N° d'ordre: 371/21

Sultan Moulay Slimane University

Sciences and Techniques Faculty

Beni-Mellal

Doctoral Studies Center: Sciences and Techniques

Doctoral program: Mathematics and Applied Physics

Thesis

Presented by
AMAL OURDOU

In order to obtain the degree of
Doctor

Discipline: Computer Science
Speciality: Computer and Data Science

Numerical Analysis of Blind Source Separation Models and Applications

Defended on 30 October 2021 in front of the jury:

President	M. Said RAGHAY	LAMAI-University Cadi-Ayyad
Thesis director	M. Abdelmoutalib METRANE	LIPIM-University Sultan Moulay Slimane
Supervisor	M. Abdelghani GHAZDALI	LIPIM-University Sultan Moulay Slimane
Reporters	M. Abdelilah HAKIM	LAMAI-University Cadi-Ayyad
	M. Abdeljalil NACHAOUI	LMJL-University of Nantes
	M. Lekbir AFRAITES	LMA-University Sultan Moulay Slimane
Examiners	M. Abderrahim ELMOATAZ	GREYCteam, University of Caen Basse Normandie
	M. Amine LAGHRIB	LMA-University Sultan Moulay Slimane

School year : 2020/2021

Acknowledgment

First and foremost I am extremely grateful to the God for the blessings throughout my research work to complete my thesis successfully.

I also want to express my sincere gratitude to Professor Abdelmoutalib Metrane, professor-researcher at the National school of applied science at Khouribga, who agreed to take me in thesis, and to show me his support and confidence. to take me in thesis, and to show me his support and his trust. I would also like to thank you for guiding me well throughout the research work for showing me your human side and guiding me through life, your advices will stick with me forever.

I would like to thank Professor Abdelghani Ghazdali, my supervisor and researcher at the National school of applied science at Khouribga, for your constant availability, for having communicated your experience to me and particularly for having left me a great freedom in the realization of this thesis. I had the pleasure to learn a lot with you, especially in my case where I transitioned from software engineering to the word of mathematics. You're immense knowledge, motivation and guidance helped me all the time whether in research or the writing of this thesis. Without their constant feedback this PhD would not have been achievable.

Alongside my supervisors, I won't forget to express my immense gratitude toward Professor Hafidi Imad, professor and researcher at the National school of applied science at Khouribga, for giving me the encouragement needed for continuing my research as well as sharing insightful suggestions at all time, your help will be forever in my memory.

My deepest thanks to my close and supportive circle of friends who have always been a major source of support, guidance, warmth when things would get discouraging. Thank you for always being there for me, and for all the fun we had. I particularly want to thanks Moad Hakim for his intellectual guidance and assistance in more ways more than one.

Lastly very special thanks to my family. Words can not express how grateful I am to my mother, my father, my sister Imane, and my little brother Mehdi, for all of the encouragement in times of doubt and all sacrifices that you have made on my behalf. I love you dearly.

Abstract

The aim of Blind Source Separation (BSS) is to recover unobserved mixed signals from their mixtures, assuming that no or minimal knowledge about the source signals and/or the mixing system is available. BSS is used in multiple scientific fields such as Image denoising, Bio-medical signals and even Machine learning. In this thesis, we present BSS algorithms in order to separate linear instantaneous mixtures for independent and dependent sources. The principle of our approach is to generalize the proposed algorithm that minimizes the Kullback-Leibler divergence between the copula densities of the source components, creating a new separation criteria based on the mutual information (MI). Copulas are the tool used to model the dependency model of random variable, hence, we deployed it in the case of mixtures of statistically dependent sources. This approach gave very satisfying results, however to have even better results with a faster divergence, we proposed to use the alpha-divergence. Note that the kullback-leibler divergence is just a special case of the alpha-divergence. We show that the proposed approaches can magnificently separate instantaneous mixtures of dependent sources with unknown copula model and/or unknown parameter which is the case in many real applications. Additionally, we tackled the case where the mixture is done in a noise contaminated environment, which make the BSS problem trickier then usual. Finally we illustrated some of these BSS applications such as image denoising, using our proposed approaches.

Resume

Le but de la séparation aveugle des sources (SAS) est de récupérer des signaux mixtes non observés à partir de leurs mélanges, en supposant qu'aucune connaissance ou une connaissance minimale des signaux sources et/ou du système de mélange est disponible. La SAS est utilisée dans de nombreux domaines scientifiques tels que le débruitage d'images, les signaux bio-médicaux et même l'apprentissage automatique. Dans cette thèse, nous présentons des algorithmes BSS afin de séparer des mélanges linéaires instantanés pour des sources indépendantes et dépendantes. Le principe de notre approche est de généraliser l'algorithme proposé qui minimise la divergence de Kullback-Leibler entre les densités de copules des composants de la source, en créant un nouveau critère de séparation basé sur l'information mutuelle (MI). Les copules sont les outils utilisés pour modéliser la structure de dépendance d'une variable aléatoire, d'où son déploiement dans le cas de mélanges de sources statistiquement dépendantes. Cette approche a donné des résultats très satisfaisants, cependant pour avoir des résultats encore meilleurs avec une divergence plus rapide, nous avons proposé d'utiliser l'alpha-divergence. Notez que la divergence de Kullback-Leibler n'est qu'un cas particulier de la divergence alpha. Nous montrons que les approches proposées peuvent magnifiquement séparer des mélanges instantanés de sources dépendantes avec un modèle de copule inconnu et/ou un paramètre inconnu, ce qui est le cas dans de nombreuses applications réelles. De plus, nous avons abordé le cas où le mélange est effectué dans un environnement contaminé par le bruit, ce qui rend le problème BSS plus délicat que d'habitude. Nous illustrons également certaines de ces applications réelles telles que le débruitage d'images.

Contents

List of Figures

List of Tables

Chapter 1

General introduction

Blind source separation (BSS) is a challenging yet highly required field of research in signal and image processing, in Artificial Intelligence and Big Data Analytics. It is considered one of the unsupervised Machine Learning problems. The main distinction between the supervised and the unsupervised approaches is the use of labeled datasets. To put it simply, supervised learning uses labeled input and output as data, while an unsupervised learning algorithm does not. In supervised learning, the algorithm "learns" from the training dataset by iteratively making predictions on the data and adjusting for the correct answer. Unsupervised learning models, in contrast, work on their own to discover the inherent structure of unlabeled data. For BSS where the aim is to recover original sources from their mixtures otherwise called observations, when the knowledge about the source signals and/or the mixing system is very limited or even close to non existing, the nature of this problem makes it an unsupervised machine learning problem. Even though BSS and its related methods were first introduced for signal processing, they are now promising in many application, especially data analysis and data mining including: redundancy reduction, denoising, feature extraction, preprocessing for various classification and recognition tasks. Furthermore, BSS approaches are useful in modelling higher mechanisms of the brain, including modelling of olfactory bulb, auditory system, selective attention and sparse coding [28, 33, 110]. BSS typically handles a mixing model of the following form:

$$x(.) = A[s(.)] + v,$$

where s and x represent respectively the source signals and the observed signals, v represents an additive noise if it exists in case of noise-contaminated mixing systems and A is a transformation, which can be instantaneous (operating on each s to produce x), or global (operating on the whole sequence $s(.)$ of source vectors). The mixing occurs during the propagation of a number of source signals to a collection of sensors each measures a mixture of the original sources with slightly different weights. The output of each sensor is what we call the observations x. These sensors can be a microphone, an antenna, a camera, etc. The signals can be, from radio waves emitted by cell phones, electrical signals from the heart, or by the brain, sound recordings of people having a discussion in the same room, or could even be images, etc. One of the concrete examples where we encounter this phenomenon is the one that happens daily to the human being when his ear receives the different sounds (signals) that are in his environment. In reality, the human being receives a mixture of sources from the environment. However, the human brain is able to separate the source of interest from the mixtures naturally and easily without extra work from the person.

Blind Source Separation tries to imitate the human brain by studying the unmixing and separation of signals or pattern mixtures within a complex multivariate system, having minimal to no a priori knowledge of the target or original sources. The original source is usually a buried pattern of valuable data into a mixture of different sorts of patterns, generally considered independent, but can be dependent. Despite the tremendous effort being put into the development of BSS algorithms, this field area is still being developed for customized techniques that are precisely designed for specific classes of data and specific use cases.

The nature of the BSS problem, makes it an ill posed problem. It is difficult, or even impossible to reconstruct the sources without making a few assumptions or hypotheses. the most common assumptions are the statistical independence of the source vector components and the condition that at most one of the components could be Gaussian. These assumptions makes BSS equivalent to Independent Component Analysis (ICA). ICA would provide the independent sources up to permutation and scaling indeterminacies. Multiple algorithms of this technique were introduced, such as FastICA [79], projection pursuit [84], Infomax [17] etc. The extraction of the original sources can be either done by minimizing the mutual information [11, 55, 128], maximizing the likelihood [25] or maximizing non-Gaussianity [82]. A good detailed review of this issue is provided in [38, 123]. There is another framework used for BSS called Sparse Component Analysis (SCA), it has been most successfully applied if the sources could be represented sparsely in a given basis [95, 129, 160, 161].

For most cases, BSS was particularly solved for mixtures of statistically independent sources, however, the independence property of sources may not hold in some real-world situations, especially in biomedical signal processing and image processing, etc. Some extensions for ICA were then introduced for separating mixtures of dependent sources. These approaches are called Dependent Component Analysis (DCA) as a whole. The first extended ICA model is the Multidimensional Independent Component Analysis (MICA) [24] where the components are not assumed to be all mutually independent, instead, it is assumed that the source signals can be divided into i-tuples, such that the source signals inside a given i-tuple may be dependent on each other, but the i-tuples should be independent on each other. We also have variance dependent BSS [81], where they simply assume that the sources are dependent only through their variances and that the sources have temporal correlation. We also have, Tree-dependent component analysis [12], topographic ICA [80], subband decomposition ICA(SDICA) [141, 164], etc. The paper [105], gives an overview on the early approaches used for dependent BSS. Nonnegative matrix factorization was also used for independent and dependent sources [67, 159], Another approach is the window-disjoint orthogonality [145, 146]. Later on, Copula was used to model the dependency structure of the sources [1, 111], this framework was proposed as a generalization on ICA, and then in [69, 70, 96] these copula approaches were improved and gave more details on how to achieve the separation for different types of samples.

As we said previously, the problem of Blind Source Separation is to reconstitute sources from unknown mixtures of sources. Many algorithms have been proposed in the literature to obtain the solution. These algorithms are sometimes very different, in terms of hypotheses and principles of separation, but in most cases, the general criterion of source separation is under the hypothesis of statistical independence and non-gaussianity of the sources. However, in many situations and real-world problems, these assumptions are not satisfied. The natural question is: is there a solution to this problem when the components of the source signals are dependent? If so, is it the best the solution we can get? If not, how do we get the solution? The use of the copula theory and the divergence criteria, will allow us to meet all these expectations. The work being presented throughout this thesis is a contribution into the field of Blind Source Separation, and more specifically for mixtures of statistically dependent sources, which was not studied as much as for the independent case. Upon reviewing the available techniques we improved the best techniques and even proposed a new BSS procedure that handles noise-contaminated mixtures. Throughout this thesis the dependency structure of the component sources will be modeled using copulas. The thesis not only presents the theoretical aspect of the method, but also presents the results of applying the proposed methods for real life problems such as image denoising and images separation.

This document is organized as follows. The first part presents the problem of blind source separation and a state of the art of the most widespread methods, a summary on the copula theory and an introduction to the divergence used as a cost function for our new proposed approach. The second part of this document is devoted to the development of a new method of source separation, using copulas, and the theoretical and experimental study of its performance and its applications. A general conclusion complete this document.

Part 1: Background information

This part of the dissertation consists of three chapters. In the first chapter, we will begin by defining BSS in more detail. Then we explain the mixing and separation procedures, distinguishing between the different types of mixtures used in blind source separation. Then we model this problem for instantaneous and convolutional mixtures, and end this chapter with a set of various methods used in BSS. The second chapter is devoted to the theory of copula. It provides an introduction to the notion of copula. We begin by giving a history of copula and explaining the choice of the copula to model the dependency structure, then we recall the main theorems and the most important properties of the copula and its role in the study of the dependency of random variables. We will also present the main types of copulas frequently used in practice, as well as their properties. Next, we will introduce two methods to estimate the copula. The first method is non-parametric. The second approach is semi-parametric which assumes a parametric model for the copula and non-parametric for the marginal distributions. Then, we are going to explain the methods used to generate vectors, according to the various copulas. Finally, we will present a method in order to select the most appropriate copula for a certain data set. The last chapter of the first part is about the Alpha-divergence, first we present what is a divergence in general, then showcase the Kullback-Leibler divergence and its properties and how it is just a special case of the alpha-divergence and we finish the chapter by giving more details about the alpha-divergence, its most important definition and properties.

Part 2: Blind source separation

The second part consist of four chapters. In the fist chapter we introduce how we can use Copula to solve the instantaneous BSS problem using the kullback-leibler divergence. The second chapter is dedicated to our new Blind Source Separation methodology for instantaneous linear mixing in the presence of additive noise. This approach is based on the minimization of a regularized criterion. Precisely, it consists in combining the total variation method to de-noise the observations, with the Kullback-Leibler divergence between the copula densities to separate the mixtures. In chapter 7, we propose the main results of the article published in the scientific journal Circuits, Systems and Signal Processing (CSSP), where we developed a new method, based on the optimization of a criterion between copula using the alpha-divergence. The novelty lies in the consideration of dependent sources and using a new criterion that converge faster and better then other divergences. This chapter also presents separation results on various types of synthetic data. And finally we showcase various applications of BSS using our approach such as Image denoising and bio-medical signals separation.

Publications

1. Ourdou, Amal & Ghazdali, Abdelghani & Laghrib, Amine & Abdelmoutalib, Metrane. (2021). Blind Separation of Instantaneous Mixtures of Independent/Dependent Sources. Circuits, Systems, and Signal Processing. 1-24. 10.1007/s00034-021-01672-2.

2. Ourdou, Amal & Ghazdali, Abdelghani & Metrane, Abelmoutalib & Hakim, Moad. (2021). Digital document image restoration using a blind source separation method based on copulas. Journal of Physics: Conference Series. 1743. 012034. 10.1088/1742-6596/1743/1/012034.

3. Ghazdali, Abdelghani & Abdelmoutalib, Metrane & Ourdou, Amal. (2021). Blind Source Separation for Text Mining. Journal of Physics: Conference Series. 1743. 012018. 10.1088/1742-6596/1743/1/012018.

4. Ghazdali, Abdelghani & Hakim, Moad, & Laghrib, Amine & Mamouni, Nazha & Abdelmoutalib, Metrane & Ourdou, Amal. (2021). Blind separation of noisy mixtures of independent and dependent sources, Applied and Computational Harmonic Analysis.

5. Ourdou, Amal & Ghazdali, Abdelghani & Laghrib, Amine & Abdelmoutalib, Metrane. (2021). Robust approach for blind separation of noisy mixtures of independent and dependent sources, Mathematical Modeling and Computing.

6. Ghazdali, Abdelghani & Abdelmoutalib, Metrane & Ourdou, Amal. (2021). Blind noisy mixture separation for dependent sources, Mathematical Control and Numerical Applications.

7. Ourdou, Amal & Ghazdali, Abdelghani & Abdelmoutalib, Metrane. (2021). Blind Separation of dependent sources using Copula, Mathematical Control and Numerical Applications.

Part I

Background information

Chapter 2

State of Art

2.1 Introduction

In this chapter we will begin by giving a brief history on BSS and defining it in more details. Then we explain the mixing and separation procedures, distinguishing between the different types of mixtures that happens. Then we model this problem for instantaneous and convolutional mixtures. After that, we will present different existing techniques for solving the blind source separation problem focusing on Independent Component Analysis, Non-negative Matrix Factorization and finally Sparse Component Analysis.

2.2 Bref history

The source separation problem was first proposed by Bernard Ans, Jeanny Herault and Christian Jutten around 1982, for motion decoding in vertebrates, in the context of neural modeling [133]. In the context of communications, the problem appears to have been sketched separately [15]. The first related contributions to Signal Processing conferences [78] and Neural Networks conferences [8, 78] were made around 1985. These articles quickly drew the attention of signal processing academics, primarily in France and later on in Europe. Since the mid-1990s, the interest in neural networks was huge and a large number of academics with expertise in a variety of domains, including signal processing, statistics, and neural networks, have been working on the BSS problem. At international conferences, a number of special sessions on these topics have taken place.

