bounded signals is also very close to the sparsity

where the edges of the parallelogram of the source scatter plot is utilized to learn the nonlinearity.

It should be noted once more that the goal of BSS is to "separate" the sources and not to "reconstruct" them. In nonlinear BSS, a component-wise nonlinear function remains as an ambiguity in reconstructing the sources that can only be resolved using other prior information about the source signals, which is out of the scope of BSS.

5.2 Proposed Method

Based on the results of the previous section, we are now going to propose an approach to separate sparse sources which are nonlinearly mixed through an unknown mixing function. The algorithm consists of two steps:

1. Clustering the observations and manifold learning,
2. Separating the sources,

In the first step, n 1-dimensional manifolds in the observation space are learned and the data is clustered so that each class corresponds to the activity of one of the sources. Then the sources are reconstructed based on subsection 5.1.2.

As mentioned before, the output of the last step will be a component-wise nonlinear function of the source vector, which can be considered as nonlinear distortion. Thus, in an additional post-processing step, a signal restoration technique can be proposed aiming at blind compensating the nonlinear distortion of the sources.

5.2.1 Clustering and Multiple Manifold Learning

The first step in the proposed algorithm is to cluster the observation points due to the manifolds that they lie on. It means that the n 1-dimensional manifolds, Γ_k of (5.11) for $k = 1, \ldots, n$, should be learned simultaneously.

In Appendix B we provide a relatively deep investigation on the problem of manifold clustering followed by proposing robust algorithms, which may also be used separately for other applications in signal processing and pattern recognition. So Appendix B concerns a more general definition of manifold clustering problem in the sense that 1) additive noise is considered, 2) the number of the manifolds and their dimensions are not necessarily equal to n and 1, respectively (but it is assumed that they are given in advance).

In the proposed method, we have used the non-parametric multiple manifold learning method proposed in Section B.4 of Appendix B. In this approach, the manifolds are learned based on an iterative method similar to the well-known k-means [MacQueen, 1967]. Our method comprises three steps as follows.

1. Initially, data points are randomly assigned to the manifolds. The label of each point $\mathbf{x}(t)$ for $t = 1, \ldots, T$ at r^{th} iteration is represented by $\Omega^{(r)}(t) \in \{1, 2, \ldots, n\}$).

2. A 1-dimensional manifold is fitted on the points assigned to each class using smoothing splines as

$$\forall\ 1 \leq i \leq n \quad \Omega_i^{(r)} : \mathcal{F}(\{\mathbf{x}(t)\} | \Omega^{(r-1)}(t) = i) \tag{5.12}$$

   where $\mathcal{F}$ represents the 1-dimensional smoothing spline procedure and the superscript (r) denotes the number of the iteration.

3. The labels of data points are updated to their closest manifold as

$$\forall\ 1 \leq t \leq T \quad \Omega^{(r)}(t) = \underset{i}{\operatorname{argmin}} \Big(\mathrm{d}_w^2(\mathbf{x}(t), \Omega_i^{(r)}) \Big), \tag{5.13}$$

   where d_w denotes a weighted distance from a point to a manifold.

The sequencing steps 2 and 3 should be iteratively repeated until the algorithm converges and the labels $\Omega^{(r)}(t)$ do not change.

Please refer to Section B.4 of Appendix B for more details, and Algorithm 6 for the pseudo-code of the proposed algorithm.

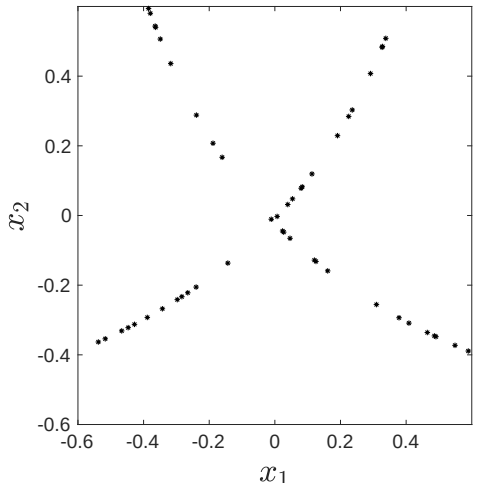


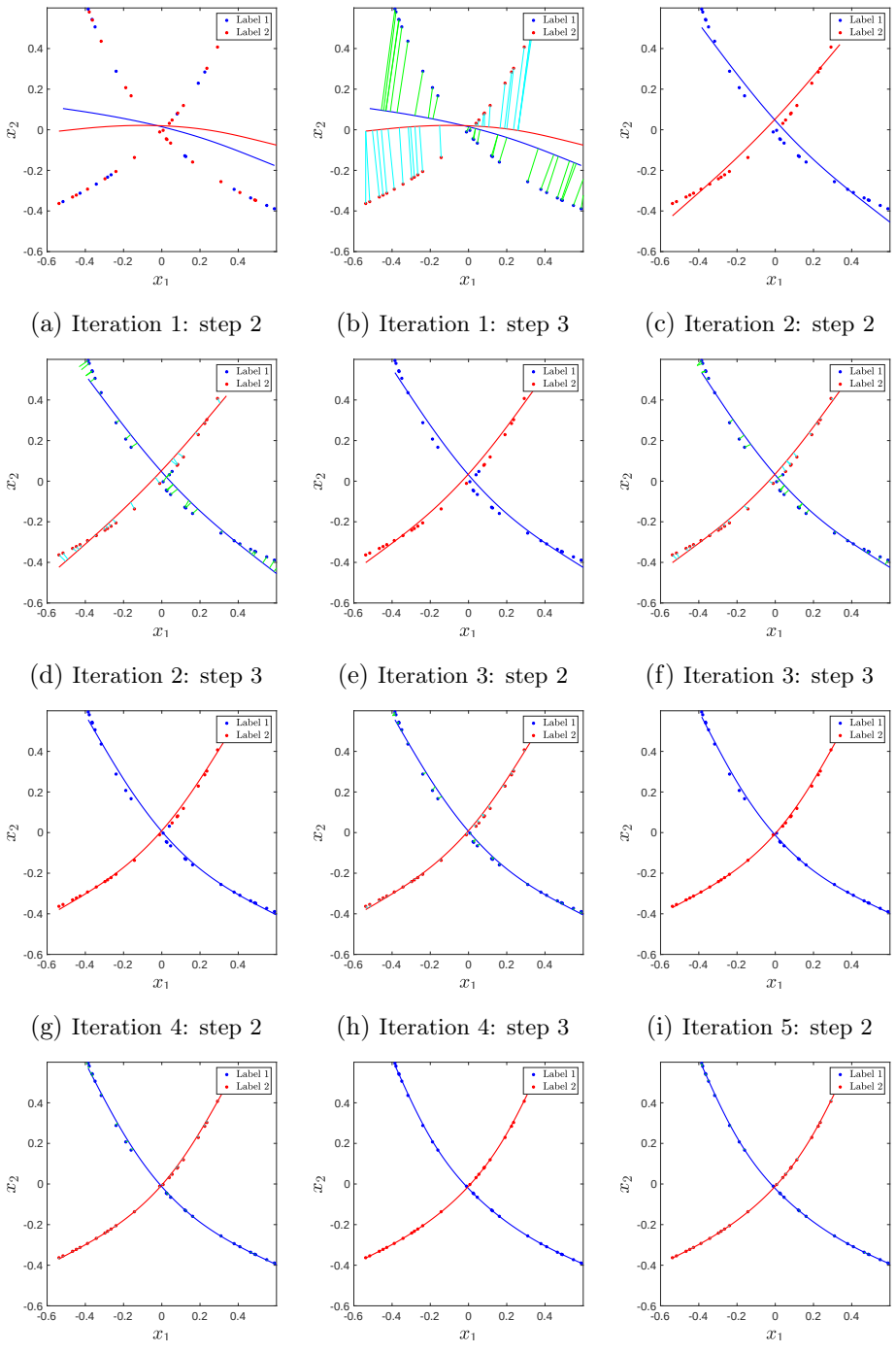
Figure 5.3: Observation data points of (5.2) and (5.3), which are going to be clustered using the proposed non-parametric approach

The proposed idea can be well illustrated visually. For example, assume two observations $x_1(t)$ and $x_2(t)$ for $t = 1, \ldots, 50$, which are realized using (5.2) and (5.3) from uniformly distributed source signals (see Fig. 5.3).

In order to cluster the data and learn the two manifolds simultaneously, the proposed non-parametric approach is utilized, and the outputs of each step in every iteration is depicted in the sequence of figures 5.4a to 5.4l. In this simulation, the data is assumed not to contain outliers, hence distances are not weighted. In these figures, two 1-dimensional manifolds in a 2-dimensional space are to be clustered and learned.

In Fig. 5.4a, each data point is randomly assigned to either red or blue class. Then, performing the smoothing splines algorithm on red (respectively, blue) points has resulted the red (respectively, blue) manifold. Then in Fig. 5.4b, distances of all data points to both red and blue manifolds are calculated and the label (color) of the points are updated to the color of their closest manifold. The algorithm iteratively does these procedures until it converges.

As shown in Fig. 5.4, the proposed algorithm has converged to the global



(a) Iteration 1: step 2 (b) Iteration 1: step 3 (c) Iteration 2: step 2

(d) Iteration 2: step 3 (e) Iteration 3: step 2 (f) Iteration 3: step 3

(g) Iteration 4: step 2 (h) Iteration 4: step 3 (i) Iteration 5: step 2

(j) Iteration 5: step 3 (k) Iteration 6: step 2 (l) Iteration 6: step 3

Figure 5.4: Illustration of the proposed non-parametric approach for learning 2 manifolds in 2-dimensional space; in figures corresponding to step 3, the *minimum* distance of each point to the manifolds is plotted

minimum in 6 iterations. As stated in Appendix B, if the weighting function used in (5.13) for calculating $\mathrm{d}_w^2(\mathbf{x}(t), \Omega_i^{(r)})$ is monotonic, the clustering error in this approach may not increase as the algorithm progresses, hence the proposed algorithm does converge. However, in order to avoid being trapped in local minima, it is necessary to run the algorithm several times, each with a different random initialization, and finally take the best result.

Coming to a conclusion, the first step of the proposed framework is clustering the manifolds in the observations space. The outputs of this step are the n 1-dimensional manifolds of (5.11) in the observation space that fit the data the best.

5.2.2 Separating the Sources

As mentioned before, each manifold in the observation space corresponds to the activity of only one of the sources. So once the manifolds are learned, sources are separated. In fact, for any time instant $t = 1, \ldots, T$, if $\mathbf{x}(t)$ belongs to manifold Γ_i, then the sources are reconstructed as

$$\forall\ 1 \leq t \leq T \quad \mathbf{x}(t) \in \Gamma_i \Rightarrow \begin{cases} \hat{y}_i(t) = \Xi_i(\mathbf{x}(t)) \\ \hat{y}_j(t) = 0 \qquad 1 \leq j \neq i \leq n \end{cases} \tag{5.14}$$

where $\hat{\mathbf{y}}(t) = [\hat{y}_1(t), \ldots, \hat{y}_n(t)]^\dagger$[1] is the reconstruction of the sources, and Ξ_k is an arbitrary nonlinear function.

Please note that $\mathbf{x}(t) \in \Gamma_i$ means that only s_i is active, i.e. $\mathbf{x}(t)$, hence any function of that $\Xi_i(\mathbf{x}(t))$ will only be a function of s_i as well. Thus, although a nonlinear distortion is remained as an ambiguity, the sources are separated and BSS is done.

One of simplest possibilities for $\Xi_i(\cdot)$ can be suggested as $\Xi_i(\mathbf{x}(t)) = x_k(t)$ for arbitrary $1 \leq k \leq n$, i.e. the k^{th} observation signal. However, it should be noted that the nonlinear function $\Xi_i(\cdot)$ should be an injection (one-to-one), hence invertible in its domain. For example, in order to reconstruct

[1] $\hat{\mathbf{y}}$ is not the final estimation of the sources, this is the reason why a "hat" is used in the notation

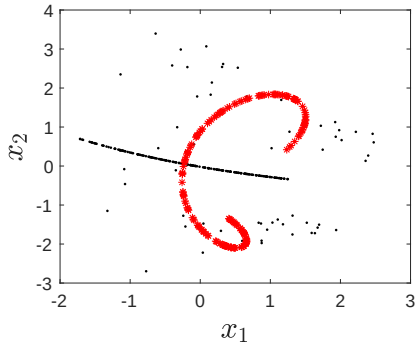


Figure 5.5: A manifold whose projections on the axes are not invertible

the source corresponding to the red data points in 2×2 example of Fig. 5.5, none of the axes (neither x_1, nor x_2) could be taken as either $\Xi_1(\cdot)$ or $\Xi_2(\cdot)$. In this figure, observations are constructed as

$$x_1 = -3s_2 + 0.76\cos(3.5s_1) + 0.43\sin(3.5s_1) - \cos(7s_1) - 0.24, \quad (5.15)$$

$$x_2 = e^{s_2}(0.57 + 1.24\sin(3.5s_1) + 0.43\cos(3.5s_1) + \sin(7s_1)). \quad (5.16)$$

A good choice for $\Xi_i(\cdot)$ can be based on a nonlinear dimension reduction algorithm (e.g. ISOMAP [Tenenbaum et al., 2000] and diffusion maps [Talmon et al., 2013]), which is supposed to transform 1-dimensional manifolds to direct lines. Besides, prior knowledge about either the sources or the unknown nonlinear mapping may even lead to find a $\Xi_i(\cdot)$ which restores the sources and resolve the ambiguity in source reconstruction.

Nonetheless, as declared in Section B.2, probably there are few time instants where more than one source are simultaneously active. These data points in the first step of the proposed method (Section 5.2.1) are found as outliers. Since outliers do not lie on any manifold, their corresponding reconstructed sources will be different from (5.14).

5.2.2.1 Separating the Outliers

Please note that the problem of estimating the sources corresponding to the outliers is ill-posed. Since the observations are mainly concentrated close to

1-dimensional manifolds, too little data is in hand to estimate the unknown mixing function $\mathbf{f}$ for the rest of the n-dimensional space. In other words, although looking at the scatter plot of the observations, marginal values of the n-dimensional function $\mathbf{f}$ are learned, infinite n-dimensional functions may have the same marginals. Thus observations do not contain enough information for separating the outliers.

Nevertheless, given some prior knowledge about either specifications of the nonlinear mapping or characteristics of the sources, the separation can be approximately achieved for the outliers. Two different approaches can be suggested for the separation of outliers which will be introduced in the following.

5.2.2.1.1 Signal-Dependent Methods One approach is to estimate the sources in case of outliers based on other estimated values for them, i.e. inliers. Through this approach, sources are firstly reconstructed for inlier observations, without any estimation in case of outliers. Then each source is individually processed in order that the missing samples are estimated based on the known ones.

This problem is known as *signal restoration*, which has been well studied in the literature. The restoration filter is usually designed by trying to retrieve known characteristics of the signal, e.g. band-limited, sparse in a domain, bounded amplitude, and so forth.

For example, [Duarte et al., 2015] has considered band-limited signals, hence sparse in the frequency domain. In this case, a nonlinear transformation of a signal will generate harmonics, which leads to enlarging the bandwidth, hence lessens its sparsity in the frequency domain. Thus, a nonlinear transformation can be applied on each "pure signal" for restoring the sparsest possible signals in the frequency domain. It should be emphasized that this is a different assumption from the sources being *spatially* sparse, which is the main assumption of this chapter.

Nonetheless, considering (5.14), the separated sources in this approach

can be expressed as

$$\forall 1 \leq i \leq n \quad y_i(t) = \begin{cases} \hat{y}_i(t) = \Xi_i(\mathbf{x}(t)) & \mathbf{x}(t) \in \Gamma_i \\ 0 & \mathbf{x}(t) \in \Gamma_j;\ 1 \leq j \neq i \leq n \\ \phi\left(\{\hat{y}_i\}_{t \in \mathcal{T}}\right) & \text{else } (\mathbf{x}(t) \text{ is outlier}) \end{cases} \tag{5.17}$$

where $\mathcal{T}$ denotes the set of time indexes that their corresponding observation $\mathbf{x}(t)$ belong to Γ_i, and $\phi(\cdot)$ is a restoration function which takes the already-estimated samples $\hat{y}_i$ as input, and provides an estimation of the corresponding source in case of outliers.