Instantaneous linear mixtures (memoryless mixtures) were the first model that were researched for source separation [78]. Later on, at the beginning of the 90s a generalization to convolutive mixtures was studied [39], and by the end of 1990s nonlinear mixtures were addressed [21, 86, 140]. Furthermore, in 1987, independent component analysis (ICA) [92] was introduced, which is a general method for solving BSS problems based on statistical independence of unknown sources, that later was formalized for linear mixtures by Comon in 1991 [40].

The number of papers written on the topic of BSS or ICA is enormous: Google Scholar records 1.810.000 research papers in Engineering, Computer Science, and Mathematics as of may 2021. Only a few books, on the other hand, cover the BSS problem and the key concepts for solving it. In particular, the book [85], written by experts in Neural Networks, contains only algorithms built by the Machine Learning community. Another rather complete book appeared slightly later [28]. However, some ways of addressing the problem were still missing (semi-blind approaches, Bayesian approaches, Sparse Components Analysis, etc.). More specific problems, i.e. separation of audio sources [113], or separation in nonlinear mixtures [7], have been the subject of other contributions. And in 2010, [41] was published giving a complete state of art.

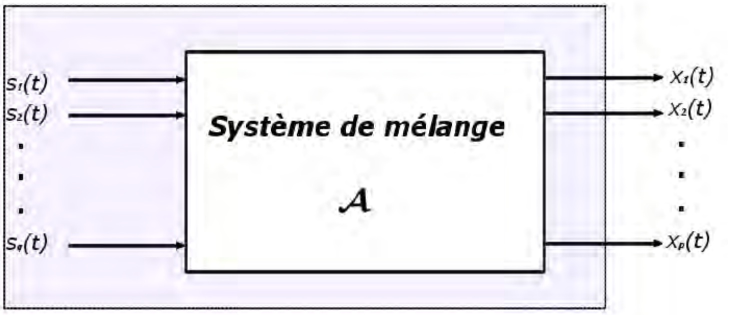
2.3 Problem statement

Generally, in our environment, source signals propagate to an end point (sensor). In this environment called propagation medium, the sources naturally undergo different transformations. Consequently the sources will be transformed then superimposed, thus building the mixtures we receive on the sensors. By placing ourselves on the sensors, we will observe (measure) more or less complex mixtures of these original/initial sources. To study the effects of each source on all the signals and to discern the original signals in order to find the components of the mixtures, it is necessary to separate the sources. The problem of blind source separation is a fundamental and very general problem. It is generally modelled independently of the field of application. The paragraphs below equate the problem by explaining the mixing and separation procedures.

2.3.1 The mixing process

To formulate the mixing process of blind source separation, let $s(t) := (s_1(t), s_2(t), \cdots, s_q(t))^\top$, be the original sources, where $s_j(t)$ is the t^{th} sample of the j^{th} source element, with $j = 1, \cdots, q$ and q the number of sources, and let $x(t) := (x_1(t), x_2(t), \cdots, x_p(t))$ be the observation vectors, where $x_i(t)$ is the t^{th} sample from the observation source coming from the i^{th} sensor, where $i = 1, \cdots, p$ and p is the number of sensors or observations, the following figure explains the mixing process

Figure 2.1: The mixing process



The mixing process between the sources and the observations (which are the mixed sources) is formed as follow:

$$x(t) := A[s(t)] \tag{2.1}$$

where A is the mixing system. If we have T samples from each source and each observation, then the source and observation samples can be grouped into the S

and X matrices respectively. The mixing process between sources and observations can be rewritten in the following matrix form:

$$X := A[S] \tag{2.2}$$

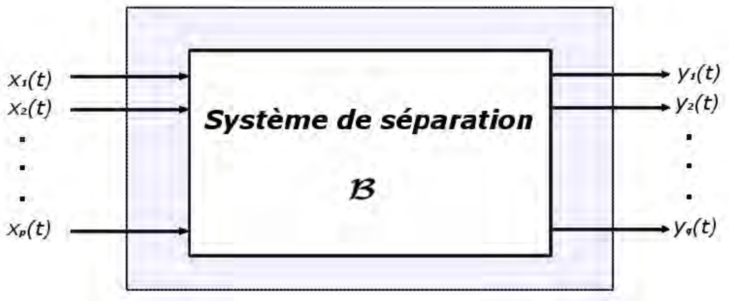
where $X \in \mathbb{R}^{p\times T}$ is the observations matrix and $S \in \mathbb{R}^{q\times T}$ is the sources matrix.

In the problem of blind source separation, the sources and the mixing system are not known a priori, only observations are available.

2.3.2 The separation

From the sole knowledge of the observations (source mixtures), blind source separation consists in constructing a separation system, denoted B, which allows the best possible estimation of the initial sources s. The following figure illustrates the separation procedure:

Figure 2.2: The separation process



Let $y(t) := (y_1(t), y_2(t), \cdots, y_j(t))$ be the estimated sources, where where $y_j(t)$ is the t^{th} sample of the j^{th} estimated source element, with $j = 1, \cdots, q$. the separation formula is giving by

$$y(t) := B[x(t)] \tag{2.3}$$

In matrix form the above relation can be written as follows:

$$Y := B[X] \tag{2.4}$$

where $B \in \mathbb{R}^{q\times p}$ is the matrix for the estimated sources.

The ideal solution for the blind source separation problem is when $B = A^{-1}$, however due to the process being blind, in other words we have no knowledge about the nature of the mixing process and environment nor the nature of the original sources, BSS is considered an ill posed problem. In fact every couple (A', s') where $A'[s'] = x$ could be an acceptable solution to the problem , without any a priori information on the sources and on the mixing system, it is not justified to privilege a solution eligibly compared to another. Given these ambiguities, solving this problem therefore requires assumptions about the sources and the mixing system in order to arrive at a single solution or at least reduce the number of allowable solutions.

Thus, BSS has several degrees of difficulty, depending on the characteristics of the sources and, more importantly, how the mixing system is being considered. The model of the mixing system must match the physical phenomenon linking the sources to the observations. Therefore, the choice of an appropriate model is necessary because it allows the selection of the separation structure to adapt to the problem being addressed.

2.4 BSS Method Categorization

From the available literature concerning Blind Source Separation, we can say that Linearity, Time-delay, and Determinism are three key characteristics that can be used to classify the BSS problem. This classification leads to the following categories: Linear versus Non-linear BSS, Instantaneous versus Convolutive BSS, and Overcomplete / Overdetermined versus Underdetermined BSS.

2.4.1 Linearity in BSS Problems

Owing to its simplicity of study and explicit separability, the linear time-invariant (LTI) model is the most extensively studied and investigated topic of BSS. The Linear BSS model assumes that the source mixture can be characterized using a linear combination, as illustrated in the equation below:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{np} \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_p(t) \end{bmatrix} + \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_p(t) \end{bmatrix} \tag{2.5}$$

Unless extra information is provided as a priori features of the sources, which is common in linear systems where multiplication is no longer commutative and the mixing coefficients according to which the mixing process occurred are non-evident, the scale and order of the separated sources will remain imprecise and unclear.

As for the Non-linear model, a more realistic environment is considered, where the observed signals are non-linear distorted signals. Despite the fact that literature holds a large number of studies and proposed algorithms on Linear BSS, nonlinear BSS has not been well developed, a review of which is presented in [94]. Linear BSS techniques have shown their inability to separate the sources which are non-linearly mixed, which led to the emerging of the Non-linear techniques, of which we state Post Non-linear (PNL) [165] and Bi-Linear (or Linear Quadratic) mixtures [48], Convolutive Post Non-linear Mixtures [168] and Conformal mappings. The mathematical representation of such Non-linear models is stated below

$$x(t) = f(s(t)) \tag{2.6}$$

where $f : \mathbb{R}^M \to \mathbb{R}^N$ is the unknown non-linear mixing function, N and M are the number of source and observation signals, respectively. The BSS problem may also be written in the following form:

$$\text{Find g such that: } g(x(t)) = g(f(s(t))) = \widehat{s}(t) \tag{2.7}$$

where $g : \mathbb{R}^N \to \mathbb{R}^M$ is the separating function to be estimated and $x(t)$ the vector of reconstructed sources, which is the output of the BSS algorithm.

2.4.2 Mixing Delay in BSS Problems

In terms of the time-wise mixing element of the sources, BSS can be built to either deal with sources that are simultaneously mixed, i.e. with the absence of time delays, as defined by the Instantaneous Mixture Model, or to deal with mixtures that were formed out of different source signals in form of combinations of several time-delayed versions of the sources themselves and/or mixed signals themselves which defines the Convolutive Mixture Model (CMM).

In more simplistic mixing models, the mixture is thought to be the sum of variably weighted source signals, but in most real-world implementations, such as acoustics, the various sources are weighted and delayed, adding to the sum or mixture with many delays, implying, for example, the manifestation of the numerous pathways through which acoustic impulses flow to a microphone. Such mixtures are termed convolutive mixtures that vary in the number of delay elements, which are application-specific, reaching thousands of delay elements in acoustics. The mentioned acoustic signals might be speech or music or underwater sonar signals, radio signals captured by antenna arrays as mixtures, astronomical data, functional brain imaging data and bio-potentials.

The mathematical mixing model of the Instantaneous Mixture Model is as follows

$$x(t) = As(t) + u(t) \tag{2.8}$$

This model is also known as the linear mixture model, where $A = A_0$ is an $M \times N$ matrix containing the mixing coefficients. As for the delayed sources' case, assuming a reverberation-free environment with propagation delays, the mixing model may be represented as follows in:

$$x_m(t) = \sum_{n=1}^{N} a_{mn} s_n(t - k_{mn}) + u_m(t) \tag{2.9}$$

where k_{mn} is the propagation delay between source n and sensor m.

$$x(t) = \sum_{k=0}^{K-1} A_k s(t-k) + u(t) \tag{2.10}$$

However, in the derivation of many algorithms, for simplification purposes, the environment is considered to be noise-free, which reduces to (2.11):

$$y(t) = \sum_{k=0}^{K-1} A_k x(t-k) \tag{2.11}$$

2.4.3 Determinism in BSS Problems

Determinism, on the other hand, is a classification criterion for BSS situations in which the number of sources N is compared to the number of observations M. As a result, there are three alternative outcomes:

- N > M: The number of sources is greater than the number of observations.

  Such systems are called over-determined systems and are easily solved using linear BSS methods, the mixing matrix being invertible.

- N = M: The number of sources is equal to the number of observations.

  Such systems are denoted determined systems and are also easily solved using linear BSS methods, the mixing matrix being an invertible square matrix.

- N < M: The number of sources is greater than the number of observations.

  Such systems are called over-determined systems and may not be solved using linear methods even under perfect knowledge of the mixing system, i.e. the sources may never be recovered via linear methods.

2.5 BSS Techniques

Throughout the years and since Blind source separation first appearance, various approaches were proposed in literature. In this section, we present some of the most famous and commonly used methods, giving a brief summary on which cases of BSS they are used and how they are applied to achieve the separation.

2.5.1 Independent Component Analysis

Independent component analysis (ICA) is a statistical and computational methodology for uncovering hidden components that lie behind a set of random variables, observations, or signals.

In the ICA model, the data variables are assumed to be linear mixtures of some unknown latent variables, and the mixing system is also unknown. The latent variables are assumed nongaussian and mutually independent. They are called the independent components of the observed data.

Although ICA appears to be connected to principal component analysis and factor analysis, it is a far more powerful methodology capable of identifying underlying factors or sources when these traditional methods fail.

The ICA concept was initially proposed by Comon [37] and was initially proposed to solve the blind source separation (BSS). The source separation problem cannot be solved if there is no knowledge of either A or S, apart from the observed mixed data X. If the mixing matrix A is known and the additive noise v is negligible, then the original sources can be estimated by evaluating the pseudo inverse of the matrix

A, which is known as the unmixing matrix B.

Estimation of the underlying independent sources is the primary objective of the BSS problem in the case of mixtures of statisticaly independent sources. The problem is solvable with the following restrictions:

- The sources are statistically independent.

- At most, one of the sources is Gaussian distributed.

- The mixing matrix is of full rank.

The above discussions make it clear that statistical independence is the key foundation of independent component analysis (ICA). For the case of two different random variables x and y, x is independent of the value of y, if knowing the value of y does not give any information on the value of x. Statistical independence is defined mathematically in terms of the probability densities as - the random variables x and y are said to be independent, if and only if

$$p_{x,y}(x,y) = p_x(x)p_y(y) \tag{2.12}$$

where $p_{x,y}(x,y)$ is the joint density of x and y, $p_x(x)$ and $p_y(y)$ are marginal probability densities of x and y respectively. Generalizing this for a random vector $s = [s_1, ..., s_N]^T$ with multivariate density $p(s)$ has statistically independent components, if the density can be factorized as

$$p(s) = \prod_{i=1}^{N} p_i(s_i) \tag{2.13}$$

When two variables $s1$ and $s2$ are independent, the density of $s1$ is unaffected by $s2$. Statistical independence is a far more stronger property than uncorrelatedness, which only considers second order statistics. The variables are uncorrelated if they are independent, but not if they are correlated.

The data model for independent component analysis is estimated by formulating a function which is an indicator of independence in some way and then minimizing or maximizing it. Such a function is often called a contrast function or cost function or objective function. The optimization of the contrast function enables the estimation of the independent components. The ICA method combines the choice of an objective function and an optimization algorithm. The statistical properties like consistency, asymptotic variance, and robustness of the ICA technique depend on the choice of the objective function and the algorithmic properties like convergence speed, memory requirements, and numerical stability depends on the optimization algorithm. The contrast function in some way or the other is a measure of independence. One of the common contrast function used for ICA is the Mutual Information, which is a natural measure of dependency between random variables i.e. it is a measure of the information that a member of a set of random variables has on the other random variable in the set.

If y is a n-dimensional random variable and $p_y(u)$ its probability density function, then vector y has mutually independent components, if and only if

$$p_y(u) = \prod_{i=1}^{n} p_{y_i}(u_i) \tag{2.14}$$

A natural way of checking whether y has Independent Components is to measure a distance between both sides of the above equation. The average mutual information of y as given by Comon as

$$MI(p_y) = \int p_y(u) log(\frac{p_y(u)}{\prod_{i=1}^{n} p_{y_i}(u_i)} du) \tag{2.15}$$

The mutual information is equal to zero if and only if the variables are mutually independent and is otherwise strictly positive.

2.5.2 Non-negative Matrix Factorization

The second method that we are going to discus is the Non-negative Matrix Factorization (NMF) method. It is a method based on the non-negative nature of the data. Initially introduced in the mid-1990s by Paatero and Tapper [126] under the name PMF (Positive Matrix Factorization), non-negative matrix factorization was revisited in 2001 by Lee and Seung [104] as a means of decomposing an image on the basis of its elementary elements.

The principle of factorization in non-negative matrices is as follows: Under the assumptions of non-negative sources (S≥0) and mixing matrix coefficients (A≥0), factorization into non-negative matrices consists in decomposing the non-negative matrix of X observations into the product of two non-negative matrices, one representing the estimated sources $\widehat{S}$ and the other the estimated mixing matrix $\widehat{A}$ such that

$$\widehat{A}\widehat{S} :\simeq X. \tag{2.16}$$

NMF methods are iterative algorithms that generally consist in minimizing a chosen criterion. The aim is to find the non-negative matrices $\widehat{A}$ and $\widehat{S}$ that minimize the measure of divergence or distance between the product matrices and the observation matrix, under the constraints of positivity

$$(\widehat{A}, \widehat{S}) = arg_{A\geq 0}\, min_{S\geq 0} \mathbb{D}(X, AS) \tag{2.17}$$

where $\mathbb{D}$ designates the measure of distance (divergence, similarity, etc.) between two matrices or vectors. The choice of the measure $\mathbb{D}$ often depends on the data structure or noise [32]. The simplest and most widely used measure is the Frobenius standard :

$$\mathbb{D}(X, AS) := \frac{1}{2} \parallel X - AS \parallel_F^2 \tag{2.18}$$

The Frobenius standard is well suited when the data contains Gaussian additive noise [32]. For non-Gaussian noise, the Kullback-Leibler divergence is often used.

Other measures of divergences can also be used such as alpha divergences [29], Bregman divergences [50] and beta divergences. Many algorithms are proposed to optimize this problem, including the multiplicative update method [104], and the constrained least square method of non-negativity [4, 19] , the projected gradient method [108, 163] and the Quasi Newton method [162].

Moreover, having only non-negativity as a constraint of the sources and the mixing matrix is not sufficient to guarantee the uniqueness of factorization [52, 101]. Methods based on on NMF have two main drawbacks. On one hand, there is an indeterminacy that adds to the problem, without additional constraint, the solution can only be found at an invertible matrix M. Indeed, if $(\widehat{A}, \widehat{S})$ is a solution of the NMF then $X = \widehat{A}\widehat{S} = \widehat{A}MM^{-1}\widehat{S}$, therefore $(\widehat{A}M, M_{-1}\widehat{S})$ is also a solution such as $\widehat{A}M \geq 0$ and $M^{-1}\widehat{S} \geq 0$, in this case M is said to be an NMF indetermination. The solution is therefore not unique.

On the other hand, in NMF, the criteria can only be convex according to one of the two matrices produced, but not for both. The algorithms therefore only allow to converge to a local minimum. Consequently, the convergence result depends strongly on the initialization of the [119] algorithm.

To make the number of solutions admissible, some NMF methods propose to add other constraints to the sources and-or the mixing matrix. This can be done by adding regularization terms to the initial divergence measure, in order to take into account a priori knowledge about the problem.

2.5.3 Sparse Component Analysis

Sparse Component Analysis (SCA) is used to estimate the mixing matrix in the case of sparing sources. This class of methods appeared over twenty years ago [73, 130]. It is based on the hypothesis that the sources are parsimonious either in their original representation or after decomposition on a dictionary. A signal is said to be parsimonious if most of its coefficients in a given area of representation are void. In the case of the blind separation of sources, the fact that the sources have representations where most of the coefficients are zero is not sufficient to perform the separation. In order to exploit parsimony, the simplest approach is to isolate each of these sources individually in areas of the observed signals. The ideal is to have a representation dictionary for which the sources have disjoint supports [93]. But in some methods it is sufficient to have for each source of the mixture, a zone in the observed signals on which it is the only one to be non-zero, for example in the case of spectral signals: it is a matter of having a spectral zone on which only one of the sources is non-zero; in the images, this would correspond to having an area of pixels where only one source is present. According to the initial assumptions about the sources, there are three categories of methods in the SCA :

- Methods based on the hypothesis of strong parsimony conditions called W-Disjoint-Orthogonality (WDO). Sources are said to be WDO if in each point of the domain of analysis, only one of these sources is active. This hypothesis assumes that the sources are completely disjoint media [91].

- Quasi-non-parcimonious methods for which it is sufficient for each source to be isolated in certain small areas of the analysis domain. These approaches generally start by detecting areas where only one source is present in the observations, in order to use them to estimate the mixing matrix [2, 49].

- Hybrid methods falling between the two previous categories of methods. Some approaches relax the WDO hypothesis in particular cases, others use a confidence measure on the mono-source quality of points in the analysis domain [9, 106].

2.6 Conclusion

In this chapter we presented the history of Blind Source Separation while giving more details about the mixing and the separation process, showcasing the different types and variations of BSS such as linear, non-linear, also instantaneous or convolutive mixtures and the determinism in the BSS problem, and finalizing the chapter with some of the most used techniques for BSS such as ICA, NMF and SCA.

Chapter 3

Introduction to Copula

3.1 Introduction

The separation problem of instantaneous mixtures for possibly dependent component sources is addressed in this thesis. The dependency structure of the component sources will be described using copulas. Therefore, in this chapter, we give definitions and properties of copulas, we present some of copula models, and finally we expose some procedures for copula model selection as well as method of statistical parameter estimation from the data.

3.2 Definitions and properties

It is common practice to study the dependency structures between two or more random variables. Several measuring tools, such as Pearson's correlation coefficient, Kendall's tau, and Spearman's rho, have been presented to accommodate and assess the dependence between random variables. These measures are simple to compute and can be easily interpreted, however they are not able to detect all of the forms of dependency of the random variables, hence, it was necessary to find other measures capturing the entire dependency structure. In fact, the copula function has the advantage of completely modelling the dependency structure between random variables.

Over the last three decades, copula theory has undergone considerable improvement in research fields such as finance, actuarial science, medicine, hydrology and biology. The term copula comes from the Latin word "cop˜ulae", which means bond, link, alliance or union. This function was sometimes referred to by other names such as the dependency function. In statistics, the notion of copula appears in certain works by Frechet [64] on the study of contingency tables. For the first time, the word copula has been used, in the theory of multivariate laws, was by Sklar [139]. The existence theorem of copulas is generally attributed to the latter. Sklar's theorem guarantees the existence and uniqueness of the copula function, characterizing the dependency between the components of the vector if the marginal distributions of the random vector are continuous. The distribution of the random vector is perfectly defined by the data of the marginal distributions and the copula.

Copula is a function which couples a joint distribution function with its univariate marginals. It is a multivariate distribution function on the unit cube $[0,1]^p$, with uniformly distributed marginals on the interval $[0,1]$. Consider a random vector $\boldsymbol{Y} := (Y_1,\ldots,Y_p)^\top \in \mathbb{R}^p$, $p \geq 2$, with joint distribution function (d.f.)

$$F_{\boldsymbol{Y}}(\cdot) : \boldsymbol{y} \in \mathbb{R}^p \mapsto F_{\boldsymbol{Y}}(\boldsymbol{y}) := F_{\boldsymbol{Y}}(y_1,\ldots,y_p) := \mathbb{P}(Y_1 \leq y_1,\ldots,Y_p \leq y_p), \tag{3.1}$$

and continuous marginal d.f.'s

$$F_{Y_i}(\cdot) : y_i \in \mathbb{R} \mapsto F_{Y_i}(y_i) := \mathbb{P}(Y_i \leq y_i), \, \forall i = 1,\ldots,p. \tag{3.2}$$

Sklar characterization theorem [139] ensures the existence of a unique function $\mathbb{C}_{\boldsymbol{Y}}(\cdot) : [0,1]^p \to [0,1]$ such that

$$F_{\boldsymbol{Y}}(\boldsymbol{y}) = \mathbb{C}_{\boldsymbol{Y}}(F_{Y_1}(y_1),\ldots,F_{Y_p}(y_p)), \, \forall \boldsymbol{y} := (y_1,\ldots,y_p)^\top \in \mathbb{R}^p. \tag{3.3}$$

The function $\mathbb{C}_{\boldsymbol{Y}}(\cdot)$ is called copula (of the random vector $\boldsymbol{Y}$). One can show that

$$\mathbb{C}_{\boldsymbol{Y}}(\boldsymbol{u}) = \mathbb{P}(F_{Y_1}(Y_1) \leq u_1,\ldots,F_{Y_p}(Y_p) \leq u_p), \, \forall \boldsymbol{u} = (u_1,\ldots,u_p)^\top \in [0,1]^p. \tag{3.4}$$

Conversely, for any marginal d.f.'s $F_1(\cdot),\ldots,F_p(\cdot)$, and any copula function $\mathbb{C}(\cdot)$, the function $\mathbb{C}(F_1(\cdot),\ldots,F_p(\cdot))$ is a multivariate d.f. on $\mathbb{R}^p$. On the other hand, since the marginal d.f.'s $F_{Y_j}(\cdot)$, $j = 1,\ldots,p$, are assumed to be continuous (on $\mathbb{R}$), then the random variables $F_{Y_1}(Y_1),\ldots,F_{Y_p}(Y_p)$ are uniformly distributed on the interval $[0,1]$. Therefore, the copula $\mathbb{C}_{\boldsymbol{Y}}(\cdot)$ is multivariate d.f. on $[0,1]^p$ with uniform marginals on $[0,1]$. If the components $Y_1,\ldots,Y_p$ are statistically independent, then we can see from (3.4), that the corresponding copula can be written as

$$\mathbb{C}_{\boldsymbol{Y}}(\boldsymbol{u}) = \prod_{i=1}^{p} u_i =: \mathbb{C}_0(\boldsymbol{u}), \, \forall \boldsymbol{u} \in [0,1]^p. \tag{3.5}$$

It is called the copula of independence. Let, if well defined,

$$\boldsymbol{c}_{\boldsymbol{Y}}(\boldsymbol{u}) := \frac{\partial^p \mathbb{C}_{\boldsymbol{Y}}(\boldsymbol{u})}{\partial u_1 \cdots \partial u_p}, \, \forall \boldsymbol{u} \in [0,1]^p, \tag{3.6}$$

be the copula density of the random vector $\boldsymbol{Y}$. The copula density of independence, denote it $\boldsymbol{c}_0(\cdot)$, is the function taking one on $[0,1]^p$ and zero otherwise. In fact, we can write, from (3.5) and (3.6),

$$\boldsymbol{c}_0(\boldsymbol{u}) := \frac{\partial^p \mathbb{C}_0(\boldsymbol{u})}{\partial u_1 \cdots \partial u_p} = \frac{\partial^p \prod_{i=1}^{p} u_i}{\partial u_1 \cdots \partial u_p} = \mathbb{1}_{[0,1]^p}(\boldsymbol{u}), \, \forall \boldsymbol{u} \in [0,1]^p, \tag{3.7}$$

where, for any set A, the notation $\mathbb{1}_A(\cdot)$ is used for the indicator function of A, i.e., the function defined by $\mathbb{1}_A(x) = 1$ if $x \in A$ and $\mathbb{1}_A(x) = 0$ if $x \notin A$, for all x. Let $f_{\boldsymbol{Y}}(\cdot)$, if it exists, be the probability density on $\mathbb{R}^p$ of the random vector $\boldsymbol{Y}$, and $f_{Y_1}(\cdot),\ldots,f_{Y_p}(\cdot)$ the marginal probability densities of the random variables $Y_1,\ldots,Y_p$, respectively. Then we have, from (3.3), the interesting relation

$$f_{\boldsymbol{Y}}(\boldsymbol{y}) = \left(\prod_{i=1}^{p} f_{Y_i}(y_i)\right) \boldsymbol{c}_{\boldsymbol{Y}}(F_{Y_1}(y_1),\ldots,F_{Y_p}(y_p)), \, \forall \boldsymbol{y} \in \mathbb{R}^p. \tag{3.8}$$

Combining the relations (3.7-3.8), one can show that $\boldsymbol{c}_{\boldsymbol{Y}}(\boldsymbol{u}) = \boldsymbol{c}_0(\boldsymbol{u})$, $\forall \boldsymbol{u} \in [0,1]^p$ if and only if (iff) the components of the vector $\boldsymbol{Y}$ are independent.

Copula is intimately connected to measures of association such as Kendall's tau, among many others. It is defined in the bivariate case, as follows. Let $\boldsymbol{Y} := (Y_1, Y_2)^\top$ and $(Y_1', Y_2')^\top$ be i.i.d. random vectors with value in $\mathbb{R}^2$. The population version of Kendall's tau, $\tau(\boldsymbol{Y})$ of $\boldsymbol{Y}$, is defined as the probability of concordance minus the probability of discordance, i.e.,

$$\tau(\boldsymbol{Y}) := \mathbb{P}[(Y_1 - Y_1')(Y_2 - Y_2') > 0] - \mathbb{P}[(Y_1 - Y_1')(Y_2 - Y_2') < 0]. \tag{3.9}$$

Recall that $\tau(\boldsymbol{Y}) \in [-1,1]$, and that $\tau(\boldsymbol{Y}) = 0$ if the components, Y_1 and Y_2 of $\boldsymbol{Y}$, are statistically independent. In the bivariate case, we have the following relation between $\tau(\boldsymbol{Y})$ and $\mathbb{C}_{\boldsymbol{Y}}$, see e.g. [120],

$$\tau(\boldsymbol{Y}) = 4 \int_{[0,1]^2} \mathbb{C}_{\boldsymbol{Y}}(u_1, u_2)\, d\mathbb{C}_{\boldsymbol{Y}}(u_1, u_2) - 1. \tag{3.10}$$

Let $(y_1(1), y_2(1))^\top, (y_1(2), y_2(2))^\top, \ldots, (y_1(N), y_2(N))^\top$ be N i.i.d. realizations with the same distribution as $\boldsymbol{Y} = (Y_1, Y_2)^\top$. An interesting statistical estimate of $\tau(\boldsymbol{Y})$ is the so-called empirical Kendall's tau (of $\boldsymbol{Y}$), denote it $\widehat{\tau}(\boldsymbol{Y})$, defined by

$$\widehat{\tau}(\boldsymbol{Y}) := \frac{2}{N(N-1)} \sum_{i<j} sgn(y_1(i) - y_1(j))\ sgn(y_2(i) - y_2(j)), \tag{3.11}$$

where $sgn(\cdot)$ is the function defined by $sgn(z) = 1$ if $z \geq 0$, and $sgn(z) = -1$ if $z < 0$, for any $z \in \mathbb{R}$.

Why the copula and not the correlation coefficient?

In signal processing, especially in source separation, we often evaluate the dependency between two or more sources, using their correlation. However, this indicator does not does not allow to model dependency on one hand and depends on margins on the other. This indicator performs well when the dependency relationship is linear and the universe under consideration is Gaussian. It is very useful for families of elliptic distributions (because for these distributions non-correlation implies independence). On the contrary, in source separation, the Gaussian case is rarely used. In addition, this measure of dependence often used by practitioners has several limitations such as:

- Correlation is a scalar measure of dependence that cannot tell us everything. what we would like to know about the structure of addiction

- The correlation coefficient and its scope of characterization only works for the Gaussian variables, for which correlation and dependence cover the same reality

- The correlation coefficient can be the same whereas the dependency structure is totally different especially for extreme values

- The correlation coefficient is not defined if the moments of order two of the variables random are not finished. This is not an appropriate measure of dependence for the heavy tail distributions where the variances can be infinite

- The linear correlation coefficient is not invariant by transformations strictly increasing

Other dependency indicators are proposed, based on the discrepancies and concordances observed in a sample, such as non-linear and non-parametric correlation coefficients (Kendall's tau and Spearman's rho). These are good overall indicators of the dependence between random variables. However, modeling dependency using statistical indicators is one thing, modeling it using a dependency function is another. This is the case of the copula function, which meets this objective and has the following advantages:

- The copula models the dependency structure from the distribution function and separate dependency and marginalized behavior.

- The multivariate distribution function carries more information than the different marginal distributions and this generally helps us to avoid the disadvantages of correlation as a measure of dependency.

- Instead of summarizing the dependency structure by a single scalar such as the coefficient of linear correlation, a model that reflects more detailed knowledge can be used.

- We have a wide range of copula families from which a model can be selected. This allows us to choose a particular family of copula according to the variables of the data we are trying to model.

- Copulas allow us to build non-Gaussian models, i.e. with a copula, we can construct for example a distribution with a Gaussian marginal and another uniform.

This explains the choice of the copula to model the dependency and not the correlation coefficient. In addition, the dependency indicators (linear correlation, Kendall's tau and Spearman's rho) can be defined in this framework from the parameters of the copula.

3.3 Copula models

We will present the main types of copula frequently used in practice, as well as their properties: elliptical copula, archimedean copula and polynomial copula. Then, we will introduce two methods for estimating the copula. The first method is non-parametric. The second approach is semi-parametric which assumes a parametric model for the copula and non-parametric for the marginal distributions. Then, we will explain the methods used to generate vectors, according to the various copulas. Finally, we will present a method to select the most suitable copula for the analysis appropriate for a certain set of data.

In literature, many copula models have been proposed. An important class of those models is the semiparametric one, which has been widely used in survival analysis where modelling and estimating the dependency structure between survival variables are fundamental problems. A semiparametric copula model is a collection of copulas $\{\mathbb{C}(\cdot;\theta);\ \theta \in \Theta \subset \mathbb{R}\}$, indexed by a parameter $\theta \in \Theta \subset \mathbb{R}$, with unknown nonparametric marginals.

3.3.1 Elliptical Copulas

The class of elliptical distributions provides a rich source of multivariate distributions which share many of the properties of the multivariate normal distribution and enables modelling of multivariate extremes and other forms of nonnormal dependences. Elliptical copulas are simply the copulas of elliptical distributions. Simulation from elliptical distributions is easy, and as a consequence of Sklar's Theorem so is simulation from elliptical copulas. Furthermore, we will show that rank correlation and tail dependence coefficients can be easily calculated. For further details on elliptical distributions we refer to Fang, Kotz and Ng (1987) and Cambanis, Huang and Simons (1981).