5.2.2.1.2 Mixture-Dependent Methods The other class of methods for performing the estimation is based on the separated manifolds and assumptions on the mixing function. In this approach, we propose nonlinear projections of the outliers on the learned manifolds as estimations for corresponding sources. However, the nonlinear projection is not unique, and the accurate projection for separating the outliers needs side-information about the mixing model.

One of the methods for performing the nonlinear projection can be defined in accordance with the concept of "curvilinear coordinate systems". In geometry, curvilinear coordinates are coordinate systems for Euclidean space in which the coordinate lines may be curved. They can be seen as a generalization of linear or affine coordinate systems. Well-known examples of curvilinear coordinate systems in three-dimensional Euclidean space ($\mathbb{R}^3$) are cylindrical and spherical polar coordinates.

In the Cartesian system, the standard basis vectors can be derived from the derivative of the location of point $\mathbf{p}$ with respect to the local coordinates.

For example in 3-dimensional space,

$$\mathbf{e}_1 = \mathbf{e}_x = \frac{\partial r^{\mathbf{p}}}{\partial x} \tag{5.18}$$

$$\mathbf{e}_2 = \mathbf{e}_y = \frac{\partial r^{\mathbf{p}}}{\partial y} \tag{5.19}$$

$$\mathbf{e}_3 = \mathbf{e}_z = \frac{\partial r^{\mathbf{p}}}{\partial z}, \tag{5.20}$$

where $\mathbf{e}_i$ represents the i^{th} basis vector and $r^{\mathbf{p}} = \|\mathbf{p}\|$ is the Euclidean norm of $\mathbf{p}$. Applying the same derivatives to the curvilinear system locally at point $\mathbf{p}$ defines the natural basis vectors as

$$\mathbf{e_1} = \frac{\partial r^{\mathbf{p}}}{\partial \epsilon_1},\ \mathbf{e_2} = \frac{\partial r^{\mathbf{p}}}{\partial \epsilon_2},\ \mathbf{e_3} = \frac{\partial r^{\mathbf{p}}}{\partial \epsilon_3}, \tag{5.21}$$

where $(\epsilon_1, \epsilon_2, \epsilon_3)$ represents the coordinates in the curvilinear system.

Such a basis, whose vectors change their direction and/or magnitude from point to point is called a local basis. All bases associated with curvilinear coordinates are necessarily local. Basis vectors that are the same at all points are global bases, and can be associated only with linear or affine coordinate systems.

In this approach, the n learned 1-dimensional manifolds play the role of "coordinate curves", based on which the coordinates of the outliers are to be estimated. Since each manifold corresponds to the activity of only one source, the coordinates of the outliers based on them are introduced as the estimations of the sources.

This concept is illustrated in Fig. 5.6. In this figure, the projections of the point $\mathbf{p}$ onto the curvy axes denote the values of the corresponding sources.

The source estimation based on this idea is not always exact, and depends on the off-diagonals of Hessian of the nonlinearities. In other words, our proposed nonlinear projection is exact iff all components of the unknown mapping, $f_i(\mathbf{s})$ for $i = 1, \ldots, n$, have diagonal Hessian matrices. The Hessian of a function $f_i(\mathbf{s}) : \mathbb{R}^n \to \mathbb{R}$ comprises of all second partial derivatives of

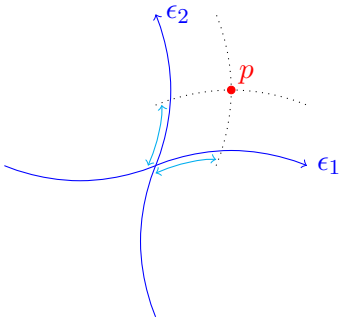


Figure 5.6: The illustration of our proposed nonlinear projection based on curvilinear coordinate system

that, and is defined as

$$\mathbf{H}_{f_i} = \begin{bmatrix} \frac{\partial^2 f_i}{\partial s_1^2} & \frac{\partial^2 f_i}{\partial s_1 \partial s_2} & \cdots & \frac{\partial^2 f_i}{\partial s_1 \partial s_n} \\ \frac{\partial^2 f_i}{\partial s_2 \partial s_1} & \frac{\partial^2 f_i}{\partial s_2^2} & \cdots & \frac{\partial^2 f_i}{\partial s_2 \partial s_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f_i}{\partial s_n \partial s_1} & \frac{\partial^2 f_i}{\partial s_n \partial s_2} & \cdots & \frac{\partial^2 f_i}{\partial s_n^2} \end{bmatrix}. \tag{5.22}$$

Evidently, the Hessian of $f_i(\mathbf{s})$ is diagonal iff

$$f_i(\mathbf{s}) = f_i^{(1)}(s_1) + f_i^{(2)}(s_2) + \cdots + f_i^{(n)}(s_n) + c_i \tag{5.23}$$

where for all $j = 1, \ldots, n$, $f_i^{(j)}(\cdot)$ is an $\mathbb{R} \to \mathbb{R}$ nonlinear function, and c_i is scalar. Considering (5.23) for all $j = 1, \ldots, n$ restricts the n-dimensional function $\mathbf{f}$ to be formulated as

$$\mathbf{f}(\mathbf{s}) = \mathbf{f}^{(1)}(s_1) + \mathbf{f}^{(2)}(s_2) + \cdots + \mathbf{f}^{(n)}(s_n) + \mathbf{c} \tag{5.24}$$

where or all $j = 1, \ldots, n$, $\mathbf{f}^{(j)}(\cdot) = [f_1^{(j)}(\cdot), f_2^{(j)}(\cdot), \ldots, f_n^{(j)}(\cdot)]^\dagger$ and $\mathbf{c} = [c_1, c_2, \ldots, c_n]^\dagger$. In other words, the Hessian of $f_i(\mathbf{s})$ for $i = 1, \ldots, n$ is diagonal iff the mixing function $\mathbf{f}$ is a linear mixture of nonlinearly filtered sources, i.e. a linear mixture of sources with distortions.

Considering this structure in our separation problem, for all $1 \leq i \leq n$, the manifold Γ_i corresponding to the activity of only one source s_i, would be the set of $\mathbf{f}^{(i)}(s_i)$. As a consequence, for time instants when more than

one source are simultaneously active, observed data will be a *linear* mixture of corresponding points on the learned manifolds.

Other methods of nonlinear projection may be proposed based on known specifications of the mixture model. For example preserving the local angles, the time-derivative being continuous, following a parametric model, and so forth, may lead to different different projections, hence different estimations of the sources signals.

Any method of the nonlinear projection imposes some restrictions on the mixture model, and should be chosen regarding the application. Nevertheless, the separated sources in this approach can be written as

$$\forall 1 \leq i \leq n \quad y_i(t) = \begin{cases} \hat{y}_i(t) = \Xi_i(\mathbf{x}(t)) & \mathbf{x}(t) \in \Gamma_i \\ 0 & \mathbf{x}(t) \in \Gamma_j;\ 1 \leq j \neq i \leq n \\ \Xi_i(\mathbf{x}_i^\circ(t)) & \text{else } (\mathbf{x}(t) \text{ is outlier}) \end{cases} \tag{5.25}$$

where $\mathbf{x}_i^\circ(t)$ is the nonlinear projection of $\mathbf{x}(t)$ onto Γ_i.

5.3 Simulation Results

In order to simulate the proposed algorithm, we have used three 2×2 nonlinear mixing models and one linear one. The simulated algorithm, as proposed previously, consists of the following steps:

1. Outliers are detected via a hard threshold weighting based on (B.8) and (B.9), hence not considered in the clustering step.

2. The outlier-free data is clustered into two classes, each of which corresponding to a manifold in the observation space. For this purpose, a parametric approach is employed based on Section B.3, where the parametric model is assumed to be polynomial. Since the order of the polynomial is not known, the algorithm starts from the first order (i.e. a linear model), and gradually increases the order until the fitting error is low enough.

3. The reconstruction of the sources is then performed, as suggested in Section 5.2.2, i.e. based on some assumptions on the mixing function, where the functions Ξ_1 and Ξ_2 are chosen as $\Xi_1(\mathbf{x}(t)) = \Xi_2(\mathbf{x}(t)) = x_1(t)$.

4. Finally, the outliers are separated using the nonlinear projection based on curvilinear coordinates system, which is a mixture-dependent method proposed in Section 5.2.2.1.2.

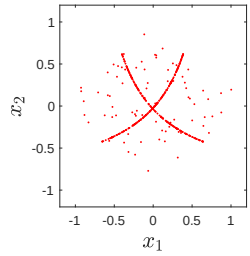
In order to see the efficiency of the simulated algorithm in separating the sources, separated sources (outputs) are plotted versus original ones. As pointed in Section 3.4.3, thickness of this plot represents the separation error. It is shown that in all simulations, the proposed method has efficiently separated the sources.

Simulation results are provided in figures 5.7 to 5.10. In each figure, (a) contains the scatter plot of the observations. Then in part (b), in addition to the observation scatter plot, the two learned manifolds are also plotted in green and purple. Moreover, outliers are shown by black crosses, and data points corresponding to the green (respectively, purple) manifold are plotted in blue (respectively, red), hence the classification is apparent. Parts (c) and (d) of the figures contain the separated signals versus the original sources.

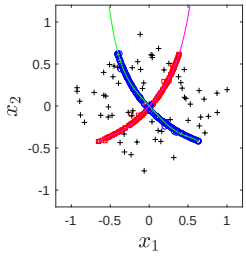
It should be mentioned that the sources in all simulations are 1000 samples of two sparse sources (with the activity rate of 25%) that are uniformly distributed in $[-0.5, 0.5]$ when they are active. The sources are not included in the figures, in order to avoid repetition.

As shown in Fig. 5.7b, the clustering method has clustered the data with very few errors, while the learned manifolds are very well fitted to the data. According to Figs. 5.7c and 5.7d, the sources are well separated and each separated source is a function of only one source. Otherwise, the scatter plot would not be a function and y_1 (respectively, y_2) would take different values for a given s_1 (respectively, s_2).

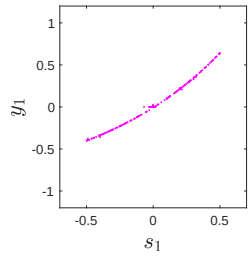
The second simulation aims at evaluating how the algorithm can handle complicated nonlinearities. As it can be seen in Fig. 5.8b, although the



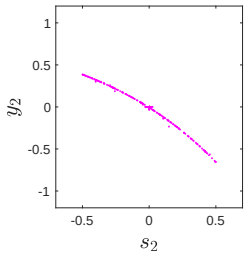
(a) Centered observations



(b) Clustered data (into blue and red classes)

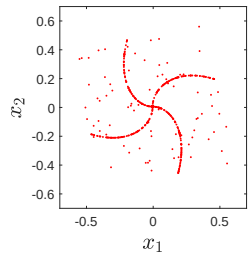


(c) The 1st signal

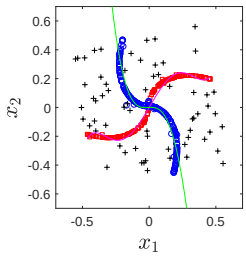


(d) The 2nd signal

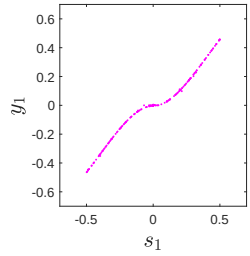
Figure 5.7: Simulation results for $x_1(t) = e^{s_1(t)} - e^{s_2(t)}$ and $x_2(t) = e^{-s_1(t)} + e^{-s_2(t)}$; observations based on (5.2) and (5.3)



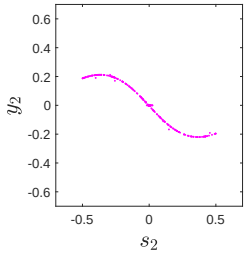
(a) Centered observations



(b) Clustered data (into blue and red classes)

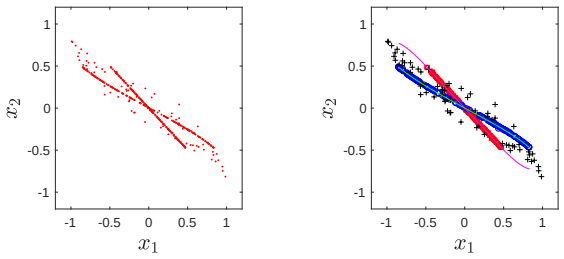


(c) The 1st signal

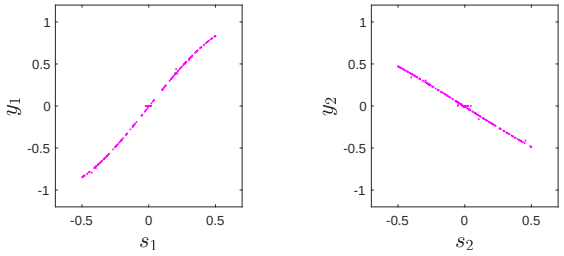


(d) The 2nd signal

Figure 5.8: Simulation results for $x_1(t) = \cos(\alpha(t))s_1(t) - \sin(\alpha(t))s_2(t)$ and $x_2(t) = \sin(\alpha(t))s_1(t) + \cos(\alpha(t))s_2(t)$ where $\alpha(t) = \frac{\pi}{2}(1 - \sqrt{s_1^2(t) + s_2^2(t)})^2$



(a) Centered observations (b) Clustered data (into blue and red classes)



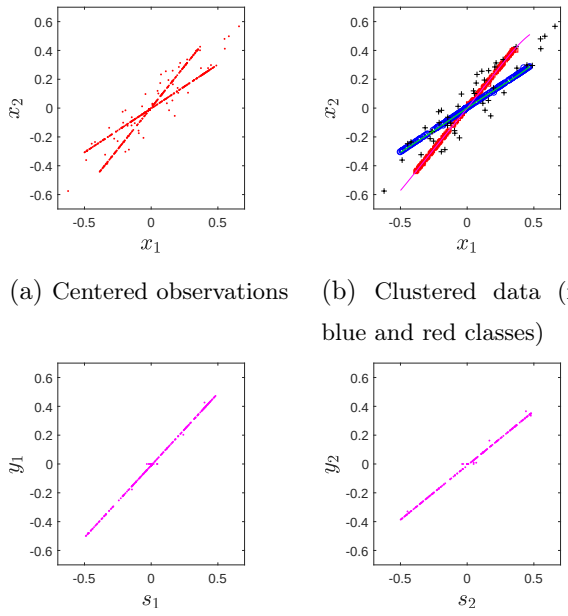
(c) The 1st signal (d) The 2nd signal

Figure 5.9: Simulation results for $x_1(t) = \sin(2s_1(t) - s_2(t))$ and $x_2(t) = \sin(s_2(t) - s_1(t))$

mixture, hence the manifolds, are relatively complicated, the algorithm has relatively well classified the data and has learned the manifolds with acceptable errors. Note that the implemented clustering algorithm is based on a parametric polynomial. Thus, since the mixture in this simulation is very far from polynomials, it was expected that the learned manifolds do not exactly fit the data.

The simulation of Fig. 5.9 is designed such that the manifolds are close to each other, which might make it more difficult for the clustering algorithm to perform correctly. However, Fig. Fig. 5.9b proves that it works successfully with a quite acceptable error. In fact, most of the errors in this simulation concern the outlier-detection pre-processing step, where it has mistaken data points as outliers in less dense areas.

The last simulation is devoted to a linear mixture $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$ with a random mixing matrix $\mathbf{A}$. The performance of the proposed algorithm is still quite well in this simulation, which brings with it the certainty that one can use this approach even for cases when even the linearity/nonlinearity of



(a) Centered observations

(b) Clustered data (into blue and red classes)

(c) The 1st signal

(d) The 2nd signal

Figure 5.10: Simulation results for a linear mixture $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$ with a random mixing matrix

the mixture is unknown.

Finally it should be recalled that, as it can be seen from the simulation results, in nonlinear BSS, a nonlinear function remains as an ambiguity in source reconstruction and it can not be resolved without further information. This could also be deducted from a mathematical point of view, similar to Section 3.3.