Elliptical distributions

Definition 3.3.1. If X is a n-dimensional random vector and, for some $\mu \in \mathbb{R}^n$ and some $nx \times n$ non-negative definite, symmetric matrix Σ, the characteristic function $\varphi_{X-\mu}(t)$ of $X-\mu$ is a function of the quadratic form $t^\top \Sigma t$, $\varphi_{X-\mu}(t) = \varphi(t^\top \Sigma t)$, we say that X has an elliptical distribution with parameters μ, Σ and φ, and we write $X \sim E_n(\mu, \Sigma, \varphi)$.

When n = 1, the class of elliptical distributions coincides with the class of one dimensional symmetric distributions. A function φ as in Definition above is called a characteristic generator.

Theorem 3.3.1. *$X \sim E_n(\mu, \Sigma, \varphi)$ with $rank(\Sigma) = k$ if and only if there exist a random variable $R \geq 0$ independent of U, a k-dimensional random vector uniformly distributed on the unit hypersphere $\{z \in \mathbb{R}^k | z^\top z = 1\}$, and an $n \times k$ matrix A with $AA^\top = \Sigma$, such that $X = \mu + RAU$.*

One practical problem with elliptical distributions in multivariate risk modelling is that all marginals are of the same type. To construct a realistic multivariate distribution for some given risks, it may be reasonable to choose a copula of an elliptical distribution but different types of marginals (not necessarily elliptical). One big drawback with such model is that the copula parameter R can no longer be estimated directly from the data. Recall that for non-degenerate elliptical distributions with finite variances, R is just the usual linear correlation matrix. In such cases, R can be estimated using (robust) linear correlation estimators. One such robust estimator is provided by the next theorem. For nondegenerate nonelliptical distributions with finite variances and elliptical copulas, R does not correspond to the linear correlation matrix. However, since the Kendall's tau rank correlation matrix for a random vector is invariant under strictly increasing transformations of the vector components, and the next theorem provides a relation between the Kendall's tau rank correlation matrix and R for nondegenerate elliptical distributions, R can in fact easily be estimated from data.

Theorem 3.3.2. *Let $X \sim E_n(\mu, \Sigma, \varphi)$ with $P(X_i = \mu_i) < 1$ and $P(X_j = \mu_j) < 1$. Then*

$$\tau(X_i, X_j) = (1 - \sum_{x \in \mathbb{R}} (P(X_i = x))^2) \frac{2}{\pi} arcsin(R_{ij}) \tag{3.12}$$

where the sum extends over all atoms of the distribution of X_i. If $rank(\Sigma) \geq 2$, then 3.12 simplifies to

$$\tau(X_i, X_j) = (1 - P(X_i = \mu_i)^2) \frac{2}{\pi} arcsin(R_{ij}) \tag{3.13}$$

Note that if $P(X_i = \mu_i) = 0$ for all i, which is true for, e.g., multivariate t-distribution or normal distributions with strictly positive definite dispersion matrices Σ, then

$$\tau(X_i, X_j) = \frac{2}{\pi} arcsin(R_{ij})$$

for all i and j .

The nonparametric estimator of R, $sin(\pi\hat{\tau}/2)$ (dropping the subscript for simplicity), provided by the above theorem, inherits the robustness properties of the Kendall's tau estimator and is an efficient (low variance) estimator of R for both elliptical distributions and nonelliptical distributions with elliptical copulas.

The Gaussian copula

The classical example of elliptic laws, it is the Gaussian distribution associated with the choice of

$$\phi(x) = c\, exp(-\frac{x}{2}), \tag{3.14}$$

with c is a normalization constant. The copula associated with the normal distribution is called the Gaussian copula or the normal copula, denoted $C_R^{Ga}(u)$ with R the linear correlation matrix and it is written as follows

$$C_R^{Ga}(u) = \phi_R^n(\phi^{-1}(u_1), ..., \phi^{-1}(u_n)) \tag{3.15}$$

where ϕ_R^n denotes the joint distribution function of the n-variate standard normal distribution function with linear correlation matrix R, and ϕ^{-1} denotes the inverse of the distribution function of the univariate standard normal distribution. Copulas of the above form are called Gaussian copulas. In the bivariate case the copula expression can be written as

$$C_R^{Ga}(u,v) = \int_{-\infty}^{\phi^{-1}(u)} \int_{-\infty}^{\phi^{-1}(v)} \frac{1}{2\pi\sqrt{1-R_{12}^2}} exp\{-\frac{s^2 - 2R_{12}^{st} + t^2}{2(1-R_{12}^2)}\}dsdt \tag{3.16}$$

Note that R_{12} is simply the usual linear correlation coefficient of the corresponding bivariate normal distribution. Since elliptical distributions are radially symmetric, the coefficient of upper and lower tail dependence are equal. Hence Gaussian copulas do not have lower tail dependence. We now address the question of random variate generation from the Gaussian copula C_R^{Ga}.

Student Copulas

Another example of elliptic distribution is the Student distribution with:

$$\varphi(z) = c\left(1 + \frac{z}{v}\right)^{-\frac{p+v}{2}} \tag{3.17}$$

where c is a normalization constant and $v \in \mathbb{R}$. The copula associated to the student multivariate distribution is called the student copula $\mathbb{C}_{St}(., \Sigma, v)$.

Let $Z = (Z_1, ..., Z_p)^\top \in \mathbb{R}^p$ be a random vector following the Student law with v the degree of liberty and $\Sigma = (\rho_{ij}), i, j = 1, ..., p$ the covariance matrix where ρ_{ij} is the linear correlation coefficient between Z_i and Z_j. Note that $\mathbb{F}_{St}(.)$ and $F_{St}(.)$ are respectively the distribution functions of Z and $Z_i, i = 1, ..., p$. The Student copula is defined as follows:

$$\begin{aligned} &\mathbb{C}_{St}(u_1, ..., u_p, \Sigma, v) := \mathbb{F}_{St}(F_{St}^{-1}(u_1), ..., F_{St}^{-1}(u_p)) \\ &= \int_{-\infty}^{F_{St}^{-1}(u_1)} ... \int_{-\infty}^{F_{St}^{-1}(u_p)} \frac{\Gamma(\frac{v+p}{2})}{(v\pi)^{\frac{v}{2}} det(\Sigma)^{\frac{1}{2}} \Gamma(\frac{v}{2})} \left(1 + \frac{1}{v} t^\top \Sigma^{-1} t\right)^{-\frac{v+p}{2}} dt_1 ... dt_p, \end{aligned} \tag{3.18}$$

where Γ is the gamma function and $t = (t_1, ..., t_p)^\top \in \mathbb{R}^p$.

The student copula density is of the following form

$$c_{St}(u_1, ..., u_p, \Sigma, v) := \frac{\Gamma(\frac{v+p}{2})[\Gamma(\frac{v}{2})]^{p-1}(1 + \frac{1}{v}\beta^\top\Gamma^{-1}\beta)^{-\frac{v+p}{2}}}{det(\Sigma)^{\frac{1}{2}} \left[\Gamma(\frac{v+1}{2})\right]^p \prod_{i=1}^p (1 + \frac{\beta_i^2}{v})^{-v+1}2} \tag{3.19}$$

where $\beta = (\beta_1, ..., beta_p)^\top = (F_{St}^{-1}(u_1), ..., F_{St}^{-1}(u_p))$ and $v > 0$.

3.3.2 Archimedean Copulas

This class of copula plays a very important role in the theory of parametric statistics. They allow the construction of a large variety of copula families, and thus to represent a wide variety of dependency structures. On the other hand, the copulas thus generated have closed analytical forms and are easy to simulate. Indeed, unlike elliptic copula, archimedean copula have the great advantage of describing very diverse dependency structures.

Let $\phi : [0,1] \mapsto [0,+\infty]$ be a convex, strictly decreasing and a continued function, such that $\phi(1) = 0$ and let $\phi^{(-1)}$ be the pseudo-inverse of ϕ. A copula is said to be Archimedean with the generator function ϕ if it is formulated as follows

$$\mathbb{C}_{Ar}(u_1, ..., u_p) := \phi^{(-1)}(\phi(u_1) + ... + \phi(u_p)), \forall u_1, ..., u_p \in [0,1] \tag{3.20}$$

The idea behind the archimedean copula with a generator ϕ is the transformation $\varphi(t) = exp(-\phi(u))$ applied to the marginals, making them independent.

$$\varphi(\mathbb{C}_{Ar}(u_1, ..., u_p)) = \prod_{i=1}^{p} \varphi(u_1). \tag{3.21}$$

Gumbel Family

We have the function $\phi(t) = (-lnt)^{\theta}$ with $\theta \geq 0$ and $t \in [0,1]$. This function is continued, convex, strictly decreasing and is defined from $[0,1]$ to $[0,+\infty]$, hence it is a generator.

The Gumbel copula is an archimedean copula where the generator function is the one defined above, hence, The Gumbel copula is written in the following form

$$\begin{aligned}\mathbb{C}_{Gum}(u_1, ..., u_p, \theta) &:= \phi^{(-1)}(\phi(u_1) + ... + \phi(u_p)) \\ &= exp(-[(-lnu_1)^{\theta} + ... + (-lnu_p)^{\theta}]^{\frac{1}{\theta}})\end{aligned} \tag{3.22}$$

Remark:

If $\theta = \theta_0 = 1$ the Gumbel copula is equivalent to the independent copula.

Frank Family

The generator of the Frank family copula is as follows:

$$\phi = -ln(\frac{exp(-\theta t) - 1}{exp(-\theta) - 1}) \tag{3.23}$$

where $t \in]0,1[$ and $\theta \in \Theta := \mathbb{R} \setminus \{0\}$. The Frank copula $\mathbb{C}_{Fr}(., \theta)$ is then of the form:

$$\mathbb{C}_{Fr}(u_1, ..., u_p, \theta) := -\frac{1}{\theta} ln(1 + \frac{\prod_{i=1}^{p}(exp(-\theta u_i) - 1)}{(exp(-\theta) - 1)^{p-1}}) \tag{3.24}$$

Remark:

If $\theta = \theta_0 = 0$ the Frank copula is equivalent to the independent copula.

Clayton Family

The Clayton family copula is an archimedean copula with the following generator function:

$$\phi(t) = \frac{t^{-\theta} - 1}{\theta} \tag{3.25}$$

with $\theta \in \Theta := [-1, +\infty[\setminus \{0\}$ and $t \in [0, 1]$. The Clayton copula $\mathbb{C}_{Cl}(., \theta)$ is of the following form:

$$\mathbb{C}_{Cl}(u_1, ..., u_p, \theta) := max\left[\left(\Sigma_{i=1}^{p} u_i^{-\theta} - p + 1\right), 0\right]^{-\frac{1}{\theta}} \tag{3.26}$$

Remark:

If $\theta = \theta_0 = 0$ the Frank copula is equivalent to the independent copula, hence:

$$\mathbb{C}_{Cl}(u, 0) = \mathbb{C}^{\perp}(u) \tag{3.27}$$

Ali-Mikhail-Haq Family

The Ali-Mikhail-Haq copula (AMH) belongs to the class of archimedean copula and for $\theta \in \Theta := [-1, 1]$ and $t \in [0, 1]$ its generator is given by

$$\phi(t) = ln\frac{1 - \theta(1 - t)}{t} \tag{3.28}$$

The AMH copula is therefore written in the following form

$$\mathbb{C}_{AMH}(u_1, ..., u_p, \theta) := \frac{\prod_{i=1}^{p} u_i}{1 - \theta(\prod_{i=1}^{p}(1 - u_i))} \tag{3.29}$$

Remark:

If $\theta = \theta_0 = 0$ the AMH copula is equivalent to the independent copula, hence:

$$\mathbb{C}_{AMH}(u, 0) = \mathbb{C}^{\perp}(u) \tag{3.30}$$

3.3.3 Polynomial Copulas

In this class, copulas are written as polynomials, for more details about this class, the reader may consult [120]. One of the most common families of polynomial copula used in practice is the Farlie-Gumbel-Morgenstern (FGM) family [118]. This copula has a very simple analytical expression, however, it remains limited due to the small range of degree of dependence.

The FGM copula is defined in the following manner

$$\mathbb{C}_{FGM}(u_1, ..., u_p, \theta) := \left[\prod_{i=1}^{p}\right]\left[1 + \prod_{i=1}^{p-1}\prod_{j=i+1}^{p} \theta(1 - u_i)(1 - u_j)\right], \theta \in \Theta := [-1, 1] \tag{3.31}$$

In addition, its density is given by

$$c_{FGM}(u_1, ..., u_p, \theta) := 1 + \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \theta(1-2u_i)(1-2u_j), \theta \in \Theta := [-1,1] \qquad (3.32)$$

Remark:

If $\theta = \theta_0 = 0$ the FGM copula is equivalent to the independent copula, hence:

$$\mathbb{C}_{FGM}(u, 0) = \mathbb{C}^{\perp}(u) \qquad (3.33)$$

3.4 Copula models summary

To conclude, in table 3.1, we give for the bivariate case, a description of the seven semiparametric copula models stated above which are the ones used in our simulation studies : the Gaussian copula, Student (T), Ali-Mikhail-Haq (AMH) [5], Clayton [34], Farlie-Gumbel-Morgenstern (FGM) [118], Frank [63] and Gumbel [74].

We provide, for each model, the corresponding parameter space Θ and the particular value θ_0 of the parameter θ corresponding to the independence hypothesis of marginals, i.e., the value θ_0 satisfying

$$\mathbb{C}\left(u_1, u_2; \theta_0\right) = \mathbb{C}_0\left(u_1, u_2\right) = u_1 u_2, \, \forall \left(u_1, u_2\right)^{\top} \in [0,1]^2. \qquad (3.34)$$

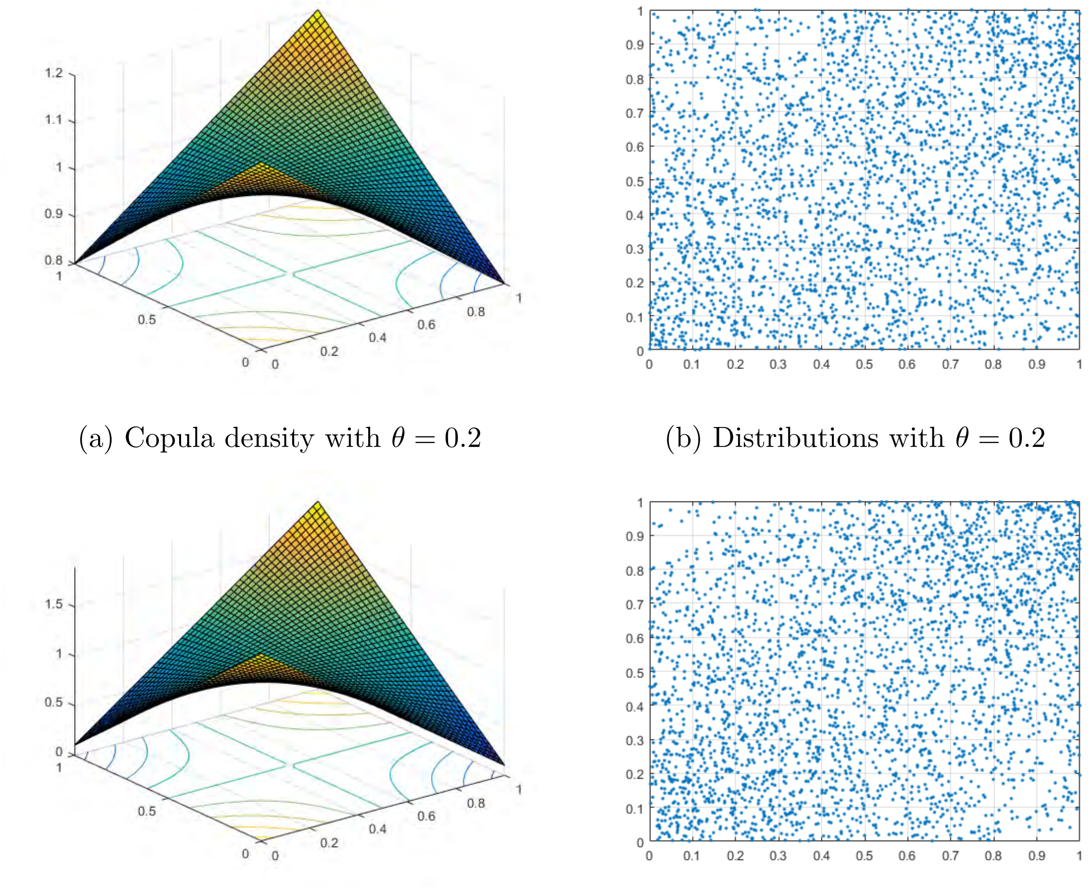
We give also, for each copula model $\mathbb{C}(\cdot; \theta)$, the corresponding theoritical Kendall's tau, see (3.10),

$$\tau_{\mathbb{C}}(\theta) := 4 \int_{[0,1]^2} \mathbb{C}(u_1, u_2; \theta) \, d\mathbb{C}(u_1, u_2; \theta) - 1, \, \forall \theta \in \Theta.$$

Figures 3.1-3.6 present the two-dimensional copulas densities for each model in Table 3.1 as well as the distribution of $N = 3000$ points generated according to each model for different values of the parameter θ. For more details on the commonly semiparametric used copulas, we can refer to [120] and [88].