5.4 Discussion and Future Works

In this work, nonlinear BSS approach is proposed for sparse sources. The proposed method is mathematically studied and its performance is approved by simulation results.

5.4.1 Discussion

We believe that the proposed algorithm works even for under-determined cases, where the number of observations is less than the number of source signals. This outstanding capability comes from the fact that the separation of sparse sources is based on separating n 1-dimensional manifolds, and even in a 2-dimensional space we can have infinite number of different 1-dimensional manifolds. In fact, the number of manifolds is limited by the practical resolution and the number of samples. Therefore, the minimum required number of the observation signals, regardless of the number of sources, is always two.

The proposed method could also be useful for cases when the number of source signals is unknown. There could be multiple manifold learning algorithms (see Section 5.2.2) in which the number of clusters is not given in advance. Utilizing such algorithms enables the proposed framework to perform nonlinear BSS when the number of the sources is unknown.

As stated before, the proposed approach separates nonlinearly mixed *spatially* sparse sources. For example, independent block sparse signals which are sparse enough, will make a set of spatially sparse signals. Moreover, the proposed approach works even if each source signal takes a constant value for most of the time (not necessarily zero) and has sparse variations. In this case, based on similar arguments to the discussions in this chapter, and via a similar algorithm to the one proposed in Section 5.2, the signals can be separated.

The proposed method is also applicable for smooth-enough mixtures of compressible signals. A signal is compressible in a domain when its coefficients in that domain observe a power law decay. In other words, given a signal $\mathbf{s}$ expressed as

$$\mathbf{s} = \mathbf{\Psi}\boldsymbol{\alpha} \tag{5.26}$$

where $\mathbf{\Psi}$ is a matrix comprising the orthonormal vectors (which can be considered as the basis of a domain) and $\boldsymbol{\alpha}$ is the vector of the coefficients of $\mathbf{s}$

with respect to $\mathbf{\Psi}$, $\mathbf{s}$ it is compressible if

$$\forall\ i \quad |\alpha_i| \leq Ci^{-q} \tag{5.27}$$

where C and q are constants. The largest possible q is the compressibility index, thus the larger the compressibility index, the faster the coefficients decay. For example, images are compressible in Wavelet domain.

The key idea which let us apply the proposed framework on smooth mixtures of compressible signals is that it performs, as long as the scatter plot of the observations contains the manifolds. In fact, the proposed algorithm does not fundamentally require the "sparsity"; it is only needed so that the observations forms manifolds which bring information about the unknown nonlinear mapping. Consequently, since compressible signals, hence their smooth mixtures, still lie on low-dimensional manifolds and can be modeled as noisy sparse signals, they are expected to be separable by the proposed method.

It should also be emphasized that in this work, the source vector $\mathbf{s}$ is assumed to be enough sparse such that the data mostly lives on 1-dimensional manifolds in the n-dimensional space. However, the proposed approach does not fundamentally require this assumption. In the first step, Section 5.2.1, the clustering algorithm can be modified so as to be able to learn higher dimensional manifolds, as proposed in Appendix B. The generalization of both parametric and non-parametric approaches for these cases is straightforward.

Once the higher-dimensional manifolds are learned, the 1-dimensional ones corresponding to the activity of exactly one of the sources can be reconstructed estimated by looking at their intersections. Particularly, the intersection of $n-1$ $(n-1)$-dimensional manifolds each of which corresponding to the simultaneous activity of $n-1$ sources, comprises a 1-dimensional manifold corresponding to the activity of exactly one source.

In other words, 1-dimensional manifolds corresponding to the activity of exactly one source, are reconstructed by the intersection of a number of higher-dimensional manifolds. As a result, n 1-dimensional manifolds can

be learned by intersecting higher-dimensional ones, thus the second step, Section 5.2.2, can be applied without any modification.

Finally, in the proposed algorithm we assumed that there are enough number of the signals such that the manifold learning algorithm converges. However, the less the signals are sparse, and the less samples we have, the less the performance of the manifold learning will be. But it should be noted that the learned manifolds are intermediate extracted information aiming at clustering the data; i.e. the manifolds, themselves, are not fundamental, it is the clustered data which plays the important role. Therefore, even if the manifold learning is not perfectly done, as long as the classification of the data is well done, its error does not propagate into the separation (e.g. Fig. 5.8).

5.4.2 Future Works

For future works, it will be interesting to develop the proposed framework for the sources that are not sparse in time domain, but in some other domain like frequency domain. In these cases, one has to firstly transform the mixing model to the sparse domain for both sources and observations in order to be able to cluster the observation from the manifolds in the sparse domain and then apply the proposed method.

It should be noted that such generalization is not straightforward. For example, even if a signal is sparse in frequency domain, its nonlinear transformation may generate frequency components that did not exist in the signal, hence make it not sparse anymore. However, for studying such cases, one should consider nonlinearities which have limited effects on the domain of sparsity. For example, smooth nonlinear functions are expected not to distort the frequency domain dramatically, hence interesting to be investigated when mixing signals having few frequency components.

Moreover, it would be useful to apply the proposed approach to practical applications and to utilize the prior information (related to the real case) for reconstruction of the sources. This information may either be related to the

source signals or the mixing model.

Due to the diverse practical applications of Gaussian Processes (GP's), it would also be interesting to study them in the proposed framework. Considering the simplicity of GP modeling and their interesting characteristics, it may also lead to noticeable theoretical results, especially for resolving the problem of separating outliers.

Conclusion and Perspectives

In this work, nonlinear BSS problem is investigated and new results and approaches are proposed. It is shown that nonlinear mixtures, which had been thought not to be generally separable for many years, can be separated assuming the sources to change enough smoothly along time, i.e. having temporal correlation. Two different approaches were proposed which utilize this information for performing the separation.

In the first approach, the global nonlinear mixture of the sources is locally transformed to linear mixture of their velocities (time-derivatives), which is treated via conventional adaptive methods. A nonlinear regression technique is also utilized in order to learn the global nonlinear de-mixing function from the local estimations, which dramatically enhances the performance of the proposed approach.

Since the proposed approach is based on local linear approximation of the nonlinear function, its efficiency evidently depends on both the level of nonlinearity of the mixture and the colorfulness of the sources. Although this relationship is visually illustrated by simulation results, it is not demonstrated by mathematical formulations, which might lead to a theoretical proof of for blind separability of nonlinear mixtures.

The second general approach is based on modeling the signal by means of Gaussian processes. As Gaussian processes attract more attentions in the signal processing domain because of their flexibility and generality in modeling diverse signals, it becomes more and more beneficial and fruitful to consider them exclusively in nonlinear mixtures. Particularly, it is interesting to see whether GPs survive passing through nonlinear mixtures or

not. It is shown that although there are nonlinear mappings which do not manipulate the distribution of the signal (especially, its Gaussianity), being restricted to polynomials, they are limited to be linear. As a consequence, it is sufficient for blind linearization of nonlinear mixtures of GPs, to retrieve the Gaussianity of the signals. Such a linearizing function followed by a traditional linear BSS method results in nonlinear BSS. Since general nonlinear functions can be approximated by polynomials with arbitrary small error (based on Taylor expansion theorem), they are supposed to separable (conditioned to satisfy some assumptions) through this approach as well.

It should be noted that our work, as well as other general nonlinear BSS algorithms, suffers from an ambiguity of a component-wise nonlinear function and a change of orders. This can be understood as the generalization of the well-known permutation and scaling indeterminacy of source reconstruction in linear BSS, to the nonlinear problem. In other words, while the continuity of local linear approximations imposes a global permutation in the nonlinear problem, local scales perform as a nonlinear function globally. These ambiguities can only be resolved employing prior knowledge about either the sources or the mixing model, hence not addressed in this word.

Nonetheless, nonlinear BSS has also been investigated for a particular case where there are further assumptions on the sources: being spatially sparse. Special characteristics of these signals lead to constraints which can be employed for the separation. Even though linearly-mixed sparse sources had already been perused and proved to be separable via effective separation algorithms, their nonlinear mixtures were left unstudied. Like linear mixtures, the observations of nonlinear mixtures of spatially sparse sources mainly lie on nonlinear manifolds whose dimensions, depending on the sparsity of the sources, is less than the dimension of the space. Thus, similar to the geographical approaches for separating linear mixtures, the nonlinear manifolds can be classified and learned to perform the separation.

Future Works

Considering above explanations, future works in nonlinear BSS can continue in the following directions.

1. As mentioned earlier, formulating the level of smoothness of the sources, and quantifying its relation with the correctness of local linear approximations of the nonlinear model might end to a proof for separability of nonlinear mixtures. Indeed, considering the current results and the proposed approach which is capable of separating general nonlinear mixtures, looking for an exact theoretical proof is of huge interest.

2. This work was mainly concentrated on theoretical aspects of nonlinear BSS, thus the generated methods were just verified by simulations on synthetic data. Although the fundamental idea of local linear approximations has been examined on real hyperspectral images and has been shown to perform well, it would be interesting to utilize the introduced approaches on practical applications and realistic data. Moreover, according to the application, additional assumptions and constraints are imposed which might be employed in order to boost the performance of the algorithm.

3. General nonlinear mixtures seem to be too diverse to be processed though a single algorithm. Being focused on specific problem models, inspired from practical applications, let us develop application-oriented separation methods which are supposed to be more impressive. This is why parts of this work were also devoted to spatially sparse sources and Gaussian processes. Therefore, it is certainly suggested for future works to consider particular problem models which happen in real world, in order that the separation algorithm benefits from further characteristics and assumptions. For example, single frequency source signals, or more generally sinusoidal ones, sound advantageous to be investigated, because of both their capability of modeling any arbitrary

signal (based on Fourier expansion) and their specifications passing through mixing systems (which is employed, for example, in DUET[2] algorithm [Jourjine et al., 2000] for linear BSS). Moreover, validating the theoretical derivations, given a parametric model as proposed in Section 3.1.4, through simulations would be an interesting short-term perspective.

Last but not least, suggested future works in regard with the discussed models of chapters 4 and 5, are individually proposed at the last section of the chapter, hence not repeated here.

[2]Degenerate Unmixing Estimation Technique

A Separability of Linear Mixtures of Sparse Sources

Assume that all the sources $s_i(t)$, $1 \leq i \leq n$, are κ_0-sparse, $\kappa_0 \approx (n-1)/n$, and are ergodic. Therefore $n\kappa_0 \approx n-1$, which means that most probably $n-1$ signals will be simultaneously active,i.e. only one source is inactive. In this case, given the linear model of (5.4), the scatter plot of the observations $\mathbf{x}(t) = [x_1(t), x_2(t), \ldots, x_n(t)]^\dagger$ mostly lies on $(n-1)$-dimensional hyper-planes in the n-dimensional space [Rivet et al., 2007].

The following nice theorem can be derived for this case as follows.

Theorem 7. *In linear mixtures of n mutually independent spatially sparse sources $n-1$ of which are most probably simultaneously active, the observation vector makes n hyper-planes of*

$$q_k(\mathbf{x}) = \mathbf{b}_k^\dagger \mathbf{x} = 0 \qquad k = 1, \ldots, n \tag{A.1}$$

each of which corresponds to the case where one of the sources s_k is not active. In this case, the $n \times n$ matrix $\mathbf{B}$ whose rows are $\mathbf{b}_k^\dagger$ for $k = 1, \ldots, n$ separates the sources, i.e. $\mathbf{y} = \mathbf{B}\mathbf{x}$ is the reconstructed source vector up to the order and scaling ambiguities.

Theorem 7 claims that in order to separate the sources in this case, one should look at the scatter plot of the observations and estimate the normal vector of the hyper-planes, then stack them over each other in a matrix to construct the separating matrix.

Proof. Combining (A.1) and (5.4) we have

$$\mathbf{b}_k^\dagger \mathbf{A}\mathbf{s}\Big|_{s_k=0} = \mathbf{c}_k^\dagger \mathbf{s}\Big|_{s_k=0} = 0 \qquad k = 1, \ldots, n \tag{A.2}$$

where $\forall k\ \mathbf{c}_k^\dagger = \mathbf{b}_k^\dagger \mathbf{A}$. Defining the matrix $\mathbf{C}$, whose k^{th} row $k = 1, \ldots, n$ is equal to $\mathbf{c}_k^\dagger$, (A.2) can rewritten as

$$\mathbf{y} = \mathbf{C}\mathbf{s} = \mathbf{B}\mathbf{A}\mathbf{s} \tag{A.3}$$

such that $\forall k$, if $s_k = 0$ then $y_k = 0$. In other words, if $s_k = 0$ then

$$y_k = \sum_{i=1}^{n} c_{ki} s_i \Big|_{s_k=0} = \sum_{\substack{i=1 \\ i \neq k}}^{n} c_{ki} s_i = 0. \tag{A.4}$$

Since the polynomial of (A.4) for all the values of s_i $(i \neq k)$ equals to zero, all the coefficients should be equal to zero which means $\forall k, i\ k \neq i$, $c_{ki} = 0$ and $\mathbf{C} = \mathbf{B}\mathbf{A}$ is a diagonal matrix. □

Theorem 7 also inspires an idea for another proof for the separability of very sparse sources which mostly lie on 1-dimensional subspaces, like (5.7). The proof would be based on the orthogonal complement of the subspaces on which the data lies, but the details are not brought here.

It would also be interesting to study the situation when a smaller number of the sources are simultaneously active, i.e. $n\kappa_0 \approx \eta < n-1$. In this case, the data is located on η-dimensional subspaces in the n-dimensional space, which should be learned. Note that each subspace corresponds to the activity of η sources and inactivity of the others. Therefore, all 1-dimensional subspaces (lines) corresponding to the activity of only one sources, can be constructed by intersecting some of the η-dimensional subspaces. Once these lines are learned, according to Section 5.1.1, sources can be separated.

Besides, another approach might be proposed based on constructing the n $(n-1)$-dimensional hyper-planes corresponding to the activity of $n-1$ sources and the silence on the other one, through unions of the η-dimensional subspaces. These hyper-planes would be subject to theorem 7, thus the sources would be separable.

B Clustering and Multiple Manifold Learning

In this appendix we provide a relatively deep investigation on the problem of manifold clustering followed by proposing robust algorithms, which may also be used separately for other applications in signal processing and pattern recognition. While the problem has already been addressed for linear manifolds [Babaie-Zadeh et al., 2006], the development for nonlinear ones proposed in the current appendix is original. Relative results in the literature for this problem could be found under the name of curvilinear component analysis, e.g. [Demartines and Hérault, 1997].

Let us firstly review the related background of the problem.

B.1 Related Background

Manifold clustering problem can be understood both as a generalization of the regression and curve fitting and a generalization of unsupervised classification. The connection between this problem and the literature is explained in the following.

B.1.1 Single Linear Regression

The n-dimensional linear regression problem consists of a set of n-dimensional data $[y(t), x_1(t), x_2(t), \ldots, x_{n-1}(t)]^\dagger$ for $t = 1, \ldots, T$, where $y(t)$ follows a

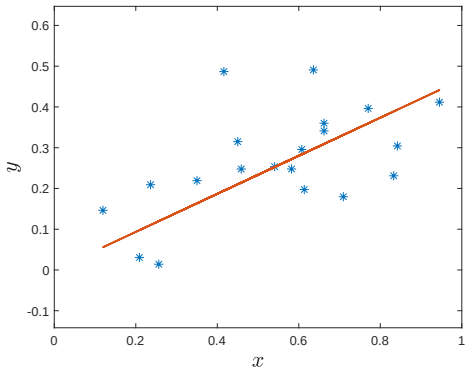


Figure B.1: Linear regression

noisy linear model as

$$\forall\ 1 \leq t \leq T \quad y(t) = \mathbf{a}^{\dagger}\mathbf{x}(t) + c + n(t) \tag{B.1}$$

where $\mathbf{x}(t) = [x_1(t), x_2(t), \ldots, x_{n-1}(t)]^{\dagger}$, $\mathbf{a} \in \mathbb{R}^{n-1}$ and c are a vector and a scalar parameters, respectively, and $n(t)$ is an additive noise. In order to find the parameters of the model, the mean squared error in estimation of the output should be minimized as

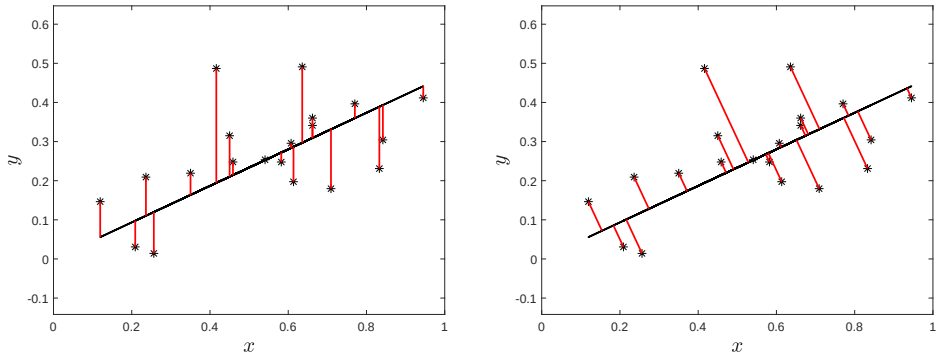
$$\text{minimize } \Big(\sum_{t=1}^{T}(y(t) - \hat{y}(t))^2\Big) = \underset{\mathbf{a},c}{\text{minimize}} \Big(\sum_{t=1}^{T}(y(t) - (\mathbf{a}^{\dagger}\mathbf{x}(t) + c))^2\Big) \tag{B.2}$$

where

$$\forall\ 1 \leq t \leq T \quad \hat{y}(t) = \mathbf{a}^{\dagger}\mathbf{x}(t) + c. \tag{B.3}$$

Fig. B.1 shows the result of a 2-dimensional linear regression. In this figure, each " * " corresponds to a data point and the red line is the result of the regression.

It can be seen from (B.2) that in this case the "vertical distance" [Babaie-Zadeh et al., 2002] of the points and the line is minimized (in mean squared sense). This is due to the assumption that in the regression problem, the scalar y is supposed to be a noisy linear mixture of the other $n-1$ signals. In



(a) Vertical distance (b) Orthogonal distance

Figure B.2: The difference between vertical and orthogonal distances

other word, the goal in this case is to estimate the best model of the scalar output as a linear function of the inputs.

However, in fitting applications, we have an n-dimensional noisy input data $\mathbf{x}(t) = [x_1(t), x_2(t), \ldots, x_n(t)]^\dagger$ for $t = 1, \ldots, T$ that lies on a hyper-plane in the n-dimensional space. It can be modeled as

$$\mathbf{a}^\dagger \mathbf{x}(t) = c + n(t). \tag{B.4}$$

In this case, in order to fit the best hyper-plane to the data, it is necessary to consider the *orthogonal* distance [Babaie-Zadeh et al., 2002] of the points to the hyper-plane (see Fig. B.2).

Therefore, the fitted hyper-plane $\Gamma : \ \mathbf{a}^\dagger \mathbf{x} = c$ is estimated by solving the minimization

$$\text{minimize} \ \Big(\sum_{t=1}^{T} \mathrm{d}^2(\mathbf{x}(t), \Gamma) \Big) = \underset{\mathbf{a},c}{\text{minimize}} \ \Big(\sum_{t=1}^{T} \frac{(\mathbf{a}^\dagger \mathbf{x}(t) - c)^2}{\mathbf{a}^\dagger \mathbf{a}} \Big) \tag{B.5}$$

where $\mathrm{d}(\mathbf{x}(t), \Gamma)$ is the distance from the point $\mathbf{x}(t)$ to Γ which is calculated in the linear model as

$$\mathrm{d}^2(\mathbf{x}(t), \Gamma) = \frac{|\mathbf{a}^\dagger \mathbf{x}(t) - c|^2}{\mathbf{a}^\dagger \mathbf{a}}. \tag{B.6}$$

B.1.2 Dealing with Outliers

Possible outliers in the data dramatically affect the result of the fitting. It is due to the fact that the *squared* distance for the outliers will be much greater than the other point and may become dominant in the summation which is going to be minimized. Thus it is better to use a weighted distance of points to manifolds so that the effect of very far distances are reduced. The weighted distance, similar to (3.61), can be defined as

$$\mathrm{d}_w^2(\mathbf{x}(t), \Gamma) = \mathrm{d}^2(\mathbf{x}(t), \Gamma) \times w(\mathrm{d}^2(\mathbf{x}(t), \Gamma)) \tag{B.7}$$

where $w(\cdot)$ denotes a weighting function.

There are several options of $w(\cdot)$ that can be used according to the data. For example, one may suggest a masking weight which simply ignores the outliers in the learning process. In this case, outliers should be detected via calculating a criterion through a pre-processing step, hence be removed from the data. This is why this method is called *Hard Thresholding*.

For example, outliers are usually much farther from their closest neighbors than the average. In other words, outliers are commonly in much less dense areas of the space. This fact can be employed in order to design the outlier-detecting pre-processing step. Mathematically speaking, $\mathbf{x}(t_i)$ is detected as an outlier if

$$\frac{1}{T}\sum_{t=1}^{T} \|\mathbf{x}(t_i) - \mathbf{x}(t)\|_2^2 \ll \frac{1}{J}\sum_{j=1}^{J} \|\mathbf{x}(t_i) - \mathbf{x}(t_j^i)\|_2^2 \tag{B.8}$$

where $\mathbf{x}(t_j^i)$ for $j = 1, \ldots, J$ are the J closest observation points to $\mathbf{x}(t_i)$.

Consequently, the corresponding weighting function can be defined as

$$w_{HT}(\mathrm{d}^2(\mathbf{x}(t), \Gamma)) = \begin{cases} 0, & \mathbf{x}(t) \text{ is an outlier} \\ 1, & \text{else} \end{cases}. \tag{B.9}$$

The second method of reducing the effect of outliers is based on *Soft Thresholding*. In this approach, instead of completely removing the outliers, a weighted squared distance is used so that the long distance of outliers

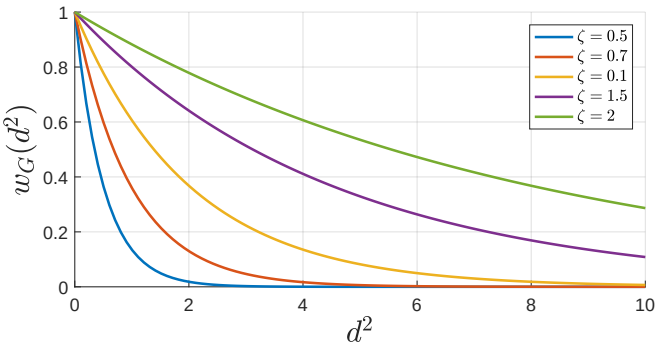


Figure B.3: Gaussian weighting function of (B.10) for different values of ζ

is less weighted and their effect on the manifold is limited. The weighting function is designed such that it is close to 1 for short distances and it tends to zero when the distance gets too large. As an example, Gaussian weighting, similar to (3.63), function is defined as

$$w_G(d^2) = e^{-\frac{d^2}{2\zeta^2}} \tag{B.10}$$

where $w_G(d^2)$ is the Gaussian weight as a function of the squared distance and ζ is a parameter which can be adjusted according to the specifications of the data. The general shape of this weighting function is illustrated in Fig. B.3.

B.1.3 Single Manifold Learning

The generalization of the idea introduced in subsection B.1.1 to nonlinear manifold learning is straightforward. A manifold Γ is best fitted to the data $\mathbf{x}(t)$ for $t = 1, \ldots, T$ if it minimizes

$$\text{minimize} \ \Big(\sum_{t=1}^{T} \mathrm{d}_w^2(\mathbf{x}(t), \Gamma) \Big) \tag{B.11}$$

where $\mathrm{d}_w(\mathbf{x}(t), \Gamma)$ is the weighted Euclidean distance between the point $\mathbf{x}(t)$ and the manifold Γ. The distance is formulated as

$$\mathrm{d}(\mathbf{x}(t), \Gamma) = \min_{\mathbf{p}} \|\mathbf{p} - \mathbf{x}(t)\|_2 \quad \text{s.t.} \ \ \mathbf{p} \in \Gamma, \tag{B.12}$$

which is supposed to be weighted according to subsection B.1.2.

The minimization (B.11) can be solved through either a parametric approach or a non-parametric one.

B.1.3.1 Parametric Approach

In this approach, a parametric model for the manifold Γ is assumed and then the mean squared distance is minimized with respect to those parameters. The manifold Γ is a D-dimensional manifold living in the n-dimensional space, thus it can be formulated as the intersection of $n-D$ $(n-1)$-dimensional manifolds, each of which is determined by equation $q^{(d)}(\mathbf{x};\boldsymbol{\theta}^{(d)}) = 0$ $(1 \leq d \leq n-D)$. Thus the manifold Γ will be formulated as

$$\Gamma: \quad \forall\ 1 \leq d \leq n-D \quad q^{(d)}(\mathbf{x};\boldsymbol{\theta}^{(d)}) = 0 \tag{B.13}$$

where $\boldsymbol{\theta}^{(d)}$ are vectors of the parametric model of Γ.

As an example, a second-order *polynomial* modeling of the manifold can be assumed as (for all $1 \leq d \leq n-D$)

$$\Gamma: \quad q^{(d)}(\mathbf{x};\boldsymbol{\theta}^{(d)}) = \mathbf{x}^{\dagger}\mathbf{A}^{(d)}\mathbf{x} + \mathbf{b}^{(d)^{\dagger}}\mathbf{x} + c^{(d)} = 0 \tag{B.14}$$

where each vector of the parameters $\boldsymbol{\theta}^{(d)}$ includes all the parameters of the $n \times n$ matrix $\mathbf{A}^{(d)}$, the n-dimensional vector $\mathbf{b}^{(d)}$, and the scalar $c^{(d)}$ (there are n^2+n+1 parameters in this model). One may assume any other parametric model depending on either prior information on the mixing model (if it exists) or a general model which is able to model a wide range of nonlinear functions.

As a consequence, (B.11) can be expressed with respect to unknown parameters as

$$\underset{\substack{\boldsymbol{\theta}^{(d)} \\ d=1,\ldots,n-D}}{\text{minimize}} \ \sum_{t=1}^{T} \Big(\mathrm{d}_w^2(\mathbf{x}(t), \Gamma) \Big) \tag{B.15}$$

where $\mathrm{d}_w^2(\mathbf{x}(t), \Gamma)$ can be calculated as a function of the parameters according to (B.12).

B.1.3.2 Non-Parametric Approach

The manifold can also be learned in a non-parametric approach. In this case, one may constructively employ a D-dimensional smoothing method (e.g. smoothing spline) to fit a manifold to the data. Naming the D-dimensional smoothing function $\mathcal{F}_D(\cdot)$, the learned manifold can be expressed as

$$\Gamma = \mathcal{F}_D(\{\mathbf{x}(t)\}) \tag{B.16}$$

where $\{\mathbf{x}(t)\}$ is the set of all data points ($1 \leq t \leq T$). It should be noted that the smoothing criterion of the function $\mathcal{F}_D(\cdot)$ needs to be robust to the outliers.

Now the related background to the manifold clustering problem is briefly reviewed and the corresponding notation is introduced. In the following, the nonlinear manifold clustering problem is defined and then the proposed algorithms are introduced.

B.2 Problem Definition

Given T sample vectors $\mathbf{x}(1), \ldots, \mathbf{x}(T)$ in an n-dimensional space lying on a union of K manifolds $\Gamma_1, \ldots, \Gamma_K$, and assuming that each of them is a noisy sample of its corresponding manifold, we aim at classifying the data according to the manifold they belong to. However, the data points do not exactly lie on the manifolds, they may be noisy, they contain outliers, i.e. the data points which do not fit any manifold, evidently they are not labeled, and each manifold Γ_i for $i = 1, \ldots, K$ has a dimension D_i which is known in advance.

Our problem of interest in nonlinear BSS for spatially sparse sources, is a special case of above problem, where $K = n$, $D_i = 1$ for $i = 1, \ldots, n$ and the amplitude of the additive noise is zero.

It should be emphasized that with a low probability, more than one source may happen to be active simultaneously. Since these observations do

not lie on any of the manifolds of (5.11), they are considered as *outliers* in the clustering step.

The goal is to find the manifolds such that they best fit the data. These manifolds are supposed to minimize the total fitting error of all data points. The fitting error for each data point is calculated as its distance to its corresponding manifold, and the corresponding manifold to each data point is the closest one to it. So the fitting problem can be expressed by minimizing the (weighted) mean squared error of the estimated models as

$$\underset{\substack{\Gamma_i \\ i=1,\dots,K}}{\text{minimize}} \sum_{t=1}^{T} \left(\min_{1\leq i\leq K} \left(\mathrm{d}_w^2(\mathbf{x}(t), \Gamma_i) \right) \right) \tag{B.17}$$

where $\mathrm{d}_w(\mathbf{x}(t), \Gamma_i)$ is the weighted distance from the point $\mathbf{x}(t)$ to the manifold Γ_i. In (B.17), the term between the big parentheses formulates the weighted squared distance of each observation $\mathbf{x}(t)$ to its closest manifold.

The reason why the squared distance in above formulation is *weighted* has been described in details in subsection B.1.2. The weighted distance $\mathrm{d}_w(\mathbf{x}(t), \Gamma_i)$, similar to (B.7), is mathematically defined as

$$\mathrm{d}_w(\mathbf{x}(t), \Gamma_i) = \mathrm{d}(\mathbf{x}(t), \Gamma_i) \times w(\mathrm{d}(\mathbf{x}(t), \Gamma_i)) \tag{B.18}$$

where $w(\cdot)$ is a weighting function and $\mathrm{d}(\mathbf{x}(t), \Gamma_i)$ represents the Euclidean distance of the point $\mathbf{x}(t)$ and the manifold Γ_i. This Euclidean distance, similar to (B.12), can be expressed as

$$\mathrm{d}(\mathbf{x}(t), \Gamma_i) = \min_{\mathbf{p}} \|\mathbf{p} - \mathbf{x}(t)\|_2 \quad \text{s.t. } \mathbf{p} \in \Gamma_i. \tag{B.19}$$

It can also be interpreted as the squared distance of the point $\mathbf{x}(t)$ from the closest point on the manifold Γ_i to it.

Since the data contains outliers, the proposed algorithm needs to be robust enough such that the solution is not influenced too much by them. Please note that although the outliers are supposed to be few, according to the power of 2 in (B.17), they might highly affect the manifold learning process (normally, manifold learning techniques are sensitive to outliers). For

this purpose, it is suggested to use a nonlinear weighting for the distance, in order that it limits the effect of large distances.

Considering (B.19), in order to calculate the distance from each observation to each manifold, generally a minimization over all points of the manifold should be performed. However, considering the structure of manifolds in our BSS problem, it can be more simplified. Returning to (5.11), the distance defined in (B.19) can be rewritten as

$$\mathrm{d}^2(\mathbf{x}(t), \Gamma_i) = \min_{\mathbf{p}} \|\mathbf{p} - \mathbf{x}(t)\|_2^2 \quad \text{s.t. } \forall 1 \leq j \neq i \leq n \quad \mathbf{g}_j(\mathbf{p}) = 0. \tag{B.20}$$

Assuming that the manifold Γ_i, or alternatively $\mathbf{g}_j(\mathbf{x})$ for all $1 \leq j \neq i \leq n$, has continuous first partial derivatives (which is normally true for practical applications), we can use the method of Lagrange multipliers for calculating the distance of (B.20). The Lagrange (Lagrangian) function is defined by

$$\mathcal{L}_i(\mathbf{p}, \lambda) \stackrel{\triangle}{=} \|\mathbf{p} - \mathbf{x}(t)\|_2^2 - \sum_{1 \leq j \neq i \leq n}^{n} \lambda_j \mathbf{g}_j(\mathbf{p}) \tag{B.21}$$

where λ_j's are Lagrange multipliers. Thus, the necessary condition for an optimal solution is given by

$$\nabla_{\mathbf{p},\lambda}\ \mathcal{L}_i(\mathbf{p}^*, \lambda_1^*, \ldots, \lambda_{i-1}^*, \lambda_{i+1}^*, \ldots, \lambda_n^*) = \mathbf{0} \tag{B.22}$$

where $\mathbf{0}$ is a vector whose elements are all equal to zero, ∇ denotes the gradient and $\mathbf{p}^*$ and λ_j^* for all $1 \leq j \neq i \leq n$ are the optimal values of $\mathbf{p}$ and λ_j for all $1 \leq j \neq i \leq n$ respectively.