	$\mathbb{C}(u_1, u_2; \theta)$	Θ	θ_0	$\tau_{\mathbb{C}}(\theta)$
Gaussian	$\frac{1}{2\pi\sqrt{1-\theta^2}} \int_{-\infty}^{\Phi^{-1}(u_1)} \int_{-\infty}^{\Phi^{-1}(u_2)} e^{\frac{s^2-2\theta st+t^2}{2(1-\theta^2)}} \, dtds$	$]-1,1[$	0	$\frac{2}{\pi} \arcsin(\theta) \in]-1,1[$
T	$\frac{1}{\nu\pi\sqrt{1-\theta^2}} \int_{-\infty}^{\psi^{-1}(u_1)} \int_{-\infty}^{\psi^{-1}(u_2)} \frac{\Gamma(\frac{\nu+2}{2})}{\Gamma(\frac{\nu}{2})}$ $\left[1+\frac{t^2-2\theta ts+s^2}{\nu(1-\theta^2)}\right]^{-\frac{\nu+2}{2}} dtds$	$]-1,1[$	0	$\frac{2}{\pi} \arcsin(\theta) \in]-1,1[$
FGM	$u_1u_2 + \theta u_1 u_2 (1-u_1)(1-u_2)$	$[-1,1]$	0	$\frac{2\theta}{9} \in \left[-\frac{2}{9}, \frac{2}{9}\right]$
AMH	$\frac{u_1u_2}{1-\theta(1-u_1)(1-u_2)}$	$[-1,1]$	0	$\frac{3\theta^2-2(\theta-(1-\theta)^2\ln(1-\theta))}{3\theta^2}$ $\in \left[\frac{5-8\ln 2}{3}, \frac{1}{3}\right]$
Clayton	$\max\left(u_1^{-\theta} + u_2^{-\theta} - 1, 0\right)^{-1/\theta}$	$[-1,+\infty[\setminus\{0\}$	0	$\frac{\theta}{\theta+2} \in [-1,1]$
Frank	$-\frac{1}{\theta}\ln\left(1-\frac{\left(1-e^{-\theta u_1}\right)\left(1-e^{-\theta u_2}\right)}{(1-e^{-\theta})}\right)$	$\mathbb{R}\setminus\{0\}$	0	$\frac{\theta-4[1-D(\theta)]}{\theta} \in [-1,1]$
Gumbel	$e^{\left(-\left[(-\ln u_1)^\theta+(-\ln u_2)^\theta\right]^{1/\theta}\right)}$	$[1,+\infty[$	1	$\frac{\theta-1}{\theta} \in [0,1[$

Table 3.1: Examples of semiparametric copula models $\{\mathbb{C}(\cdot;\theta);\, \theta \in \Theta \subset \mathbb{R}\}$. $\Phi :$ the standard normal distribution function, $\psi :$ the standard student distribution function, $\nu :$ the number of degrees of freedom, $\Gamma :$ Gamma function, and $D : x \in \mathbb{R}_+^* \mapsto D(x) := \frac{1}{x}\int_0^x \frac{t}{e^t-1}\, dt$ is the Debye function.