Expanding (B.22) ends to

$$\begin{cases} \frac{\partial \mathcal{L}_i}{\partial \mathbf{p}} = 2(\mathbf{p}^* - \mathbf{x}(t)) - \sum_{1 \leq j \neq i \leq n}^{n} \lambda_j^* \nabla \mathbf{g}_j(\mathbf{p}^*) = \mathbf{0} \\ \frac{\partial \mathcal{L}_i}{\partial \lambda_j} = \mathbf{g}_j(\mathbf{p}^*) = 0 \quad \forall 1 \leq j \neq i \leq n \end{cases} \tag{B.23}$$

which is a system of $2n-1$ equations and $2n-1$ unknowns (λ_j^* for $1 \leq j \neq i \leq n$ and n elements of $\mathbf{p}^*$). The solutions of this system are candidates for minimizing (B.20). Therefore we have to calculate the distance from the point $\mathbf{x}(t)$ to all the solutions of (B.23) to find the global minimum, which is called as the distance between the point and the manifold.

Nevertheless, the optimization problem (B.17) can be solved through both parametric and non-parametric approaches. These approaches will be described in the following.

B.3 Parametric Approach

Similar to subsection B.1.3, in the parametric approach the manifolds are expressed in a parametric model. Thus (B.17) can be rewritten with respect to the parameters.

The K manifolds Γ_i for $i = 1, \ldots, K$ of D_i dimensions, lying in the n-dimensional space, are formulated as

$$\Gamma_i : \quad \forall\ 1 \leq d_i \leq n - D_i \quad Q_i^{(d_i)}(\mathbf{x}; \boldsymbol{\theta}_i^{(d_i)}) = 0 \tag{B.24}$$

where $\boldsymbol{\theta}_i^{(d_i)}$ is the vector of the parametric model of Γ_i.

It is worth noting again that D_i-dimensional manifolds lying in the n-dimensional space are determined by systems of $n-D_i$ independent equations of $Q_i^{(d_i)}(\mathbf{x}) = 0$ for $d_i = 1, \ldots, n - D_i$.

As a consequence, the problem (B.17) can be expressed with respect to unknown parameters as

$$\underset{\substack{\boldsymbol{\theta}_i^{(d_i)} \\ i=1,\ldots,K \\ d_i=1,\ldots,n-D_i}}{\text{minimize}} \quad \sum_{t=1}^{T} \left(\min_{1 \leq i \leq K} \left(\mathrm{d}_w^2(\mathbf{x}(t), \Gamma_i) \right) \right) \tag{B.25}$$

where $\mathrm{d}_w^2(\mathbf{x}(t), \Gamma_i)$ can be calculated as a function of the parameters.

The value of the cost function which is minimized in (B.25), for the calculated optimal parameters, indicates how well the manifolds are learned. So, especially when there is no prior information about the mixing model, one may try to solve (B.25) many times, each time given a different parametric model, and finally selects the one with the best result which has the minimum value of cost function.

For instance, in our simulations (which are described in more details in Section 5.3), a polynomial model is chosen for clustering the data. Assuming

a first-order polynomial (linear model) to cluster the manifolds, the value of the cost function for the optimal solution found is calculated to see whether the model fits well enough or not. The order of the polynomial model is gradually increased until the value of the cost function based on the learned manifolds is low enough, i.e. less than a predefined threshold.

B.4 Non-Parametric Approach

The idea of this section basically comes from the well-known K-means method for unsupervised classification. K-means comprises two different steps which should be run in an iterative manner. Starting from a random assignment of the data to the classes, the first step is to calculate the centroid (center) of each class and the second one is to update the label according to the latest centroids (each point is labeled as its closest centroid).

Therefore, the non-parametric multiple manifold learning can be proposed as follows.

1. Firstly, data points are randomly assigned to the manifolds. Let us denote the label of each point $\mathbf{x}(t)$ for $t = 1, \ldots, T$ at r^{th} iteration by $\Omega^{(r)}(t) \in \{1, 2, \ldots, K\}$).

2. A manifold is fitted on the points assigned to its class using a non-parametric smoothing approach (B.16) as

$$\forall\ 1 \leq i \leq K \quad \Omega_i^{(r)} = \mathcal{F}_{D_i}(\{\mathbf{x}(t)\} | \Omega^{(r-1)}(t) = i) \tag{B.26}$$

   where the superscript (r) denotes the number of the iteration.

3. The labels of data points are updated regarding their closest manifold as

$$\forall\ 1 \leq t \leq T \quad \Omega^{(r)}(t) = \underset{i}{\operatorname{argmin}} \Big(\mathrm{d}_w^2(\mathbf{x}(t), \Omega_i^{(r)}) \Big). \tag{B.27}$$

The steps 2 and 3 should be iteratively repeated until the algorithm converges.

Algorithm 6 Non-Parametric Multiple Manifold Learning

procedure STEP1: RANDOM INITIALIZATION
  for $t = 1, \ldots, T$ **do**
    $\Omega^{(0)}(t) \leftarrow \text{rand}(1, 2, \ldots, K)$
  end for
  $r \leftarrow 0$
end procedure
repeat
  $r \leftarrow r + 1$
  procedure STEP2: UPDATING MANIFOLDS ($\Omega^r(t);\ t = 1, \ldots, T$)
    for $i = 1, \ldots, K$ **do**
      $\Omega_i^r \leftarrow \mathcal{F}_{D_i}(\{\mathbf{x}(t)\} | \Omega^{(r-1)}(t) = i)$
    end for
  end procedure
  procedure STEP3: UPDATING LABELS ($\Omega_i^r;\ i = 1, \ldots, K$)
    for $t = 1, \ldots, T$ **do**
      $\Omega^{(r)}(t) \leftarrow \underset{i}{\text{argmin}} \left(\text{d}_w^2(\mathbf{x}(t), \Omega_i^{(r)}) \right)$
    end for
  end procedure
until $\Omega^r(t) \neq \Omega^{r-1}(t);\ t = 1, \ldots, T$

Algorithm 6 contains the pseudo-code of the proposed non-parametric multiple manifold learning method.

It can be generally shown that if the weighting function used in (B.27) for calculating $\text{d}_w^2(\mathbf{x}(t), \Omega_i^{(r)})$ is monotonic, the clustering error in this approach may not increase as the algorithm progresses. Therefore, since the number of different possibilities for labeling the observations is finite, the proposed algorithm does converge. However, depending on the initial labeling, it may converges to a *local* minimum instead of the *global* one (this is also a well-known drawback of conventional k-means). Thus, it has to be run several times with different random initialization, and finally the best answer that

has been achieved should be taken.

C Résumé en Francais

C.1 Introduction

Dans un problème de séparation aveugle de source (BSS), on dispose de plusieurs signaux d'observation qui sont des mélanges par une fonction inconnue de plusieurs signaux également inconnus nommés sources. Le but est de reconstituer les sources ayant uniquement accès aux observations, c'est-à-dire sans connaître ni les sources, ni le modèle de mélange.

Le problème BSS est formellement décrit comme suit. À chaque instant t considérons m observations $x_i(t)$, $i = 1, \ldots, m$, qui sont des fonctions inconnues invariantes dans le temps $f_i(\cdot)$ des sources inconnues $s_j(t)$, $j = 1, \ldots, n$. Pour chaque échantillons $t = 1, \ldots, T$, nous pouvons exprimer mathématiquement le modèle comme

$$\mathbf{x}(t) = \mathbf{f}(\mathbf{s}(t)), \qquad t = 1, \ldots, T \tag{C.1}$$

où $\mathbf{x}(t) = [x_1(t), ..., x_m(t)]^\dagger$ († note la transposition de matrice) et $\mathbf{s}(t) = [s_1(t), ..., s_n(t)]^\dagger$ représentent les vecteurs d'observation et source, respectivement, et $\mathbf{f}(\cdot)$ est une fonction de $\mathbb{R}^n$ à $\mathbb{R}^m$. Le modèle associé à ce problème est représenté sur la figure C.1. Dans ce modèle, nous désirons généralement que chacun des éléments de $\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t))$ soit fonction d'un seul des signaux sources (et que chaque signal source apparaisse dans un seul element de $\mathbf{y}(t)$).

La séparation de sources est généralement un problème mal-posé, mais on montre que, dans le cas de mélanges linéaires instantanés, si les sources sont mutuellement indépendantes, elles peuvent être reconstruites à un une

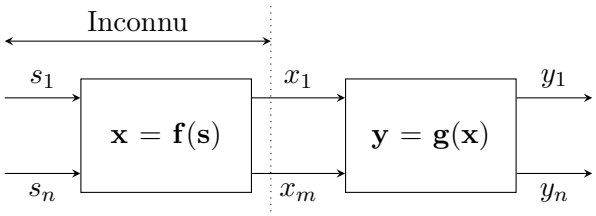


Figure C.1: Modèle de base de problème non-linéaire BSS

permutation et un facteur d'échelle près. Cependant, ce résultat ne peut pas être généralisé au cas de mélanges non-linéaires. En effet, il est montré par des contre-exemples, par exemple [Hosseini and Jutten, 2003, Babaie-Zadeh, 2002], que l'ACI[1], où l'indépendance est mesurée au sens de variables aléatoires, n'est pas capable de séparer les sources dans des mélanges non-linéaires.

Pour cette raison, le problème BSS non-linéaire est presque inexploré dans le cas général. Dans ce travail, de nouvelles approches pour résoudre le BSS non-linéaire sont proposées. Ces approches supposent que les signaux ont une autocorrélation temporelle, c'est-à-dire qu'ils sont colorés, ce qui est une hypothèse réalistes pour la plupart des signaux physiques.

C.2 Une approche générale pour résoudre la BSS non-linéaire

L'approche proposée est principalement basée sur l'utilisation de dérivées de signaux afin d'utiliser l'information temporelle des signaux, comme précédemment introduit [Ehsandoust et al., 2017a]. La relation entre cette approche pour des mélanges non-linéaires et la séparation dans des images hyperspectrales dans le cas de variabilité spectrale, a été établie, et présentée dans [Drumetz et al., 2017].

L'idée principale est basée sur le fait que *les dérivés des sources sont*

[1] Analyse en Composantes Indépendantes

mélangés localement linéairement même si le modèle de mélange est non-linéaire. En effet, si la transformation non-linéaire $\mathbf{f}$ est différentiable en chaque point, on peut en déduire une approximation linéaire locale impliquant les dérivées de sources et d'observations. Ceci s'écrit facilement :

$$x_i(t) = f_i(\mathbf{s}(t)) \quad \Rightarrow \quad \frac{dx_i}{dt} = \sum_{j=1}^{n} \frac{\partial f_i}{\partial s_j} \frac{ds_j}{dt} \tag{C.2}$$

$$\Rightarrow \quad \dot{\mathbf{x}} = \mathbf{J}_{\mathbf{f};t}(\mathbf{s})\dot{\mathbf{s}}, \tag{C.3}$$

où $\mathbf{J}_{\mathbf{f};t}(\mathbf{s})$ est le jacobien de la fonction de mélange $\mathbf{f}$.

En supposant que $\mathbf{J}_{\mathbf{f};t}(\mathbf{x}(t))$ dans (C.3) varie assez lentement pour qu'il reste presque constant dans le voisinage temporel de chaque point $\mathbf{x}(t)$, un algorithme préliminaire (appelé AATVL[2]) a été proposé pour résoudre localement les problèmes BSS linéaires déduits à chaque instants. Le principal problème de cet algorithme est la question de la convergence, qui doit être atteinte à chaque nouvel échantillon d'observations. Ce problème peut être résolu par une technique de régression non-linéaire. En fait, le problème de convergence de l'algorithme AATVL est dû au fait qu'il n'exploite pas l'invariance temporelle et la régularité de la fonction de mélange $\mathbf{f}$. En fait, la non-linéarité $\mathbf{f}$ et son inverse $\mathbf{g}$ étant invariantes dans le temps, la dépendance de $\mathbf{J}_{\mathbf{f};t}$ (respectivement $\mathbf{J}_{\mathbf{g};t}$) sur $\mathbf{s}$ (respectivement $\mathbf{x}$) ne varie pas dans le temps. Par conséquent, une modification de l'algorithme AATVL (appelé BATIN[3]) est proposée en apprenant le modèle non-linéaire de $\mathbf{J}_{\mathbf{g};t}(\mathbf{x})$ à partir de ses estimations à différents échantillons (disons $\hat{\mathbf{J}}_{\mathbf{g}}(\mathbf{x}(t))$ pour $t = 1, \ldots, T$, les sorties de la méthode linéaire adaptative BSS).

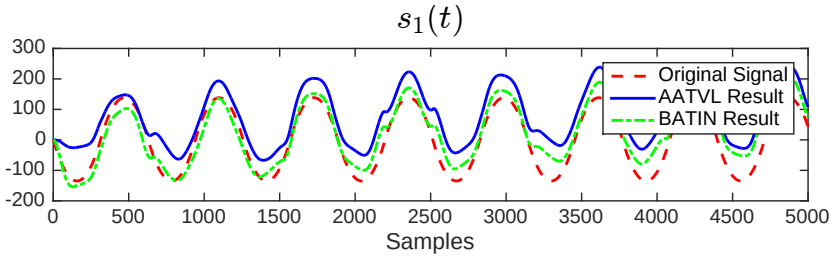
C.2.1 Résultats de la simulation

Considérons le système à deux entrées et à deux sorties de

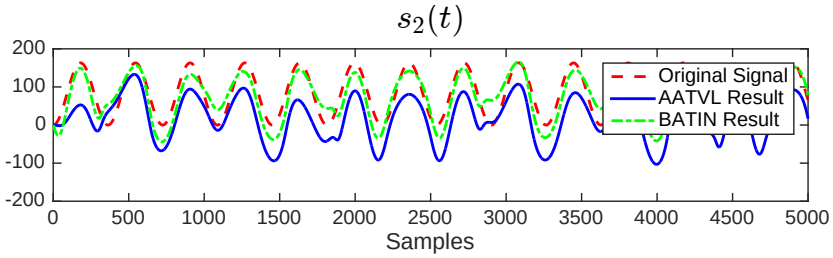
$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \cos\alpha(\mathbf{s}(t)) & -\sin\alpha(\mathbf{s}(t)) \\ \sin\alpha(\mathbf{s}(t)) & \cos\alpha(\mathbf{s}(t)) \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix} \tag{C.4}$$

[2]Algorithme adaptatif pour les mélanges linéaires variant dans le temps

[3]Batch algorithme pour les mélanges non-linéaires invariants dans le temps



(a) La première source



(b) La seconde source

Figure C.2: Résultats des algorithmes AATVL et BATIN pour le mélange (C.4)

où $\alpha(\mathbf{s}(t))$ est défini par le modèle paramétrique

$$\alpha(\mathbf{s}(t)) = \alpha_0 + \gamma \times \sqrt{s_1^2(t) + s_2^2(t)} \tag{C.5}$$

et où α_0 et γ sont quelques paramètres. Tout d'abord, (C.5) est considéré pour $\alpha_0 = 0$ et $\gamma = 1$. Les deux sources qui sont mélangées dans cette simulation sont les intégrales d'un signal sinusoïdal et d'un signal triangulaire.

En appliquant les algorithmes AATVL et BATIN sur les observations, nous obtenons les résultats présentés dans la figure C.2. Comme prévu, BATIN surpasse AATVL dans l'estimation des sources séparées dans les deux simulations. En particulier, le problème de convergence tardive avec AATVL a été presque entièrement résolu par BATIN.

Cependant, dans le cas de mélanges non-linéaires, les sources ne peuvent être reconstruites qu'à une fonction non-linéaire près. Ainsi, l'erreur RMS[4] classique ne peut pas représenter l'erreur de séparation dans le cas non-linéaire. Nous avons donc proposé un nouvel indice de performance pour le BSS non-linéaire qui sera introduit dans la suite.

[4]Root Mean Squared

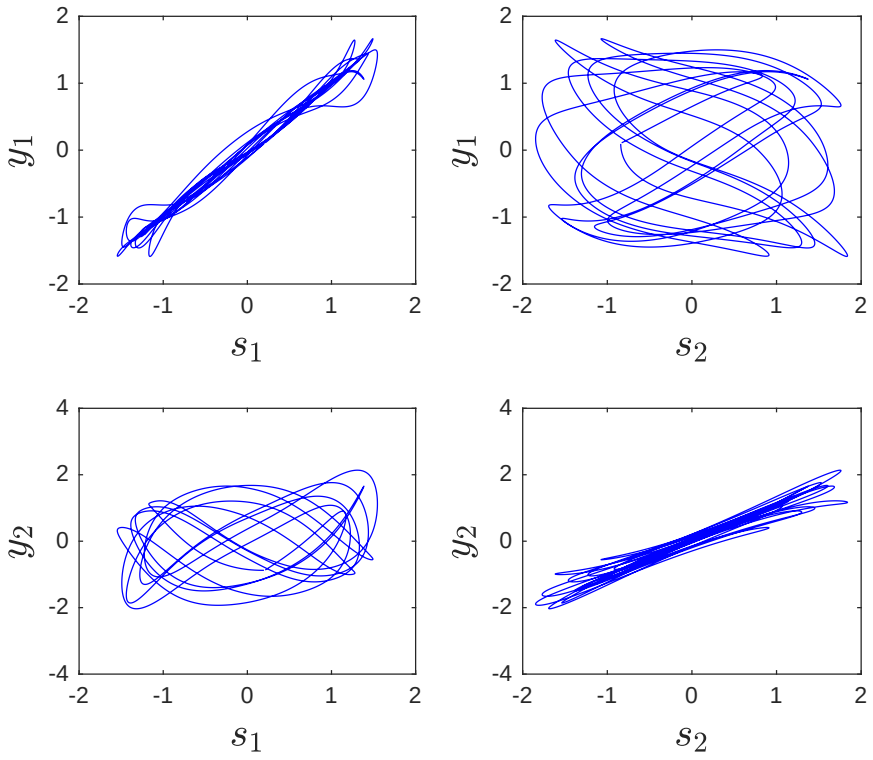


Figure C.3: Les sources estimées $y_1(t)$ et $y_2(t)$ par rapport aux sources théoriques $s_1(t)$ et $s_2(t)$, où l'épaisseur du tracé indique combien la source estimée (axe vertical) dépend de l'autre source

L'*épaisseur* du nuage des points (source estimée, source théorique) indique s'il existe une dépendance à une autre source. En effet, si la séparation est parfaite, la source estimée sera une fonction mathématique de la source théorique, et les points sont localisés sur une courbe unidimensionnelle. Les diagrammes de dispersion de la sortie par rapport aux sources sont illustrés sur la figure C.3.

Nous proposons donc cette mesure de dispersion comme indice général pour l'évaluation des performances des méthodes BSS non-linéaires, et nous la nommerons *Erreur normalisée d'ajustement non-linéaire (N-ENF)*. Les résultats de simulation des algorithmes sont également comparés en termes

Table C.1: Erreur N-ENF pour AATVL et BATIN

	AATVL	**BATIN**
N-ENF pour la **Source 1**	0.0030	0.0019
N-ENF pour la **Source 2**	0.0084	0.0031

d'erreur ENF normalisée et peuvent être trouvés dans la table C.1.

C.3 Linéarisation aveugle des mélanges non-linéaires

Une autre approche est basée sur la modélisation des sources par des processus gaussiens et l'approximation du mélange par un polynôme [Ehsandoust et al., 2017b]. En utilisant ces hypothèses, nous proposerons une nouvelle méthode dont la première étape linéarise ce mélange non-linéaire. Il reste ensuite à résoudre ce mélange résiduel linéaire par un algorithme BSS*linéaire*.

Nous avons prouvé que les mélanges*polynômiaux* perdent la propriété de Gaussianité, sauf si ces mélanges se réduisent à une transformation affine linéaire, ce qui est mathématiquement énoncé dans le théorème suivant.

Théorème 1. *Soient n sources $s_1, \cdots, s_n$ de distribution conjointe normal et mélangées via un transformation polynômiale inversible $\mathbf{p}: \mathbb{R}^n \to \mathbb{R}^n$ fournissant n signaux $y_1, \ldots, y_n$. Si les signaux $y_1, \ldots, y_n$ suivent aussi une distribution gaussienne, le polynôme $\mathbf{p}$ est limité à une transformation affine :*

$$\mathbf{y} = \mathbf{p}(\mathbf{s}) = \mathbf{A}\mathbf{s} + \mathbf{b} \tag{C.6}$$

où $\mathbf{A}$ et $\mathbf{b}$ sont respectivement une matrice $n \times n$ et un vecteur $n \times 1$ de constantes.

Corollaire 1. *En supposant que $\mathbf{f}: \mathbb{R}^n \to \mathbb{R}^n$ est un polynôme inversible, et que les sources sont des processus gaussiens, si nous trouvons un polynôme*

$\mathbf{g}(\mathbf{x})$ *tel que les sorties* $y_1(t), y_2(t), \ldots, y_n(t)$ *sont des processus gaussiens, la fonction totale* $\mathbf{h} = \mathbf{g} \circ \mathbf{f}$ *sera un mélange affine des sources, c'est-à-dire de la forme* $\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) = \mathbf{h}(\mathbf{s}(t)) = \mathbf{A}\mathbf{s}(t) + \mathbf{b}$.

Par conséquent, pour linéariser le mélange il est nécessaire et suffisant d'estimer un polynôme $\mathbf{g}$ tel que $\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t))$ soit un vecteur à distribution gaussienne. On peut donc proposer un algorithme, dont la fonction de coût est une mesure de "non-gaussianité" qui est minimisée par rapport au polynôme $\mathbf{g}$.

Dans ce travail, la néguentropie [Comon, 1994, Hyvärinen, 1999b] est choisie comme mesure de la gaussianité, parce qu'elle est toujours non-négative et invariante par toute transformation linéaire inversible, et s'annule si le signal est gaussien. En supposant un modèle paramétrique pour le polynôme inverse $\mathbf{g}$, l'optimisation est faite par rapport aux paramètres de notre modèle comme

$$\mathbf{g}(\mathbf{x}) = \mathbf{\Theta}\mathbf{k}(\mathbf{x}) \tag{C.7}$$

où $\mathbf{\Theta}$ est la matrice des coefficients et $\mathbf{k}(\mathbf{x}) \in \mathbb{R}^{P\times 1}$ est le vecteur colonne contenant les monômes.

C.3.1 Résultats de la simulation

Le théorème proposé est illustré par un simple exemple simulé 2-par-2 comme suit. Les deux sources s_1 et s_2 sont aléatoirement choisies comme $\mathcal{N}(0,1)$ et sont mélangées par un polynôme bidimensionnel comme

$$\begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} s_1 + (s_1 + s_2)^3 \\ s_2 - (s_1 + s_2)^3 \end{bmatrix}. \tag{C.8}$$

Dans cette expérience, nous avons choisi un modèle cubique pour le mélange polynomial de deux signaux x_1 et x_2 (donc avec 10 paramètres), nous recherchons les paramètres ${\boldsymbol{\theta_1}}^\dagger = [\theta_{10}, \ldots, \theta_{19}]$ dans

$$\begin{aligned} y_1 = \theta_{10}x_1^3 + \theta_{11}x_1^2x_2 + \theta_{12}x_1^2 + \theta_{13}x_1x_2^2 + \theta_{14}x_1x_2 \\ + \theta_{15}x_1 + \theta_{16}x_2^3 + \theta_{17}x_2^2 + \theta_{18}x_2 + \theta_{19} \end{aligned} \tag{C.9}$$

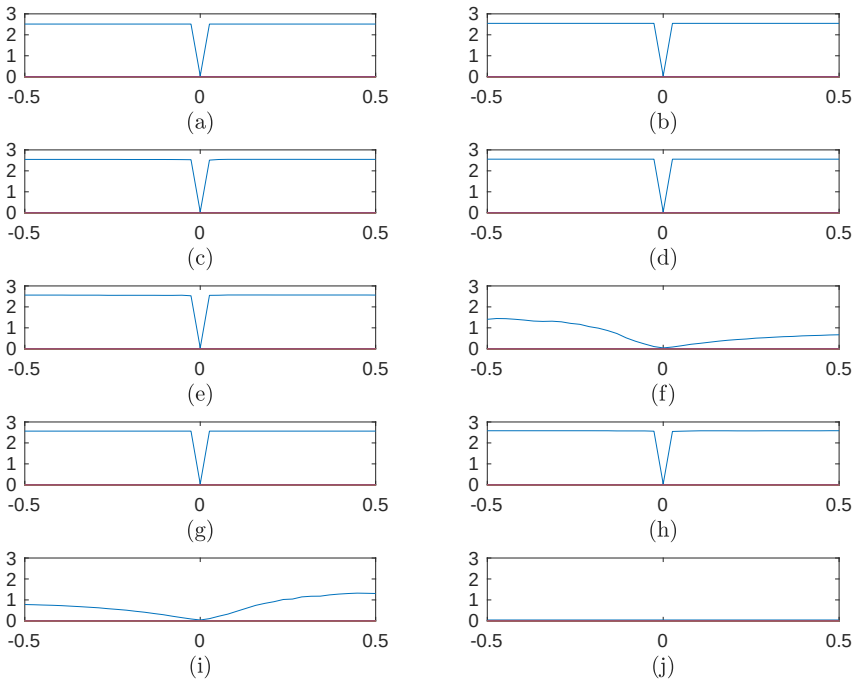
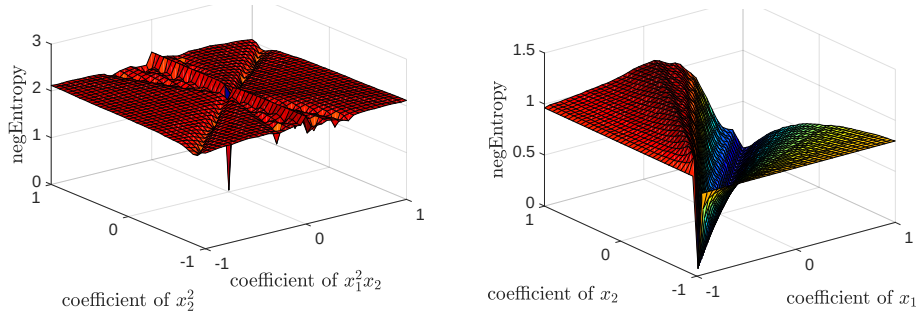


Figure C.4: Néguentropie de y_1 dans (C.9) par rapport aux entrées de $\boldsymbol{\theta_1}$ centrées autour de leur valeur optimale $[0,0,0,0,0,1,0,0,1,0]$ (de θ_{10} à θ_{19} dans les figures (a) à (j) respectivement). Tracé par rapport à chaque entrée, les autres paramètres sont maintenus constants.

tel que y_1 suive une distribution gaussienne. Les résultats de simulation valident la méthode proposée en montrant comment la fonction de coût (ici, la nég-entropie) se comporte autour d'un minimum global, par exemple $\boldsymbol{\theta_1} = [0,0,0,0,0,1,0,0,1,0]$, ce qui donne $y_1 = x_1 + x_2 = s_1 + s_2$.

La figure C.4 illustre la variation de la néguentropie par rapport à l'une des entrées de $\boldsymbol{\theta_1}$ autour de sa valeur optimale $[0,0,0,0,0,1,0,0,1,0]$. Il est à noter que changer θ_{19} n'affecte pas la linéarité du mélange y_1 par rapport à s_1 et s_2, donc ne change pas la nég-entropie.

De plus, la valeur de la néguentropie tout en changeant simultanément θ_{11} et θ_{17} autour de zéro est représentée sur la figure C.5a. Comme on peut



(a) Les coefficients de $x_1^2x_2$ et x_2^2 (b) Les coefficients de x_1 et x_2

Figure C.5: La valeur de la néguentropie de y_1 dans (C.9) par rapport à 2 coefficients du modèle paramétrique, tandis que les autres paramètres sont maintenus constants et égaux à leur valeur optimale en $[0, 0, 0, 0, 0, 1, 0, 0, 1, 0]$.

le voir sur cette figure, bien que le minimum global soit dans l'origine, il y a beaucoup d'autres minima locaux qui peuvent piéger l'algorithme de minimisation. La fig. C.5b montre aussi que la valeur de la néguentropie est minimisée par rapport aux coefficients de x_1 et x_2 (sans changer les autres paramètres) tant que $\theta_{15} = \theta_{18}$.

C.4 Mélanges non-linéaires de sources parcimonieux

Dans cette section [Ehsandoust et al., 2016], nous avons étudié un cas particulier du problème BSS non-linéaire où les sources sont supposées spatialement parcimonieuses, c'est-à-dire qu'elles prennent rarement des valeurs non-nulles en même temps. Voyons d'abord ce qu'il advient des observations données spatialement parcimonieux.

La Fig. C.6 montre les observations pour un système de mélange non-linéaire 2×2 de

$$x_1(t) = e^{s_1(t)} - e^{s_2(t)} \tag{C.10}$$

$$x_2(t) = e^{-s_1(t)} + e^{-s_2(t)} \tag{C.11}$$

où les observations $x_1(t)$ et $x_2(t)$ sont centrées avant d'être tracées. Comme on peut le voir sur les figures, lorsque les sources sont parcimonieuces (Fig. C.6c), rarement plus d'une d'entre sont simultanément actives, donc le nuage de points des observations (Fig. C.6d) contient des variétés dont chacune est le résultat de la transformation de l'un des axes dans l'espace de source.

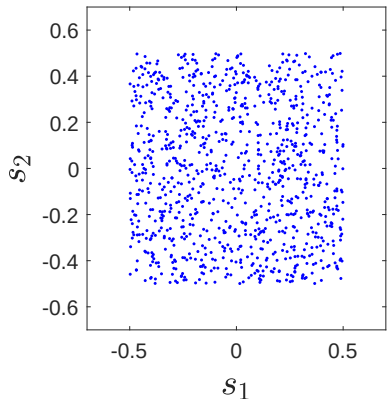
Par conséquent, nous proposons un algorithme en deux étapes: (1) regrouper les observations et l'apprentissage des clusters, et (2) séparer les sources. Dans la première étape, n 1-dimension clusters dans l'espace d'observation sont appris et les données sont regroupées de sorte que chaque classe corresponde à l'activité de l'une des sources. Pour ce faire, les variétés sont apprises sur la base d'une méthode itérative similaire aux k-means bien connus [MacQueen, 1967]. Notre méthode comprend trois étapes comme suit.

1. Initialement, les points de données sont assignés au hasard aux clusters.
2. Un cluster à 1 dimension est ajusté sur les points assignés à chaque classe en utilisant des splines pour le lissage.
3. Les étiquettes des points de données sont associées au cluster le plus proche.

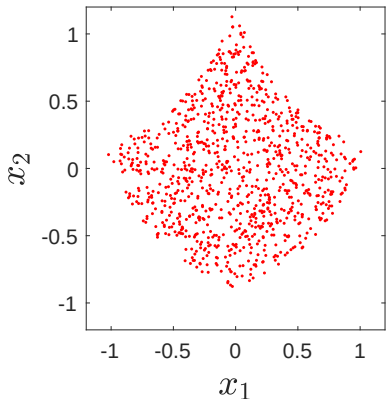
Les étapes de séquencement 2 et 3 doivent être répétées itérativement jusqu'à ce que l'algorithme converge et que les étiquettes ne changent plus.

L'idée proposée peut être facilement illustrée visuellement en représentant les sorties de chaque étape dans chaque itération pour un vecteur d'observation bidimensionnel synthétique dans la séquence de figures C.7a à C.7l. Dans ces figures, deux clusters unidimensionnels dans un espace 2-D doivent être groupés et appris.