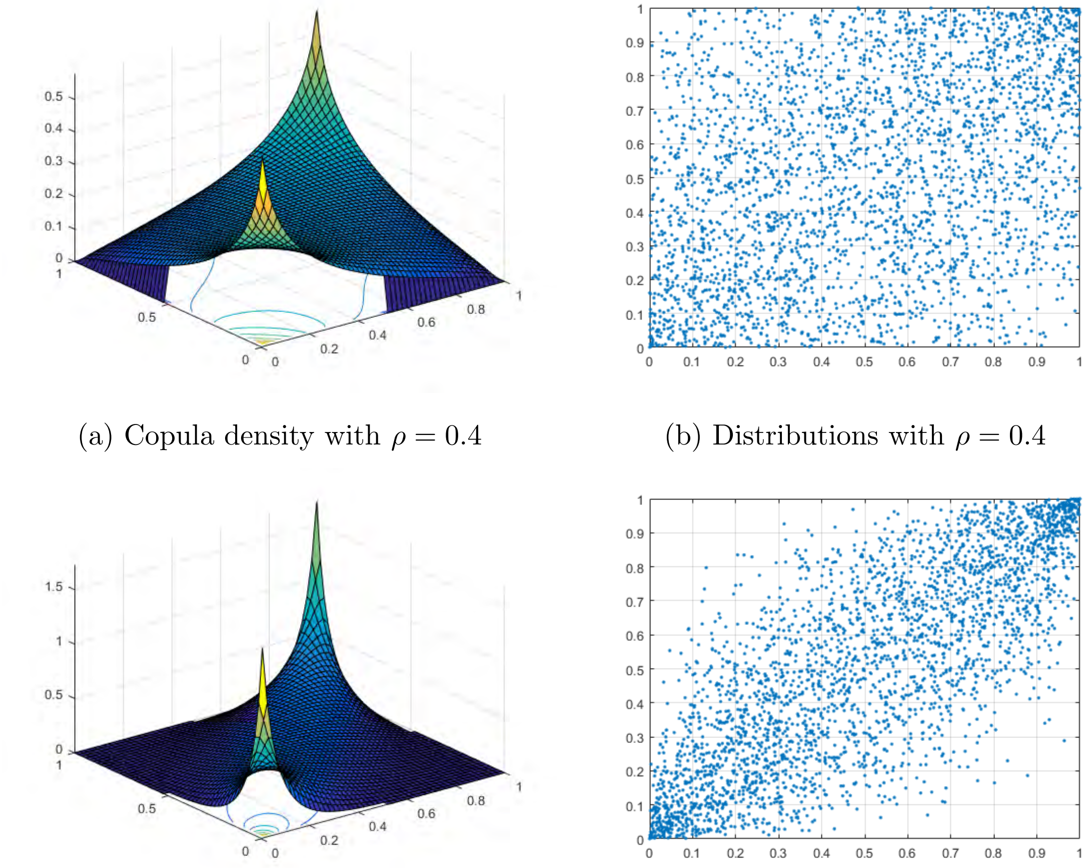
(a) Copula density with $\theta = 0.2$

(b) Distributions with $\theta = 0.2$

(c) Copula density with $\theta = 0.8$

(d) Distributions with $\theta = 0.8$

Figure 3.1: FGM copula



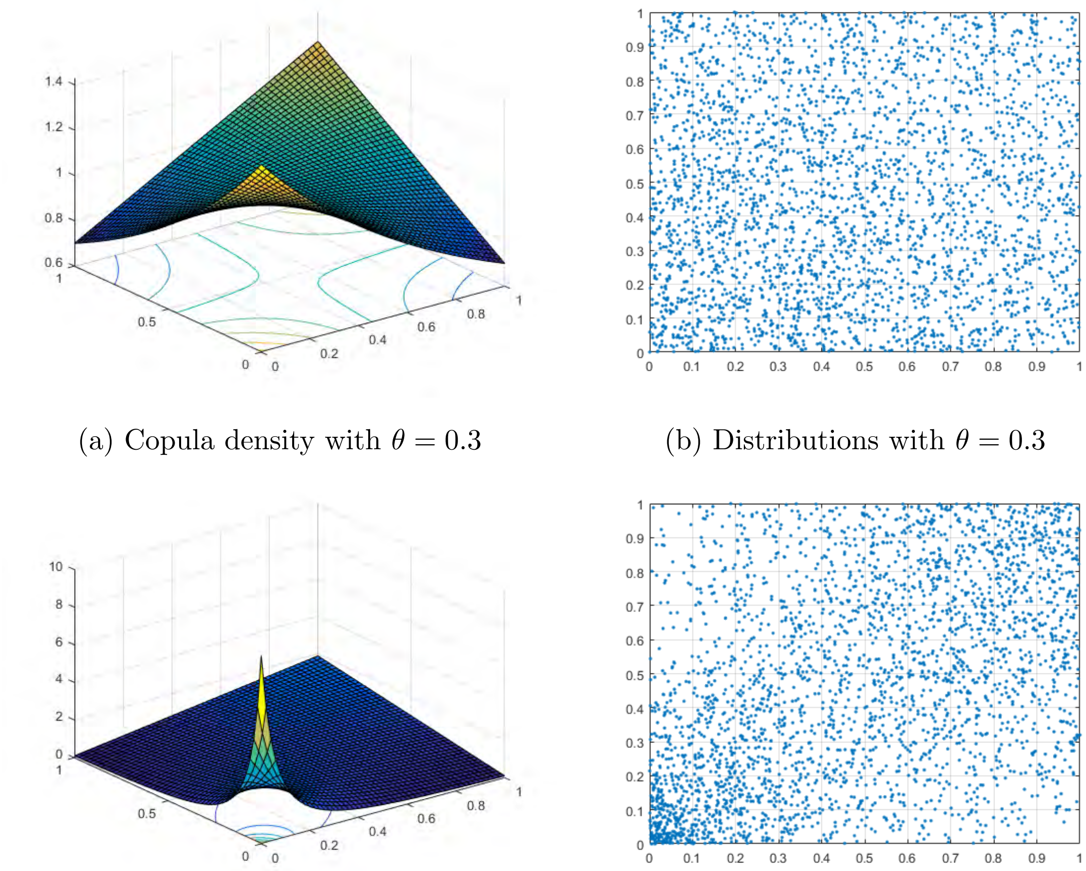
(a) Copula density with $\rho = 0.4$

(b) Distributions with $\rho = 0.4$

(c) Copula density with $\rho = 0.8$

(d) Distributions with $\rho = 0.8$

Figure 3.2: Gaussian copula, ρ : correlation coefficient



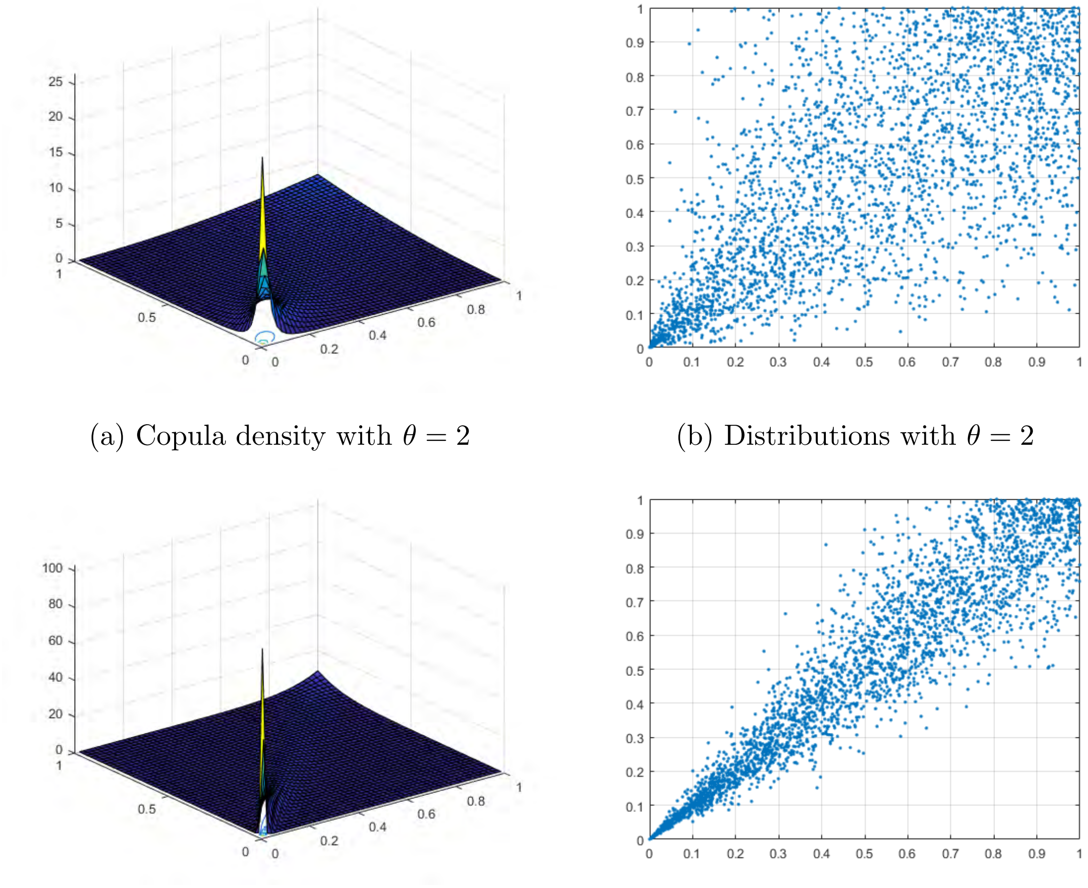
(a) Copula density with $\theta = 0.3$

(b) Distributions with $\theta = 0.3$

(c) Copula density with $\theta = 0.9$

(d) Distributions with $\theta = 0.9$

Figure 3.3: AMH copula



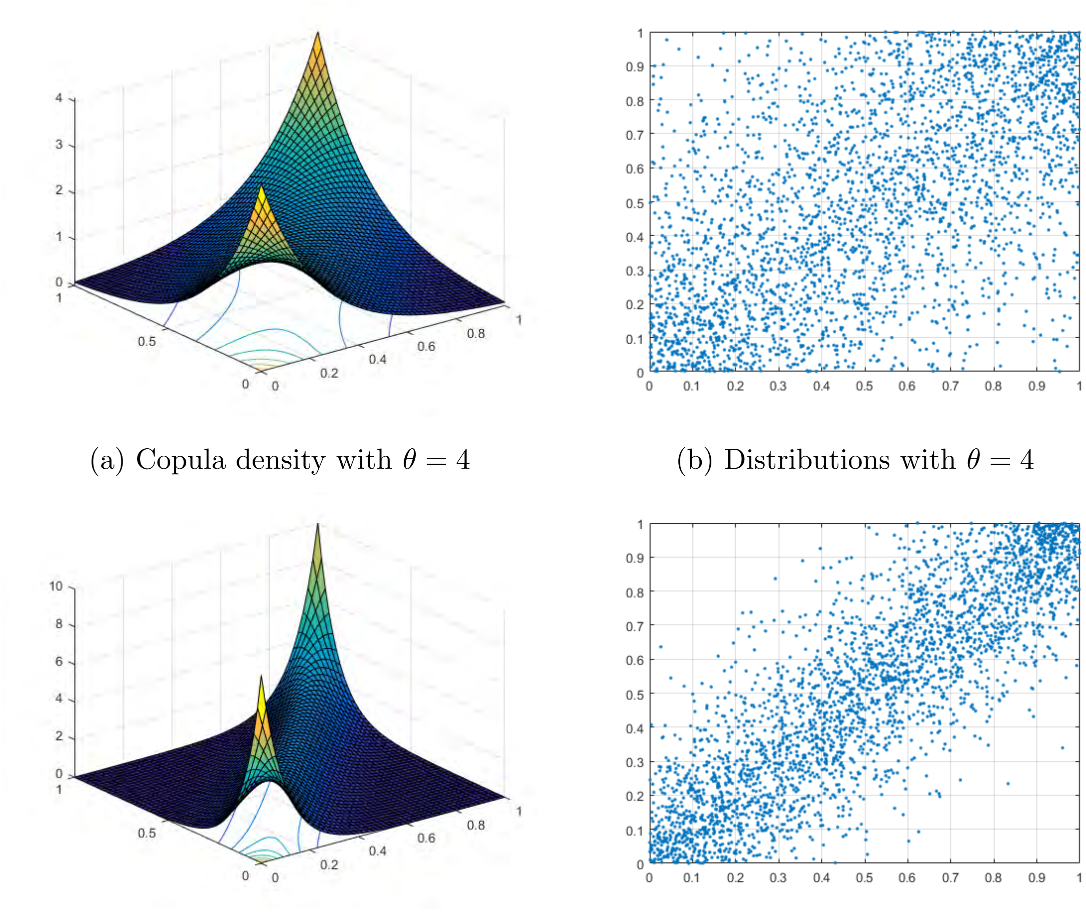
(a) Copula density with $\theta = 2$

(b) Distributions with $\theta = 2$

(c) Copula density with $\theta = 8$

(d) Distributions with $\theta = 8$

Figure 3.4: Clayton copula



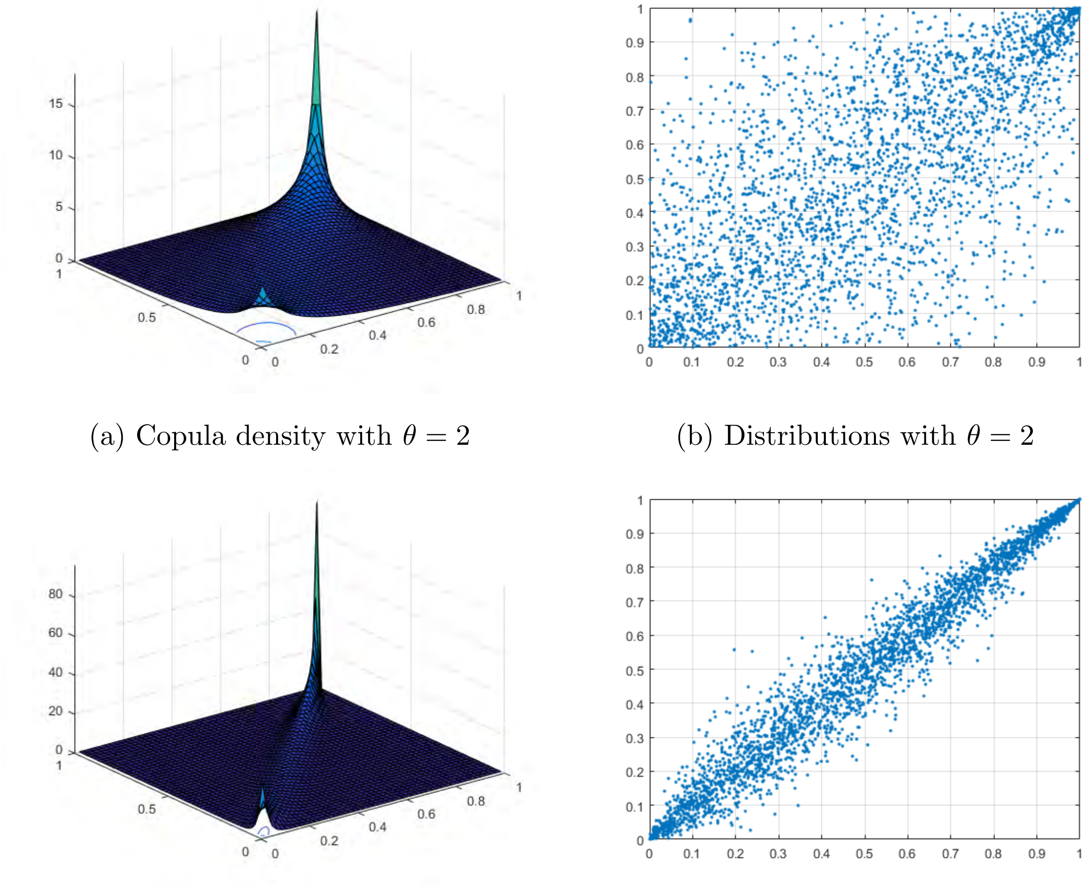
(a) Copula density with $\theta = 4$

(b) Distributions with $\theta = 4$

(c) Copula density with $\theta = 10$

(d) Distributions with $\theta = 10$

Figure 3.5: Frank copula



(a) Copula density with $\theta = 2$

(b) Distributions with $\theta = 2$

(c) Copula density with $\theta = 10$

(d) Distributions with $\theta = 10$

Figure 3.6: Gumbel copula

3.5 Copula estimation

The copula has the advantage of completely modeling the dependence between random variables. In practice, however, the copula is unknown, hence the need to estimate it. The proposed Approaches for the estimation of a copula are divided into parametric approaches, semi-parametric and non-parametric. The parametric approach first proposed by Genest et al [68], assumes a specific model for the copula and distributions marginal at the same time. Usually we use the method of the maximum of complete likelihood (FML) to obtain parameter estimators [22, 88]. We can jointly estimate the parameters of the marginal distributions and the parameters of the of copules. One disadvantage of this method is that it requires intensive calculations and sometimes the optimization problem is difficult to solve. In [89] , Joe and Xu proposed a two-step procedure called the Margin Inference Function (MIF). First the parameters of the marginals are estimated, then the copula parameters are estimated using the method of maximum likelihood. As with MLF, the disadvantage of this method is that it depends on marginal distribution assumptions [90]. This approach is widely used in practice because of its simplicity.

Then, the semi-parametric approach, imposes a parametric model for the copula, and non-parametric for marginal distributions. For example the CML method (Canonical Maximum Likelihood) is a semiparametric approach, where we estimate the distributions marginal by the empirical distribution, and we use the maximum-of-average likelihood method to estimate the vector of the parameters of the parametric copula. This approach is studied in [68, 124]. In [97], the authors compare semi-parametric and parametric methods for copula estimation.

Finally, the non-parametric approach takes into account both the non-parametric models for the copula and for the marginal distributions. [46] suggests the empirical multidimensional distribution to estimate the copula function. In [71], the authors estimate a copula using a method based on kernel smoothing. For i.i.d data, in [135], the authors develop an estimator based on Bernstein polynomials.

In what follows, we present the two main methods of copula estimation that we will use for our approaches. We begin with the kernel methods in the non-parametric case to estimate a copula, then the Canonical Maximum Likelihood method in the semi-parametric case to estimate the parameter of a copula.

3.5.1 Non-parametric estimation (Kernel methods)

Kernel estimation [136] is a non-parametric approach to estimation. It is based on a sample of a statistical population. Let $Z \in \mathbb{R}^p$ be a random vector and $\mathbb{F}_Z(.)$ a distribution function, $F_{Z_i}(.), i = 1, ..., p$ the marginals and $\mathbb{C}_Z(.)$ the copula, such as

$$\mathbb{F}_Z(z_1, ..., z_p) := \mathbb{C}_Z(F_{Z_1}(z_1), ..., F_{Z_p}(z_p)) \tag{3.35}$$

Suppose that we have N i.i.d. of Z, denoted $Z(1), ..., Z(N)$. To build the kernel estimator we must introduce kernels, i.e., symmetrical, integrable k_i functions defined from $\mathbb{R}$ into $\mathbb{R}^+$ such that

$$\int_{\mathbb{R}} k_i(z)dz = 1, ..., p \tag{3.36}$$

we have the p-dimensional kernel

$$k(z) = \prod_{i=1}^{p} k_i(x_i) \tag{3.37}$$

and its distribution function

$$K(z) := \prod_{i=1}^{p} \int_{-\infty}^{z_i} k_i(x_i)dx_i := \prod_{i=1}^{p} K_i(z_i) \tag{3.38}$$

In the general case, we can consider

$$k(z,h) = \prod_{i=1}^{p} k_i(\frac{z_i}{h_i}) \quad and \quad K(z,h) = \prod_{i=1}^{p} K_i(\frac{z_i}{h_i}) \tag{3.39}$$

where h is a diagonal matrix, with elements h_i, $i = 1, ..., n$ and a determinant denoted $|h|$ otherwise called the smoothing window. The individual smoothing windows h_i are functions of N number of samples, such that :

$$h_i \to 0 \quad N \to +\infty \tag{3.40}$$

The kernel estimator, named $\widehat{f_{Z_i}}$ of the density f_{Z_i} in the point z_i is defined as follows

$$\widehat{f_{Z_i}}(z_i) := \frac{1}{Nh_i} \sum_{n=1}^{N} k_i(\frac{z_i - Z_i(n)}{h_i}), i = 1, ..., p \tag{3.41}$$

The kernel estimator, named $\widehat{f_Z}$ of the density f_Z in the point $z = (z_1, ..., z_p)$ is given by:

$$\begin{aligned} \widehat{f_Z}(z) &:= \frac{1}{N|h|} \sum_{n=1}^{N} k(z - Z(n), h) \\ &= \frac{1}{\prod_{i=1}^{p} h_i} \sum_{n=1}^{N} \prod_{i=1}^{p} k_i(\frac{z_i - Z_i(n)}{h_i}) \end{aligned} \tag{3.42}$$

Therefore, an estimator of the marginal distribution function F_{Z_i} , at point z_i, can be defined by

$$\widehat{F}_{Z_i}(z_i) := \frac{1}{N} \sum_{n=1}^{N} K_I(\frac{z_i - Z_i(n)}{h_i}) \tag{3.43}$$

and the estimator of the joint distribution function $\mathbb{F}_Z$ in the point $z = (z_1, ..., z_p)$ will be defined as:

$$\begin{aligned}\widehat{\mathbb{F}}_Z(z) &:= \frac{1}{N}\sum_{n=1}^{N} K(z - Z(n), h)\\ &= \frac{1}{N}\sum_{n=1}^{N}\prod_{i=1}^{p} K_i(\frac{z_i - Z_i(n)}{h_i})\end{aligned} \tag{3.44}$$

Suppose that the copula $\mathbb{C}_Z(.)$ is p-time continuously derivable and therefore it admits a density, denoted $c_Z(.)$. We notice that this density represents the joint density of the vector $U = (U_1, ..., U_p) = (F_{Z_1}(Z_1), ..., F_{Z_p}(Z_p))$ By applying the same kind of arguments above, a simple non-parametric estimator of the density $c_Z(.)$ could have the form of a kernel estimator, given by

$$\begin{aligned}\widehat{c}_Z(u) &:= \frac{1}{N|h|}\sum_{n=1}^{N} k(u - \widehat{U}(n), H)\\ &= \frac{1}{N\prod_{i=1}^{p} H_i}\sum_{n=1}^{N}\prod_{i=1}^{p} k_i(\frac{u_i - \widehat{U}_i(n)}{H_i})\end{aligned} \tag{3.45}$$

where $\widehat{U} = (\widehat{U}_i(1), ..., \widehat{U}_i(N)) = (\widehat{F}_{Z_i}(Z_i(1)), ..., \widehat{F}_{Z_i}(Z_i(N)))$ and H designates the matrix of the smoothing window.
The copula $\mathbb{C}_Z$ estimator on point $u = (u_1, ..., u_p)^\top$ is given by

$$\begin{aligned}\widehat{\mathbb{C}}_Z(u) &:= \frac{1}{N}\sum_{n=1}^{N} K(u - \widehat{U}(n), H)\\ &= \frac{1}{N}\sum_{n=1}^{N}\prod_{i=1}^{p} K_i(\frac{u_i - \widehat{U}_i(n)}{H_i})\end{aligned} \tag{3.46}$$

In practice, we usually chose h_i and H_i according to Silverman's [Sil86] rule, i.e., for all i = 1, ... , p, we have

$$\begin{aligned}h_i &= (\frac{4}{3})^{\frac{1}{5}} N^{\frac{-1}{5}} \widehat{\sigma}_i,\\ H_i &= (\frac{4}{p+2})^{\frac{1}{p+4}} N^{\frac{-1}{p+4}} \widehat{\Sigma}_i,\end{aligned}$$

where $\widehat{\sigma}_i$ is the empirical standard deviation of the sample $Z_i(1), ..., Z_i(N)$, and $\widehat{\Sigma}_i$ is the empirical standard deviation of $(\widehat{U}_i(1), ..., \widehat{U}_i(N)) = (\widehat{F}_Z)$
Very often, k_i functions are chosen as the standard Gaussian density (null expectation and unit variance), i.e., for all i = 1, ... , p,

$$k_i(z) = k(z) := \frac{1}{\sqrt{2\pi}} exp(-\frac{1}{2}z^2) \tag{3.47}$$

The study of the kernel estimator has been considered by various authors, for more details we can consult [59, 71, 125, 155] .

3.5.2 Semi-parametric estimation (CML method)

Let $Z \in \mathbb{R}^p$ be a random vector of a distribution function $\widehat{F}_Z(.)$, marginal functions $F_{Z_i}(.), i = 1, ...p$ and a copula $\widehat{C}_Z$. The semi-parametric approach imposes a parametric model for the copula, $\widehat{C}_Z = \widehat{C}_Z(., \theta)$, of parameter $\theta \in \Theta \subset \mathbb{R}^p$, and non-parametric for marginal distributions. The semi-parametric estimation method for estimating the parameters of a copula is also known as the pseudo maximum likelihood or canonical maximum likelihood. Consider N i.i.d. samples of Z, denoted $Z(1), ..., Z(N)$, we describe the method as follows

- 1. Estimate the marginals using the kernel method

$$\widehat{F}_{Z_i}(z_i) := \frac{1}{N}\sum_{n=1}^{N} K_i(\frac{z_i - Z_i(n)}{h_i}) \tag{3.48}$$

  where K_i designates the kernel primitive k_i and h_i is the smoothing window.

- 2. Estimate the parametric copula by the maximum likelihood method. The procedure consists in selecting the value of the parameter that maximizes the pseudo log-likelihood.

$$\ell(\theta) := \frac{1}{N}\sum_{n=1}^{N} log \left[c_z(\widehat{F}_{Z_1}(Z_1(n)), ..., \widehat{F}_{Z_p}(Z_p(n)), \theta) \right] \tag{3.49}$$

  where $c_Z(., \theta)$ is the copula density, and $\widehat{F}_{Z_j}(.)$ is an estimate of the marginal distribution function $F_{Z_j}(.)$ of the random variable Z_j. The maximum log-likelihood is therefore given by

$$\widehat{\theta} := arg_{\theta \in \Theta} max\, \ell(\theta) \tag{3.50}$$

  where $\Theta \subset \mathbb{R}^d$ is the parameter space.

  For more details the reader can refer to [68, 151].

3.6 Copula model selection and parameter estimation

In the following lines, we briefly expose some procedures for copula model selection as well as method of estimating the parameter θ from the data. Let $\boldsymbol{S} := (S_1, \ldots, S_p)^\top \in \mathbb{R}^p, p \geq 2$, be a random vector with unknown copula $\mathbb{C}_{\mathbf{S}}(\cdot)$. We assume in this Subsection that training samples of $\boldsymbol{S}$ are available, that is, we dispose of N i.i.d. realizations $\boldsymbol{s}(1), \ldots, \boldsymbol{s}(N)$ of $\boldsymbol{S}$. The aim is to select, using these samples, the "best" copula model, among a list of candidates, modeling the dependency structure of the components of $\boldsymbol{S}$, and to estimate the associate parameter θ of the selected model. Assume that $\mathbb{C}_{\mathbf{S}}(\cdot)$ can be described through a model to be selected from a set of candidates

$$M_l := \{\mathbb{C}_l(\cdot;\theta_l);\ \theta_l \in \Theta_l \subset \mathbb{R}^{d_l}\}, \quad l = 1, \ldots, L. \tag{3.51}$$

Each M_l is parameterized by θ_l, a vector of the parameter space Θ_l, a subset of $\mathbb{R}^{d_l}$. Model selection can be made using the classical Akaike information criterion (AIC) [3] or Bayesian information criterion (BIC) [137], derived from the semiparametric log-likelihood, see e.g. [68, 151]. For all $l = 1, \ldots, L$, we denote by $\boldsymbol{c}_l(\cdot;\theta_l)$ the density of the copula $\mathbb{C}_l(\cdot;\theta_l)$; see (3.6).