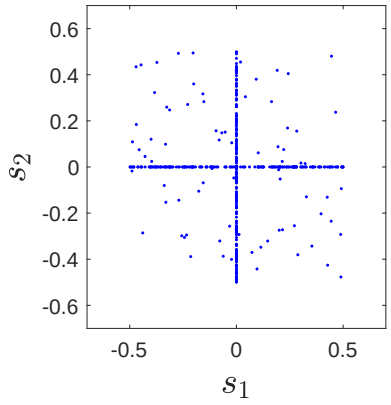
Une fois les clusters appris, les sources sont séparées. En fait, à tout instant $t = 1, \ldots, T$, si $\mathbf{x}(t)$ appartient àu cluster Γ_i, alors chaque source est reconstruite par une fonction arbitraire non-linéaire de l'un des clusters. Le choix des fonctions non-linéaires dépend de l'application et de la nature du mélange, mais une bonne option peut être basée sur un algorithme de réduction de dimension non-linéaire (par exemple ISOMAP [Tenenbaum et al.,



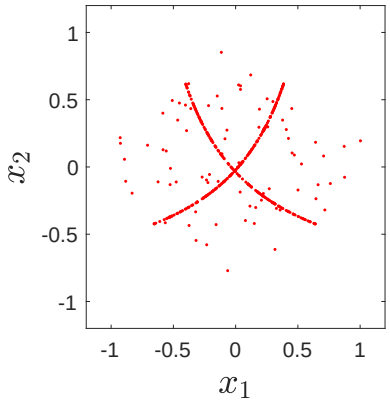
(a) Sources uniformément réparties



(b) Observations centrées de (C.10) et (C.11) pour des sources uniformément distribuées

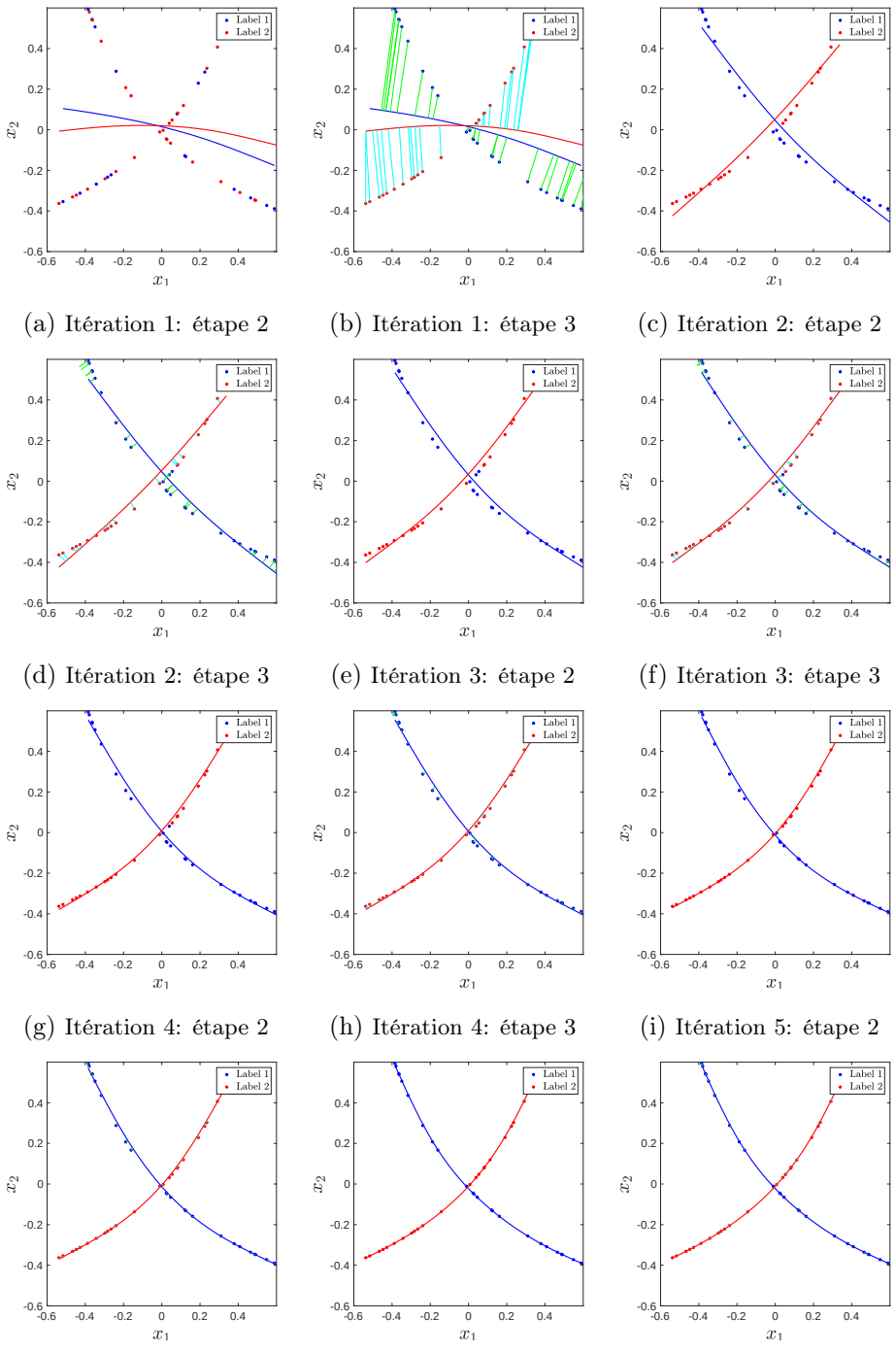


(c) Sources parcimonieuses



(d) Observations centrées de (C.10) et (C.11) pour des sources parcimonieuses

Figure C.6: Comparaison des diagrammes de dispersion des vecteurs de source et d'observation du mélange non-linéaire (C.10) et (C.11), si les sources sont parcimonieuses ou non



(a) Itération 1: étape 2 (b) Itération 1: étape 3 (c) Itération 2: étape 2

(d) Itération 2: étape 3 (e) Itération 3: étape 2 (f) Itération 3: étape 3

(g) Itération 4: étape 2 (h) Itération 4: étape 3 (i) Itération 5: étape 2

(j) Itération 5: étape 3 (k) Itération 6: étape 2 (l) Itération 6: étape 3

Figure C.7: Illustration de l'approche non-paramétrique proposée pour l'apprentissage de 2 clusters dans un espace bidimensionnel; en chiffres correspondant à l'étape 3, la distance *minimum* de chaque point aux clusters est tracée.

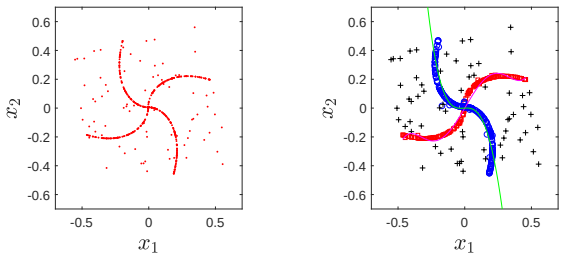
2000] et les cartes de diffusion [Talmon et al., 2013]), qui est supposé transformer des collecteurs unidimensionnels en lignes directes. Pour les valeurs aberrantes, c'est-à-dire aux rares instants où plusieurs sources sont actives, une méthode de projection non-linéaire nous permet d'estimer les valeurs de sources correspondantes.

C.4.1 Résultats de la simulation

Un résultat de simulation est fourni sur la figure C.8. Dans cette figure, (a) contient le nuage de points des observations. Ensuite, dans la partie (b), en plus du diagramme de dispersion d'observation, les deux clusters appris sont également représentés en vert et en violet. De plus, les valeurs aberrantes sont représentées par des croix noires, et les points de données correspondant àu collecteur vert (respectivement, violet) sont représentés en bleu (respectivement, en rouge), d'où la classification est apparente. Les parties (c) et (d) de la figure contiennent les signaux séparés par rapport aux sources originales. Il convient de mentionner que dans cette simulation, les sources parcimonieuses sont constituées de 1000 échantillons (avec un taux d'activité de 25%) uniformément réparties en $[-0.5, 0.5]$ lorsqu'elles sont actives.

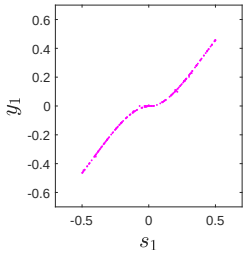
C.5 Conclusion et perspectives

Dans ce travail, le problème se séparation de sources dans le cas non-linéaire est étudié et de nouvelles approches générales sont proposées. Il a été montré que les mélanges non-linéaires, que l'on pensait ne pas être généralement séparables pendant de nombreuses années, peuvent être séparés en supposant que les sources sont suffisamment lisses au cours du temps, c'est-à-dire qu'elles ont une autocorrélation temporelle. Deux approches différentes, exploitant cette information pour réaliser la séparation, ont été proposées. La première méthode repose sur une approximation locale du mélange linéaire, la seconde exploite l'hypothèse que les sources sont des processus gaussiens.

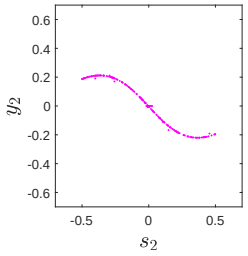


(a) Observations centrées

(b) Données groupées (en classes bleue et rouge)



(c) Le premier signal



(d) Le second signal

Figure C.8: Résultats de la simulation pour $x_1(t) = \cos(\alpha(t))s_1(t) - \sin(\alpha(t))s_2(t)$ et $x_2(t) = \sin(\alpha(t))s_1(t) + \cos(\alpha(t))s_2(t)$ où $\alpha(t) = \frac{\pi}{2}(1 - \sqrt{s_1^2(t) + s_2^2(t)})^2$

De plus, le BSS non-linéaire a également été étudié pour le cas particulier de sources spatialement parcimonieuses, pour lequel une approche géométrique de la séparation a été proposée.

En considérant les résultats ci-dessus, nous suggéronts les travaux futurs dans le BSS non-linéaire dans les directions suivantes.

1. La formulation du niveau de régularité des sources et la quantification de sa relation avec l'exactitude des approximations linéaires locales du modèle non-linéaire pourraient aboutir à une preuve de séparabilité des mélanges non-linéaires. En effet, compte tenu des résultats actuels et de l'approche proposée qui permet de séparer les mélanges non-linéaires généraux, la recherche d'une preuve théorique précise est d'un grand intérêt.

2. Ce travail était principalement concentré sur les aspects théoriques du BSS non-linéaire, donc les méthodes générées ont été vérifiées par des

simulations sur des données synthétiques. Bien que l'idée fondamentale des approximations linéaires locales ait été examinée sur de vraies images hyperspectrales et se soit révélée performante, il serait intéressant d'utiliser les approches introduites sur des applications pratiques et des données réalistes.

Bibliography

[Achard and Jutten, 2005] Achard, S. and Jutten, C. (2005). Identifiability of post-nonlinear mixtures. *IEEE Signal Processing Letters*, 12(5):423–426.

[Altmann et al., 2014] Altmann, Y., Dobigeon, N., and Tourneret, J. Y. (2014). Unsupervised post-nonlinear unmixing of hyperspectral images using a hamiltonian monte carlo algorithm. *IEEE Transactions on Image Processing*, 23(6):2663–2675.

[Altmann et al., 2012] Altmann, Y., Halimi, A., Dobigeon, N., and Tourneret, J. Y. (2012). Supervised nonlinear spectral unmixing using a postnonlinear mixing model for hyperspectral imagery. *IEEE Transactions on Image Processing*, 21(6):3017–3025.

[Babaie-Zadeh, 2002] Babaie-Zadeh, M. (2002). *On blind source separation in convolutive and nonlinear mixtures*. PhD thesis, Grenoble, INPG.

[Babaie-Zadeh et al., 2006] Babaie-Zadeh, M., Jutten, C., and Mansour, A. (2006). Sparse ICA via cluster-wise PCA. *Neurocomputing*, 69(13):1458–1466.

[Babaie-Zadeh et al., 2002] Babaie-Zadeh, M., Jutten, C., and Nayebi, K. (2002). A geometric approach for separating post non-linear mixtures. In *Signal Processing Conference, 2002 11th European*, pages 1–4. IEEE.

[Bach and Jordan, 2002] Bach, F. R. and Jordan, M. I. (2002). Kernel independent component analysis. *Journal of machine learning research*, 3(Jul):1–48.

[Baringhaus et al., 1988] Baringhaus, L., Henze, N., and Morgenstern, D. (1988). Some elementary proofs of the normality of $XY/(X^2 + Y^2)^{1/2}$ when X and Y are normal. *Computers & Mathematics with Applications*, 15(11):943–944.

[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.

[Blaschke et al., 2007] Blaschke, T., Zito, T., and Wiskott, L. (2007). Independent slow feature analysis and nonlinear blind source separation. *Neural computation*, 19(4):994–1021.

[Bofill and Zibulevsky, 2001] Bofill, P. and Zibulevsky, M. (2001). Underdetermined blind source separation using sparse representations. *Signal Processing*, 81:2353–2362.

[Buchner et al., 2003] Buchner, H., Aichner, R., and Kellermann, W. (2003). Blind source separation for convolutive mixtures exploiting nongaussianity, nonwhiteness, and nonstationarity. In *Conf. Rec. of the Seventh International Workshop on Acoustic Echo and Noise Control (IWAENC 03)*. Citeseer.

[Cardoso, 1998] Cardoso, J.-F. (1998). Blind signal separation: statistical principles. *Proceedings IEEE*, 9:2009–2025.

[Cardoso and Laheld, 1996] Cardoso, J. F. and Laheld, B. H. (1996). Equivariant adaptive source separation. *IEEE Transactions on Signal Processing*, 44(12):3017–3030.

[Cardoso and Souloumiac, 1993] Cardoso, J. F. and Souloumiac, A. (1993). An efficient technique for the blind separation of complex sources. In *Proceeding of IEEE Signal Processing Workshop on Higher-Order Statistics, Lake Tahoe*, pages 275–279.

[Cichocki et al., 2006] Cichocki, A., Zdunek, R., and Amari, S.-i. (2006). New algorithms for non-negative matrix factorization in applications to blind source separation. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 5, pages V–V. IEEE.

[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. (2010). *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press.

[De Boor, 1978] De Boor, C. (1978). *A practical guide to splines*, volume 27. Springer-Verlag New York.

[De Lathauwer et al., 1995] De Lathauwer, L., Callaerts, D., De Moor, B., and Vandewalle, J. (1995). Fetal electrocardiogram extraction by blind source subspace separation. In *IEEE Signal Processing Athos workshop on High-order statistics (HOS)*, pages 134–138, Begur, Spain.

[De Lathauwer et al., 2000] De Lathauwer, L., De Moor, B., and Vandewalle, J. (2000). On the best rank-1 and rank-$(r_1, r_2, \ldots, r_n)$ approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1324–1342.

[Demartines and Hérault, 1997] Demartines, P. and Hérault, J. (1997). Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on neural networks*, 8(1):148–154.

[Deville and Duarte, 2015] Deville, Y. and Duarte, L. T. (2015). An overview of blind source separation methods for linear-quadratic and post-nonlinear mixtures. In *International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, pages 155–167. Springer.

[Deville and Hosseini, 2007] Deville, Y. and Hosseini, S. (2007). Blind identification and separation methods for linear-quadratic mixtures and/or linearly independent non-stationary signals. In *Signal Processing and Its Applications, 2007. ISSPA 2007. 9th International Symposium on*, pages 1–4. IEEE.

[Dobigeon et al., 2014] Dobigeon, N., Tourneret, J.-Y., Richard, C., Bermudez, J. C. M., McLaughlin, S., and Hero, A. O. (2014). Nonlinear unmixing of hyperspectral images: Models and algorithms. *IEEE Signal Processing Magazine*, 31(1):82–94.

[Dogancay, 2005] Dogancay, K. (2005). Blind compensation of nonlinear distortion for bandlimited signals. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 52(9):1872–1882.

[Drumetz et al., 2017] Drumetz, L., Ehsandoust, B., Chanussot, J., Rivet, B., Babaie-Zadeh, M., and Jutten, C. (2017). Relationships between nonlinear and space-variant linear models in hyperspectral image unmixing. *IEEE Signal Processing Letters*, PP(99):1–1.