The AIC, of a given model M_l, is defined by

$$AIC(l) := -2 \sup_{\theta_l \in \Theta_l} \sum_{n=1}^{N} \log\left(\boldsymbol{c}_l\left(\widehat{F}_{S_1}(s_1(n)), \ldots, \widehat{F}_{S_p}(s_p(n)); \theta_l\right)\right) + 2\, d_l, \tag{3.52}$$

where d_l is the dimension of the parameter space Θ_l, and $\widehat{F}_{S_i}(\cdot)$ is the "rescaled" empirical distribution function of the random variable (the component) S_i, $i = 1 \ldots, p$, defined by

$$\forall x \in \mathbb{R},\ \widehat{F}_{S_i}(x) := \frac{1}{N+1} \sum_{n=1}^{N} \mathbb{1}_{]-\infty,x]}(s_i(n)).$$

The rescaling by $N+1$, instead of N, is often used in order to avoid difficulties of possible unboundedness of the considered copula density $\boldsymbol{c}_l(\boldsymbol{u}, \theta_l)$ when u_i tends to 1. The BIC, of a given model M_l, is defined by

$$BIC(l) := -2 \sup_{\theta_l \in \Theta_l} \sum_{n=1}^{N} \log\left(\boldsymbol{c}_l\left(\widehat{F}_{S_1}(s_1(n)), \ldots, \widehat{F}_{S_p}(s_p(n)); \theta_l\right)\right) + \log(N)\, d_l. \tag{3.53}$$

The optimal model in term of AIC (respectively, BIC) is the one minimizing the AIC (respectively, BIC) value, namely, the density copula model $\left\{\boldsymbol{c}_{l^*}(\cdot;\theta_{l^*});\ \theta_{l^*} \in \Theta_{l^*} \subset \mathbb{R}^{d_{l^*}}\right\}$, where

$$l^* = \arg \min_{l \in \{1,\ldots,L\}} AIC(l) \quad (\text{respectively, } l^* = \arg \min_{l \in \{1,\ldots,L\}} BIC(l)).$$

Denote, for simplicity, $\left\{\boldsymbol{c}_{\mathbf{S}}(\cdot,\theta); \theta \in \Theta \subset \mathbb{R}^d\right\}$ instead of $\left\{\boldsymbol{c}_{l^*}(\cdot;\theta_{l^*});\ \theta_{l^*} \in \Theta_{l^*} \subset \mathbb{R}^{d_{l^*}}\right\}$, the selected model according to one of the above procedures. Following [68, 151], the parameter θ of the considered copula model can be estimated by maximizing the semiparametric log-likelihood

$$\overline{\theta} := \arg\sup_{\theta \in \Theta} \sum_{n=1}^{N} \log\left(\boldsymbol{c}_{\mathbf{S}}\left(\widehat{F}_{S_1}(s_1(n)), \ldots, \widehat{F}_{S_p}(s_p(n)); \theta\right)\right). \tag{3.54}$$

3.7 Conclusion

In this chapter, we have given recalls on copulas. We have examined some of semiparametric copula models by presenting the associated copula density and the distributions of points. Then, we have shown how we can select the best copula model among a list of candidate copula models and finally how we can estimate the associated parameter of the chosen copula model.

Chapter 4

Introduction to Alpha-divergence

4.1 Introduction

The choice of an appropriate measure of distance has always been a study subject due to its importance for many applications of probability theory. In this chapter we go into details about divergences giving extra attention to the alpha-divergence due to its particular importance, applicability and generality of multiple other known and widely used divergences such as the Kullback-Leibler divergence and the Hellinger divergence.

4.2 Divergences

The idea of divergences, which in some sense assess how 'close' two probability distributions are from one another, has been widely employed in probability, statistics, information theory, and related fields. However divergence is a weaker notion than that of the distance, in particular the divergence need not be symmetric (that is, in general the divergence from p to q is not equal to the divergence from q to p), and need not satisfy the triangle inequality.

Suppose S is a space of all probability distributions with common support. Then a divergence on S is a function $D(. \,||\, .) : S \times S \to R$ satisfying

1. $D(p \,||\, q) \geq 0$ for all $p, q \in S$,
2. $D(p \,||\, q) = 0$ if and only if $p = q$

The two most important divergences are the relative entropy (Kullback–Leibler divergence, KL divergence), which is central to information theory and statistics, and the squared Euclidean distance (SED). Minimizing these two divergences is the main way that linear inverse problem are solved, via the principle of maximum entropy and least squares, notably in logistic regression and linear regression.

The two most important classes of divergences are the f-divergences and Bregman divergences; however, other types of divergence functions are also encountered in the literature. The only divergence that is both an f-divergence and a Bregman divergence is the Kullback–Leibler divergence; the squared Euclidean divergence is a Bregman divergence.

We will talk in the following about these two families of divergences but giving more details and insights on the divergences that interest us which are the Kullback-Leibler divergence and the Alpha divergence.

4.2.1 f-divergences

Let $f : (0, \infty) \to \mathbb{R}$ be a convex function with $f(1) = 0$. Let P and Q be two probability distributions on a measurable space (χ, F). If $P \ll Q$ then the f-divergence is defined as

$$D_f(P \parallel Q) \triangleq \mathbb{E}_Q \left[f \left(\frac{dP}{dQ} \right) \right] \tag{4.1}$$

where $\frac{dP}{dQ}$ is a Radon-Nikodyn derivative and $f(0) \triangleq f(0+)$. More generally, let $f'(\infty) \triangleq lim_{x\to 0} x f(1/x)$. Suppose that $Q(dx) = q(x)\mu(dx)$ and $P(dx) = p(x)\mu(dx)$ for some common dominating μ, then we have

$$D_f(P \parallel Q) = \int_{q>0} q(x) f \left(\frac{p(x)}{q(x)} \right) d\mu + f'(\infty) P[q = 0] \tag{4.2}$$

with the agreement that if $P[q = 0] = 0$ the last term is taken to be zero regardless of the value of $f'(\infty)$ (which could be infinite).

For the discrete case, with $Q(x)$ and $P(x)$ being the respective probabilities distribution functions, we can also write

$$D_f(P \parallel Q) = \sum_x Q(x) f \left(\frac{P(x)}{Q(x)} \right) \tag{4.3}$$

with the understanding that

- $f(0) = f(0+)$,
- $0f(\frac{0}{0}) = 0$, and
- $0f(\frac{\alpha}{0}) = lim_{x\to 0} x f(\frac{a}{x}) = af'(\infty) for\, a > 0$

f-divergences have been introduced in the sixties [6, 44] and then again in the seventies [3]. Kullback-Leibler divergence, Hellinger distance, χ^2 divergence, Csiszar α-divergence, Squared Hellinger distance, Le Cam distance, Jensen-Shannon divergence and Kolmogorov total variation distance are some well known instances of f-divergences. Other instances may be found in [98, 99, 109].

f-divergences can usefully play the role of surrogate functions, that are functions majorizing or minorizing the objective or the risk functions at hand. For example, f-divergences are used for defining loss functions that yield Bayes consistency for joint estimation of the discriminant function and the quantizer in [121], as surrogate functions for independence and ICA in [116], and the α-divergence is used in [115] as a surrogate function for maximizing a likelihood in an EM-type algorithm.

Bounds on the minimax risk in multiple hypothesis testing and estimation problems are expressed in terms of the f-divergences in [18, 75], respectively.

f-divergences, used as general (entropic) distance-like functions, allow a non-smooth non-convex optimization formulation of the partitioning clustering problem, namely the problem of clustering with a known number of classes, for which a generic iterative scheme keeping the simplicity of the k-means algorithm is proposed in [143, 144].

Nonnegative matrix factorization (NMF), of widespread use in multivariate analysis and linear algebra [52], is another topic that can be addressed with f-divergences. For example, NMF is achieved with the aid of Kullback divergence and alternating minimization in [60], Itakura-Saito divergence in [61], f-divergences in [30], or alpha-divergences in [31, 100]; see also [32].

4.2.2 Bergman divergences

Bregman divergences, are defined for vectors, matrices, functions and probability distributions. The Bregman divergence between vectors is defined as:

$$D_\varphi(x,y) = \varphi(x) - \varphi(y) - (x-y)^\top \nabla\varphi(y) \tag{4.4}$$

with φ a differentiable strictly convex function $\mathbb{R}^d \to \mathbb{R}$. The symmetrized Bregman divergence writes:

$$\bar{D}_\varphi(x,y) = (\nabla\varphi(x) - \nabla\varphi(y))^\top (x-y) \tag{4.5}$$

The Bregman matrix divergence is defined as:

$$D_\phi(X,Y) = \phi(X) - \phi(Y) - Tr((\nabla\phi(Y))^\top (X-Y)) \tag{4.6}$$

for X, Y real symmetric dxd matrices, and ϕ a differentiable strictly convex function $\mathbb{S}^d \to \mathbb{R}$.

For $\phi(X) = ln \mid X \mid$, the divergence (4.6) is identical to the distance between two positive matrices defined as the Kullback-Leibler divergence between two Gaussian distributions having those matrices as covariance matrices.

The Bregman divergence between probability densities is defined as

$$D_\varphi(p,q) = \int (\varphi(p) - \varphi(q) - (p-q)\varphi'(q))dx \tag{4.7}$$

One important instance of the use of Bregman divergences for learning is the case of inverse problems [103] , where convex duality is extensively used. Convex duality is also used for minimizing a class of Bregman divergences subject to linear constraints in [47], whereas a simpler proof using convex analysis is provided in [16], and the results are used in [35].

Mixture models are estimated using Bregman divergences within a modified EM algorithm in [65]. Learning continuous latent variable models with Bregman divergences and alternating minimization is addressed in [157]. The use of Bregman divergences as surrogate loss functions for the design of minimization algorithms for learning that yield guaranteed convergence rates under weak assumptions is discussed in [122]. Learning structured models, such as Markov networks or combinatorial models, is performed in [142] using large-margin methods, convex-concave saddle point problem formulation and dual extra-gradient algorithm based on Bregman projections. Proximal minimization schemes handling Bregman divergences that achieve message passing for graph-structured linear programs (computation of MAP configurations in Markov random fields) are investigated in [131].

There are also many instances of application of Bregman divergences for solving clustering problems. Used as general (entropic) distance-like functions, they allow a non-smooth non-convex optimization formulation of the partitioning clustering problem, for which a generic iterative scheme keeping the simplicity of the k-means algorithm is proposed in [152]. Clustering with Bregman divergences unifying k-means, LBG and other information theoretic clustering approaches is investigated in [14], together with a connection with rate distortion theory.

Nonnegative matrix approximation with low rank matrices is discussed in [50], whereas matrix approximation based on the minimum Bregman information principle (generalization to all Bregman loss functions of MaxEnt and LS principles) is the topic of [13]. Nonnegative matrix factorization (NMF) with Bregman divergences is addressed in [31]; see also [32]. The particular case of using the density power divergence and a surrogate function for NMF is investigated in [62].

4.3 The Kullback-Leibler divergence

To measure the difference between two probability distributions over the same variable x, a measure, called the Kullback-Leibler divergence, or simply, the KL divergence, has been popularly used. The concept was originated in probability theory and information theory.

The KL divergence, which is closely related to relative entropy, information divergence, and information for discrimination, is a non-symmetric measure of the difference between two probability distributions $p(x)$ and $q(x)$. Specifically, the Kullback-Leibler (KL) divergence of $q(x)$ from $p(x)$, denoted $D_{KL}(p(x), q(x))$, is a measure of the information lost when $q(x)$ is used to approximate $p(x)$.

Let $p(x)$ and $q(x)$ are two probability distributions of a discrete random variable x. That is, both $p(x)$ and $q(x)$ sum up to 1, and $p(x) > 0$ and $q(x) > 0$ for any x in X. $D_{KL}(p(x), q(x))$ is defined in Equation (4.8):

$$D_{KL}(p(x) \mid\mid q(x)) = \sum_{x \in X} p(x) ln \frac{p(x)}{q(x)} \tag{4.8}$$

The KL-divergence measures the expected number of extra bits required to code samples from $p(x)$ when using a code based on q(x), rather than using a code based on $p(x)$. Typically $p(x)$ represents the 'true' distribution of data, observations, or a precisely calculated theoretical distribution. The measure $q(x)$ typically represents a theory, model, description, or approximation of $p(x)$. The continuous version of the KL-divergence is

$$D_{KL}(p(x) \mid\mid q(x)) = \int_{-\infty}^{\infty} p(x) ln \frac{p(x)}{q(x)} dx \tag{4.9}$$

Although the KL divergence measures the 'distance' between two distributions, it is not a distance measure. This is because that the KL divergence is not a metric measure. It is not symmetric: the KL from $p(x)$ to $q(x)$ is generally not the same as the KL divergence from $q(x)$ to $p(x)$. Furthermore, it need not to satisfy triangular inequality. Nevertheless, $D_{KL}(P \mid\mid Q)$ is a non-negative measure. $D_{KL}(P \mid\mid Q) \geq 0$ and $D_{KL}(P \mid\mid Q) = 0$ if and only if $P = Q$.

Having defined KL-divergence, we may now describe the information content between two random variables X and Y. The mutual information $I(X, Y)$ between X and Y is the KL-divergence between their joint distribution and their products (marginal) distributions. More mathematically,

$$I(X, Y) = \sum_{x,y} p(x, y) log \frac{p(x, y)}{p(x)p(y)} \tag{4.10}$$

We can rewrite this in several ways. First, using Bayes' rule, we have $p(x, y)/p(y) = p(x \mid y)$, so

$$\begin{aligned} I(X, Y) &= \sum_{x,y} p(y)p(x \mid y) log \frac{p(x \mid y)}{p(x)} \\ &= -\sum_{x}\sum_{y} p(y)p(x \mid y) log p(x) + \sum_{y} p(y) \sum_{x} p(x \mid y) log p(x \mid y) \\ &= H(X) - H(X \mid Y). \end{aligned} \tag{4.11}$$

Similarly, we have $I(X, Y) = H(Y) - H(Y \mid X)$, so mutual information can be thought of as the amount of entropy removed (on average) in X by observing Y. We may also think of mutual information as measuring the similarity between the joint distribution of X and Y and their distribution when they are treated as independent.

Comparing the definition (4.10) to that for KL-divergence, we see that if P_{XY} is the joint distribution of X and Y , while P_X and P_Y are their marginal distributions (distributions when X and Y are treated independently), then

$$I(X, Y) = D_{kl}(P_{XY} \mid\mid P_X P_Y) \geq 0. \tag{4.12}$$

Moreover, we have $I(X, Y) > 0$ unless X and Y are independent.

4.4 The α-divergence

The alpha-divergence also known as the Renyi divergence is related to Renyi entropy much like Kullback-Leibler divergence is related to Shannon's entropy, and comes up in many settings. It was introduced by Renyi as a measure of information that satisfies almost the same axioms as Kullback-Leibler divergence, and depends on a parameter that is called its order. In particular, the Renyi divergence of order 1 equals the Kullback-Leibler divergence. so in a way the alpha divergence is a generalization of the kullback-leibler divergence

Shannon entropy and Kullback-Leibler divergence (also known as information divergence or relative entropy) are perhaps the two most fundamental quantities in information theory and its applications. Because of their success, there have been many attempts to generalize these concepts, and in the literature one will find numerous entropy and divergence measures. Most of these quantities have never found any applications, and almost none of them have found an interpretation in terms of coding. The most important exceptions are the Renyi entropy and Renyi divergence. Renyi divergence appears as a crucial tool in proofs of convergence of minimum description length and Bayesian estimators, both in parametric and nonparametric models. and one may recognize it implicitly in many computations throughout information theory. It is also closely related to Hellinger distance, which is commonly used in the analysis of nonparametric density estimation.