[Drumetz et al., 2016] Drumetz, L., Veganzones, M. A., Henrot, S., Phlypo, R., Chanussot, J., and Jutten, C. (2016). Blind hyperspectral unmixing using an extended linear mixing model to address spectral variability. *IEEE Transactions on Image Processing*, 25(8):3890–3905.

[Duarte and Jutten, 2014] Duarte, L. T. and Jutten, C. (2014). Design of smart ion-selective electrode arrays based on source separation through nonlinear independent component analysis. *Oil & Gas Science and Technology–Revue d'IFP Energies nouvelles*, 69(2):293–306.

[Duarte et al., 2009] Duarte, L. T., Jutten, C., and Moussaoui, S. (2009). A bayesian nonlinear source separation method for smart ion-selective electrode arrays. *IEEE Sensors Journal*, 9(12):1763–1771.

[Duarte et al., 2015] Duarte, L. T., Suyama, R., Attux, R., Romano, J. M. T., and Jutten, C. (2015). A sparsity-based method for blind compensation of a memoryless nonlinear distortion: Application to ion-selective electrodes. *IEEE Sensors Journal*, 15(4):2054–2061.

[Ehsandoust et al., 2015] Ehsandoust, B., Babaie-Zadeh, M., and Jutten, C. (2015). Blind source separation in nonlinear mixture for colored sources using signal derivatives. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 193–200. Springer.

[Ehsandoust et al., 2017a] Ehsandoust, B., Babaie-Zadeh, M., Rivet, B., and Jutten, C. (2017a). Blind source separation in nonlinear mixtures: Separability and a basic algorithm. *IEEE Transactions on Signal Processing*, 65(16):4339–4352.

[Ehsandoust et al., 2017b] Ehsandoust, B., Rivet, B., Babaie-Zadeh, M., and Jutten, C. (2017b). Blind compensation of polynomial mixtures of Gaussian signals with application in nonlinear blind source separation. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 4681–4685. IEEE.

[Ehsandoust et al., 2016] Ehsandoust, B., Rivet, B., Jutten, C., and Babaie-Zadeh, M. (2016). Nonlinear blind source separation for sparse sources. In *2016 24th European Signal Processing Conference (EUSIPCO)*, pages 1583–1587.

[Eisenberg and Sullivan, 2008] Eisenberg, B. and Sullivan, R. (2008). Why is the sum of independent normal random variables normal? *Mathematics Magazine*, 81(5):362–366.

[Fantinato et al., 2017] Fantinato, D. G., Duarte, L. T., Rivet, B., Ehsandoust, B., Attux, R., and Jutten, C. (2017). Gaussian processes for source separation in overdetermined bilinear mixtures. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 300–309. Springer.

[Févotte and Dobigeon, 2015] Févotte, C. and Dobigeon, N. (2015). Nonlinear hyperspectral unmixing with robust nonnegative matrix factorization. *IEEE Transactions on Image Processing*, 24(12):4810–4819.

[Girolami and Fyfe, 1996] Girolami, M. and Fyfe, C. (1996). Negentropy and kurtosis as projection pursuit indices provide generalised ICA algorithms. *Advances in Neural Information Processing Systems*, 9.

[Golbabaee et al., 2013] Golbabaee, M., Arberet, S., and Vandergheynst, P. (2013). Compressive source separation: Theory and methods for hyperspectral imaging. *IEEE Transactions on Image Processing*, 22(12):5096–5110.

[Gribonval and Lesage, 2006] Gribonval, R. and Lesage, S. (2006). A survey of sparse component analysis for blind source separation: principles, perspectives, and new challenges. In *ESANN'06 proceedings-14th European Symposium on Artificial Neural Networks*, pages 323–330. d-side publi.

[Halimi et al., 2011] Halimi, A., Altmann, Y., Dobigeon, N., and Tourneret, J. Y. (2011). Nonlinear unmixing of hyperspectral images using a generalized bilinear model. *IEEE Transactions on Geoscience and Remote Sensing*, 49(11):4153–4162.

[Halimi et al., 2015] Halimi, A., Dobigeon, N., and Tourneret, J. Y. (2015). Unsupervised unmixing of hyperspectral images accounting for endmem-

ber variability. *IEEE Transactions on Image Processing*, 24(12):4904–4917.

[Hamedani and Volkmer, 2001] Hamedani, G. and Volkmer, H. (2001). Certain characterizations of normal distribution via transformations. *Journal of Multivariate Analysis*, 77(2):286 – 294.

[Henrot et al., 2016] Henrot, S., Chanussot, J., and Jutten, C. (2016). Dynamical spectral unmixing of multitemporal hyperspectral images. *IEEE Transactions on Image Processing*, 25(7):3219–3232.

[Hérault and Jutten, 1986] Hérault, J. and Jutten, C. (1986). Space or time adaptive signal processing by neural network models. In *AIP Conference Proceedings 151 on Neural Networks for Computing*, pages 206–211, Woodbury, NY, USA. American Institute of Physics Inc.

[Heylen et al., 2014] Heylen, R., Parente, M., and Gader, P. (2014). A review of nonlinear hyperspectral unmixing methods. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6):1844–1868.

[Hosseini and Jutten, 2003] Hosseini, S. and Jutten, C. (2003). On the separability of nonlinear mixtures of temporally correlated sources. *IEEE Signal Processing Letters*, 10(2):43–46.

[Hwang, 1988] Hwang, C. (1988). Simulated annealing: Theory and applications. *Acta Applicandae Mathematicae*, 12(1):108–111.

[Hyvärinen, 1999a] Hyvärinen, A. (1999a). Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634.

[Hyvärinen, 1999b] Hyvärinen, A. (1999b). Survey on independent component analysis. *Neural computing surveys*, 2(4):94–128.

[Hyvärinen et al., 2004] Hyvärinen, A., Karhunen, J., and Oja, E. (2004). *Independent component analysis*, volume 46. John Wiley & Sons.

[Hyvärinen and Pajunen, 1999] Hyvärinen, A. and Pajunen, P. (1999). Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439.

[Jourjine et al., 2000] Jourjine, A., Rickard, S., and Yilmaz, O. (2000). Blind separation of disjoint orthogonal signals: Demixing N sources from 2 mixtures. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, volume 5, pages 2985–2988. IEEE.

[Jutten and Karhunen, 2003] Jutten, C. and Karhunen, J. (2003). Advances in nonlinear blind source separation. In *Proceeding of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003)*, pages 245–256.

[Jutten and Karhunen, 2004] Jutten, C. and Karhunen, J. (2004). Advances in blind source separation (BSS) and independent component analysis (ICA) for nonlinear mixtures. *International Journal of Neural Systems*, 14(05):267–292.

[Kagan et al., 1973] Kagan, A. M., Linnik, Y. V., and Rao, C. R. (1973). Extension of darmois-skitcvic theorem to functions of random variables satisfying an addition theorem. *Communications in Statistics-Theory and Methods*, 1(5):471–474.

[Kennedy, 2011] Kennedy, J. (2011). Particle swarm optimization. In *Encyclopedia of machine learning*, pages 760–766. Springer.

[Kiviluoto and Oja, 1998] Kiviluoto, K. and Oja, E. (1998). Independent component analysis for parallel financial time series. In *ICONIP*, volume 2, pages 895–898.

[Larue et al., 2004] Larue, A., Jutten, C., and Hosseini, S. (2004). Markovian source separation in post-nonlinear mixtures. In *ICA*, pages 702–709. Springer.

[Lee et al., 2007] Lee, I., Kim, T., and Lee, T.-W. (2007). Fast fixed-point independent vector analysis algorithms for convolutive blind source separation. *Signal Processing*, 87(8):1859–1871.

[Levin, 2010] Levin, D. N. (2010). Performing nonlinear blind source separation with signal invariants. *IEEE Transactions on Signal Processing*, 58(4):2131–2140.

[Levin, 2017] Levin, D. N. (2017). Model-independent method of nonlinear blind source separation. In *Latent Variable Analysis and Signal Separation - 13th International Conference, LVA/ICA 2017, Grenoble, France, February 21-23, 2017, Proceedings*, pages 310–319.

[Li and Liu, 1998] Li, Y. and Liu, K. (1998). Adaptive blind source separation and equalization for multiple-input/multiple-output systems. *Information Theory, IEEE Transactions on*, 44(7):2864–2876.

[Liutkus et al., 2011] Liutkus, A., Badeau, R., and Richard, G. (2011). Gaussian processes for underdetermined source separation. *Trans. on SP*, 59(7):3155–3167.

[MacQueen, 1967] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

[Malek, 2013] Malek, J. (2013). Blind compensation of memoryless nonlinear distortions in sparse signals. In *21st European Signal Processing Conference (EUSIPCO 2013)*, pages 1–5.

[Marvasti and Jain, 1986] Marvasti, F. and Jain, A. K. (1986). Zero crossings, bandwidth compression, and restoration of nonlinearly distorted band-limited signals. *Journal of Optical Society of America A*, 3(5):651–654.

[Meganem et al., 2011] Meganem, I., Deliot, P., Briottet, X., Deville, Y., and Hosseini, S. (2011). Physical modelling and non-linear unmixing method for urban hyperspectral images. In *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 3rd Workshop on*, pages 1–4. IEEE.

[Meganem et al., 2014] Meganem, I., Deville, Y., Hosseini, S., Déliot, P., and Briottet, X. (2014). Linear-quadratic blind source separation using nmf to unmix urban hyperspectral images. *IEEE Transactions on Signal Processing*, 62(7):1822–1833.

[Mei et al., 2009] Mei, T., Yin, F., and Wang, J. (2009). Blind source separation based on cumulants with time and frequency non-properties. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(6):1099–1108.

[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 (IJDAR)*, 14(4):319–333.

[Muller et al., 2001] Muller, K.-R., Mika, S., Ratsch, G., Tsuda, K., and Scholkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEE transactions on neural networks*, 12(2):181–201.

[Naini et al., 2008] Naini, F. M., Mohimani, G. H., Babaie-Zadeh, M., and Jutten, C. (2008). Estimating the mixing matrix in sparse component analysis (SCA) based on partial k-dimensional subspace clustering. *Neurocomputing*, 71(10):2330–2343.

[Noorzadeh et al., 2014] Noorzadeh, S., Niknazar, M., Rivet, B., Fontecave-Jallon, J., Guméry, P., and Jutten, C. (2014). Modeling quasi-periodic signals by a non-parametric model: Application on fetal ECG extraction. In *36th Int. Conf. of EMBC*, pages 1889–1892. IEEE.

[Noorzadeh et al., 2015a] Noorzadeh, S., Rivet, B., and Guméry, P. (2015a). An application of Gaussian processes on ocular artifact removal from EEG. In *37th Int. Conf. of EMBC*, pages 554–557. IEEE.

[Noorzadeh et al., 2015b] Noorzadeh, S., Rivet, B., and Guméry, P. (2015b). A multi-modal approach using a non-parametric model to extract fetal ECG. In *ICASSP*, pages 832–836. IEEE.

[Nuzillard and Bijaoui, 2000] Nuzillard, D. and Bijaoui, A. (2000). Blind source separation and analysis of multispectral astronomical images. *Astronomy and Astrophysics Supplement Series*, 147(1):129–138.

[Parra and Spence, 2000] Parra, L. and Spence, C. (2000). Convolutive blind separation of non-stationary sources. *Speech and Audio Processing, IEEE Transactions on*, 8(3):320–327.

[Pérez-Cruz et al., 2013] Pérez-Cruz, F., Van Vaerenbergh, S., Murillo-Fuentes, J. J., Lázaro-Gredilla, M., and Santamaria, I. (2013). Gaussian processes for nonlinear signal processing: An overview of recent advances. *Signal Processing Magazine*, 30(4):40–50.

[Petersen et al., 2008] Petersen, K. B., Pedersen, M. S., et al. (2008). The matrix cookbook. *Technical University of Denmark*, 7:15.

[Pham, 2000] Pham, D. T. (2000). Blind separation of instantaneous mixture of sources based on order statistics. *IEEE Transactions on Signal Processing*, 48:363–375.

[Quine, 1994] Quine, M. (1994). A result of Shepp. *Applied Mathematics Letters*, 7(6):33 – 34.

[Rasmussen and Williams, 2006] Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian processes for machine learning*, volume 1. MIT press Cambridge.

[Reid, 1987] Reid, J. (1987). Normal functions of normal random variables. *Computers & Mathematics with Applications*, 14(3):157 – 160.

[Reinsch, 1967] Reinsch, C. H. (1967). Smoothing by spline functions. *Numerische mathematik*, 10(3):177–183.

[Revel et al., 2016] Revel, C., Deville, Y., Achard, V., and Briottet, X. (2016). A linear-quadratic unsupervised hyperspectral unmixing method dealing with intra-class variability. In *IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS 2016)*.

[Rivet, 2006] Rivet, B. (2006). *La bimodalité de la parole au secours de la séparation de sources*. PhD thesis, Institut National Polytechnique de Grenoble-INPG.

[Rivet et al., 2010] Rivet, B., Duarte, L. T., and Jutten, C. (2010). Blind extraction of intermittent sources. In *LVA/ICA*, pages 402–409. Springer.

[Rivet et al., 2007] Rivet, B., Girin, L., and Jutten, C. (2007). Visual voice activity detection as a help for speech source separation from convolutive mixtures. *Speech Communication*, 49(7):667–677.

[Rivet et al., 2012] Rivet, B., Niknazar, M., and Jutten, C. (2012). Nonparametric modelling of ECG: applications to denoising and to single sensor fetal ECG extraction. In *Int. Conf. on Latent Variable Analysis and Signal Separation*, pages 470–477. Springer.

[Spivak, 1965] Spivak, M. (1965). *Calculus on Manifolds: A Modern Approach to Classical Theorems of Advanced Calculus*. Advanced book program. Avalon Publishing.

[Taleb and Jutten, 1999] Taleb, A. and Jutten, C. (1999). Source separation in post-nonlinear mixtures. *IEEE Transactions on Signal Processing*, 47(10):2807–2820.

[Talmon et al., 2013] Talmon, R., Cohen, I., Gannot, S., and Coifman, R. R. (2013). Diffusion maps for signal processing: A deeper look at manifold-

learning techniques based on kernels and graphs. *IEEE Signal Processing Magazine*, 30(4):75–86.

[Tenenbaum et al., 2000] Tenenbaum, J. B., Silva, V. d., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.

[Thouvenin et al., 2016] Thouvenin, P.-A., Dobigeon, N., and Tourneret, J.-Y. (2016). Hyperspectral unmixing with spectral variability using a perturbed linear mixing model. *IEEE Transactions on Signal Processing*, 64(2):525–538.

[Tong et al., 1991] Tong, L., Liu, R. W., Soon, V. C., and Huang, Y. F. (1991). Indeterminacy and identifiability of blind identification. *IEEE Transactions on Circuits and Systems*, 38(5):499–509.

[Tong et al., 1990] Tong, L., Soon, V. C., Huang, Y. F., and Liu, R. (1990). AMUSE: a new blind identification algorithm. In *IEEE International Symposium on Circuits and Systems*, pages 1784–1787 vol.3.

[Van Vaerenbergh and Santamaría, 2006] Van Vaerenbergh, S. and Santamaría, I. (2006). A spectral clustering approach to underdetermined post-nonlinear blind source separation of sparse sources. *IEEE Transactions on Neural Networks*, 17(3):811–814.

[Vigário et al., 2000] Vigário, R., Särelä, J., Jousmiki, V., Hämäläinen, M., and Oja, E. (2000). Independent component approach to the analysis of EEG and MEG recordings. *Biomedical Engineering, IEEE Transactions on*, 47(5):589–593.

[Vilenkin, 1978] Vilenkin, N. (1978). *Special Functions and the Theory of Group Representations*. Translations of mathematical monographs. American Mathematical Soc.

[Wesołowski, 1997] Wesołowski, J. (1996-1997). Are continuous mappings preserving normality necessarily linear? *Applicationes Mathematicae*, 24(1):109–112.

[Zare and Ho, 2014] Zare, A. and Ho, K. (2014). Endmember variability in hyperspectral analysis: Addressing spectral variability during spectral unmixing. *IEEE Signal Processing Magazine*, 31(1):95–104.