4.4.1 Definition and properties

Denote by L and G be two probabilities on $\mathbb{R}^M$ where G is absolutely continuous with respect to P. Given a convex function $\varphi : [0, +\infty] \to [0, +\infty]$ such that $\varphi(1) = 0$, we define the φ-divergence between G and L as follows:

$$D_\varphi(G, L) := \int_{\mathbb{R}^M} \varphi\left(\frac{dG}{dL}(\boldsymbol{t})\right) dP(\boldsymbol{t}), \tag{4.13}$$

where $\left(\dfrac{dG}{dL}\right)(\cdot)$ stands for the Radon-Nikodym derivative of G with respect to L. We define $\varphi(0/0) = 0$ and $\varphi(b/0) = \lim_{t\to 0} t\,\varphi(b/t) = \lim_{s\to\infty} \varphi(s)/s$. If G is not absolutely continuous with respect to L, then we set $D_\varphi(G, L) = +\infty$. Note that if $g(\cdot)$ and $l(\cdot)$ are respectively the densities of G and L, with respect to the Lebesgue measure on $\mathbb{R}^M$, the α-divergence between the two probabilities (4.13) is formulated in this case as follows:

$$D_\varphi(G, L) := \int_{\mathbb{R}^M} \varphi\left(\frac{q(\boldsymbol{t})}{l(\boldsymbol{t})}\right) l(\boldsymbol{t}) d(\boldsymbol{t}). \tag{4.14}$$

We are interested in the class of the so-called "power divergences" especially the α-divergence that belongs to φ-divergences which was introduced by [43], using the following real convex functions $\varphi_\alpha(y)$ for different values of α

$$\frac{y^\alpha - \alpha y + \alpha - 1}{\alpha(1-\alpha)}, \quad \forall \alpha \in \mathbb{R} \setminus \{0, 1\}, \tag{4.15}$$

$$-\log(y) + y - 1, \quad \alpha = 0, \tag{4.16}$$

$$y\log(y) - y + 1, \qquad \alpha = 1. \tag{4.17}$$

The $\varphi-$divergence has many properties such as:

• **Nonnegativity** The $\varphi-$divergence is always nonnegative, and equal to zero if and only if probability densities $p(x) = q(x)$. This can be done by the Jensens inequality

$$D_\varphi(Q,P) = \int_{\mathbb{R}^M} \varphi\left(\frac{q(\boldsymbol{t})}{p(\boldsymbol{t})}\right) p(\boldsymbol{t})d(\boldsymbol{t}) \geq \varphi\left(\int_{\mathbb{R}^M} \frac{q(\boldsymbol{t})}{p(\boldsymbol{t})} p(\boldsymbol{t})d(\boldsymbol{t})\right) \varphi(1) = 0. \tag{4.18}$$

• **Convexity** $\forall \quad 0 \leq \beta \leq 1$ we have

$$D_\varphi(\beta Q_1 + (1-\beta)Q_2, \beta P_1 + (1-\beta)P_2) \leq \beta D_\varphi(Q_1, P_1) + (1-\beta)D_\varphi(Q_2, P_2). \tag{4.19}$$

• **Scaling** $\forall \lambda > 0$ we get

$$\lambda D_\varphi(Q,P) = D_{\lambda\varphi}(Q,P). \tag{4.20}$$

• **Invariance** The $\varphi-$divergence is invariant to bijective transformations [6], i.e. Let $f = k(t)$ be a bijective transformation, $P(t)$ and $\tilde{P}(f)$ be respectively the probability distribution of t and f,then

$$D_\varphi(Q,P) = D_\varphi(\tilde{Q}, \tilde{P}). \tag{4.21}$$

• **Boundedness** If the following limit exists and it is finite, $\varphi-$divergence for positive densities is bounded see [107, 153]

$$0 \leq D_\varphi(Q,P) \leq \lim_{t\to 0^+} (\varphi(t) + t\varphi(1/t)). \tag{4.22}$$

Furthermore we have the following inequality see [53]

$$0 \leq D_\varphi(Q,P) \leq \int (q(\boldsymbol{t}) - p(\boldsymbol{t}))\varphi'(\frac{q(\boldsymbol{t})}{p(\boldsymbol{t})})d(\boldsymbol{t}). \tag{4.23}$$

• **Symmetricity** It is possible to construct a symmetric divergence from the $\varphi-$divergence by

$$\varphi_s(t) = \varphi(t) + \varphi^*(t). \tag{4.24}$$

In the following, we denote by D_α the alpha-divergence. Similar to the KL divergence, these properties allow us to minimize the α-divergence in order to find the best approximating distribution q(x) in some class of potential approximations. There are several special cases for various values of α that are of interest to us. The most important cases are The well known Kullback Leibler divergence and its modified version:

$$lim_{\alpha\to 0} D_\alpha(p \parallel q) = KL(q \parallel p) \tag{4.25}$$

$$lim_{\alpha\to 1} D_\alpha(p \parallel q) = KL(p \parallel q) \tag{4.26}$$

Hence the α-divergences include the KL divergences as a special case. Other special cases are

$$D_{-1}(p \parallel q) = \frac{1}{2}\int \frac{(q(x)-p(x))^2}{q(x)}dx \tag{4.27}$$

$$D_2(p \parallel q) = \frac{1}{2}\int \frac{(q(x)-p(x))^2}{p(x)}dx \tag{4.28}$$

$$D_{\frac{1}{2}}(p \parallel q) = 2\int (\sqrt{p(x)}-\sqrt{q(x)})^2 dx \tag{4.29}$$

$D_{\frac{1}{2}}$ is known as the Hellinger distance. $\sqrt{D_{\frac{1}{2}}}$ is a valid distance metric (it satisfies both the traingle inequality and symmetric properties).

Table 4.1 gives, according to the choice of φ_α, the associated D_α-divergence.

The function φ_α	φ_{-1}	φ_0	$\varphi_{1/2}$	φ_1	φ_2
The divergence D_α	$\mathcal{X}_m^2$	KL_m	H	KL	$\mathcal{X}^2$

Table 4.1: Examples of standard divergences.

4.4.2 α-divergence as a cost function

Let $\boldsymbol{Z} := (Z_1,\ldots,Z_p)^\top \in \mathbb{R}^p$, $p \geq 1$, a random vector. The α-divergence D_α between the product density $\prod_{i=1}^{p} f_{Z_i}(\cdot)$ of the marginal densities f_{Z_i} of the components Z_i, $i \in \{1,\ldots,p\}$, and the joint density $f_{\boldsymbol{Z}}(\cdot)$ of the random vector $\boldsymbol{Z}$, is given as follows:

$$\begin{aligned} D_\alpha\left(\prod_{i=1}^{p} f_{Z_i}, f_{\boldsymbol{Z}}\right) &:= \int_{\mathbb{R}^p} \varphi_\alpha\left(\frac{\prod_{i=1}^{p} f_{Z_i}(z_i)}{f_{\boldsymbol{Z}}(\boldsymbol{z})}\right) f_{\boldsymbol{Z}}(\boldsymbol{z}) dz_1,\ldots,dz_p, \\ &:= \mathbb{E}\left[\varphi_\alpha\left(\frac{\prod_{i=1}^{p} f_{Z_i}(Z_i)}{f_{\boldsymbol{Z}}(\boldsymbol{Z})}\right)\right], \end{aligned} \tag{4.30}$$

with $\mathbb{E}$ corresponds to the mathematical expectation.

This measure $D_\alpha\left(\prod_{i=1}^{p} f_{Z_i}, f_{\boldsymbol{Z}}\right)$ is always positive and only reaches zero, the minimum value, if $\prod_{i=1}^{p} f_{Z_i}(\cdot) = f_{\boldsymbol{Z}}(\cdot)$, in other words, when the components of the random vector $\boldsymbol{Z}$ are mutually independent.

4.5 Conclusion

In this chapter we gave an overview on α-divergences and its properties. It should be stated that it has been proved that the use of α-divergences lead to better results compared with mutual information especially for noisy mixture signals [54]. We expect that we have the same results for dependent sources.

Part II

Blind source separation for independent and dependent sources

Chapter 5

Instantaneous BSS via copulas

5.1 Introduction

For separating linear instantaneous mixtures of independent/dependent source components, a new blind source separation (BSS) technique was proposed. The proposed approaches are based on minimizing the kullback-leibler divergence between copula densities. They cover the more general case in which the source components' dependency structure and/or the associated parameter are unknown. Many simulation results are presented, demonstrating that the proposed algorithms ensure accurate separation in situations where other approaches fail.

5.2 The proposed methodology

Let $\boldsymbol{Y} = (Y_1, \dots, Y_p)^\top \in \mathbb{R}^p$ be any random vector, with continuous marginal distribution functions $F_{Y_1}(\cdot)$, $\dots$, $F_{Y_p}(\cdot)$, joint probability density $f_{\boldsymbol{Y}}(\cdot)$, and marginal probability densities $f_{Y_1}(\cdot), \dots, f_{Y_p}(\cdot)$. It has been shown in [96] that the Mutual Information (MI) of $\boldsymbol{Y}$ can be written as the Kullback-Leibler divergence between the copula density $\boldsymbol{c}_{\boldsymbol{Y}}$ of the random vector $\boldsymbol{Y}$ and the copula density $\boldsymbol{c}_0$ of independence. In fact, by changing variables in the integral and using the relation (3.8), we can write

$$\begin{aligned}
MI(\boldsymbol{Y}) &:= \int_{\mathbb{R}^p} \log\left(\frac{f_{\boldsymbol{Y}}(\boldsymbol{y})}{\prod_{i=1}^{p} f_{Y_i}(y_i)}\right) f_{\boldsymbol{Y}}(\boldsymbol{y})\, d\boldsymbol{y} \\
&= \int_{[0,1]^p} \log\left(\frac{\boldsymbol{c}_{\boldsymbol{Y}}(\boldsymbol{u})}{1}\right) \boldsymbol{c}_{\boldsymbol{Y}}(\boldsymbol{u})\, d\boldsymbol{u} \\
&= \int_{[0,1]^p} \log\left(\frac{\boldsymbol{c}_{\boldsymbol{Y}}(\boldsymbol{u})}{\boldsymbol{c}_0(\boldsymbol{u})}\right) \boldsymbol{c}_{\boldsymbol{Y}}(\boldsymbol{u})\, d\boldsymbol{u} \qquad (5.1) \\
&=: KL\left(\boldsymbol{c}_{\boldsymbol{Y}}, \boldsymbol{c}_0\right) \\
&= \mathbb{E}\left[\log\left(\boldsymbol{c}_{\boldsymbol{Y}}(F_{Y_1}(Y_1), \dots, F_{Y_p}(Y_p))\right)\right], \qquad (5.2)
\end{aligned}$$

where $\mathbb{E}(\cdot)$ denotes the mathematical expectation. By definition the integral in (5.1) is, the Kullback-Leibler divergence between $\boldsymbol{c}_{\boldsymbol{Y}}$ and $\boldsymbol{c}_0$. Note that the criterion

$KL(\boldsymbol{c_Y}, \boldsymbol{c}_0)$ is always nonnegative, and

$$KL(\boldsymbol{c_Y}, \boldsymbol{c}_0) = 0 \quad \text{iff} \quad \boldsymbol{c_Y}(\boldsymbol{u}) = \boldsymbol{c}_0(\boldsymbol{u}),\ \forall \boldsymbol{u} \in [0,1]^p. \tag{5.3}$$

To put it another way, $KL(\boldsymbol{c_Y}, \boldsymbol{c}_0) \geq 0$, and $KL(\boldsymbol{c_Y}, \boldsymbol{c}_0) = 0$ if and only if the components of the vector $\boldsymbol{Y}$ are independent.

We present three different BSS procedures, depending on whether or not training samples of the sources are available, and/or whether or not prior knowledge about their dependency structure is available. In what follows

- We assume that the random (vector) processes $\boldsymbol{s}(t)$, $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t) = \boldsymbol{B}\,\boldsymbol{x}(t)$, $t \in [0,T]$, are stationary, so that the corresponding sampled versions, with certain time period, say T_e, $\forall n = 1, \ldots, N_t$,

  $$\boldsymbol{s}(n) := \boldsymbol{s}(nT_e),\ \boldsymbol{x}(n) := \boldsymbol{x}(nT_e) \text{ and } \boldsymbol{y}(n) := \boldsymbol{y}(nT_e) = \boldsymbol{B}\,\boldsymbol{x}(n), \tag{5.4}$$

  can be viewed as realizations of random vectors in $\mathbb{R}^p$, which will be denoted, respectively, by

  $$\boldsymbol{S}, \quad \boldsymbol{X} \quad \text{and} \quad \boldsymbol{Y} := \boldsymbol{B}\boldsymbol{X}; \tag{5.5}$$

- The source components may be statistically dependent with unknown semiparametric copula density $\{\boldsymbol{c_S}(\cdot;\theta);\ \theta \in \Theta \subset \mathbb{R}\}$. We introduce the following objective function

  $$\boldsymbol{B} \quad \mapsto \quad KL(\boldsymbol{c_Y}, \boldsymbol{c_S}(\cdot;\theta)) := \int_{[0,1]^p} \log\left(\frac{\boldsymbol{c_Y}(\boldsymbol{u})}{\boldsymbol{c_S}(\boldsymbol{u};\theta)}\right) \boldsymbol{c_Y}(\boldsymbol{u})\, d\boldsymbol{u} \tag{5.6}$$

  $$= \quad \mathbb{E}\left[\log\left(\frac{\boldsymbol{c}_{\mathbf{Y}}\left(F_{Y_1}(Y_1), \ldots, F_{Y_p}(Y_p)\right)}{\boldsymbol{c}_{\mathbf{S}}\left(F_{Y_1}(Y_1), \ldots, F_{Y_p}(Y_p);\theta\right)}\right)\right]. \tag{5.7}$$

  The integral (5.6) is, by definition, the Kullback-Leibler divergence between the copula density of the random vector $\boldsymbol{Y}$ and the copula density of the random source vector $\boldsymbol{S}$.

- For the proposed algorithms dealing with linear instantaneous mixtures of dependent source components (we limit ourselves to the case of two mixtures/two sources), we provide numerous simulation results. As synthetic source signals, we will use two RGB images,“Lena” and “Barbara”. For estimating the proposed criteria and the demixing matrix, each image will be converted to grayscale, and will be considered as 1d-signal of dimension $N_t = 512 \times 512 = 562144$ pixels. To select a model of the semiparametric copula density $\{\boldsymbol{c_S}(F_{S_1}(\cdot), F_{S_2}(\cdot);\theta);\ \theta \in \Theta \subset \mathbb{R}\}$ of the source vector $\boldsymbol{S} = (S_1, S_2)^\top$, we apply the AIC model selection step, see Subsection 3.6, among seven copula models : T, Gaussian, Clayton, FGM, AMH, Frank, and Gumbel. We mixed the image sources using the mixing matrix $\boldsymbol{A} = [1\ 0.7; 0.7\ 1]$. In order to estimate the proposed separation criteria and the demixing matrix, we use $N = 3000 \ll N_t = 512 \times 512$ (vector) observations randomly sampled with replacement from the whole (vector) signal $\boldsymbol{y}(n)$, $n = 1\ldots, N_t$.

The simulation results will be compared to some well-known instantaneous ICA approaches results under the same conditions: [54] (MI), [23] (JADE), [83] (FastICA), [117] (RADICAL) and [17] (InfoMax). All simulations are repeated 100 times.

The signal-to-noise ratio $SNR\,(dB)$ and the peak signal-to-noise ratio $PSNR\,(dB)$ are used to assess the accuracy of source estimation, and they are defined respectively by:

$$SNR_i := 10\log_{10}\frac{\sum_{n=1}^{N} s_i(n)^2}{\sum_{n=1}^{N}\left(\widehat{s}_i(n)-s_i(n)\right)^2}$$

and

$$PSNR_i := 10\log_{10}\frac{\max\limits_{n=1,\ldots,N_t} s_i(n)^2}{\frac{1}{N_t}\sum_{n=1}^{N_t}\left(\widehat{s}_i(n)-s_i(n)\right)^2},\ i=1,2.$$

5.2.1 Procedure 1 : Both the copula model and the parameter are known

We assume that the copula density model of the sources is known and has a known parameter. If it is not the case, We assume that we have training samples $\boldsymbol{s}(1),\ldots,\boldsymbol{s}(N)\in\mathbb{R}^p$ of the source vector $\boldsymbol{S}$.

We may use the model selection procedure described in Subsection 3.6 to select a model from these training samples, among a collection of semiparametric copula density models $M_l := \{\boldsymbol{c}_l(\cdot;\theta_l);\ \theta_l\in\Theta_l\subset\mathbb{R}\}$, $\forall\, l=1,\ldots,L$. Denote $\{\boldsymbol{c}_{\mathbf{S}}(\cdot;\theta);\ \theta\in\Theta\subset\mathbb{R}\}$, the selected model, and $\overline{\theta}$ the corresponding estimated parameter; see (4.27). The criterion function $\boldsymbol{B}\mapsto KL\left(\boldsymbol{c}_{\boldsymbol{Y}},\boldsymbol{c}_{\mathbf{S}}(\cdot;\overline{\theta})\right)$ is nonnegative, and achieves its minimum value zero iff $\boldsymbol{B}=\boldsymbol{A}^{-1}$ (up to scale and permutation indeterminacies of rows), i.e., $\boldsymbol{A}^{-1}=\arg\inf\limits_{\boldsymbol{B}} KL\left(\boldsymbol{c}_{\boldsymbol{Y}},\boldsymbol{c}_{\mathbf{S}}(\cdot;\overline{\theta})\right)$, provided that the copula density $\boldsymbol{c}_{\mathbf{S}}(\cdot;\overline{\theta})$ of the random vector source $\boldsymbol{S}$ satisfies the following assumption (A.1): for any regular matrix $\boldsymbol{M}$, if the copula density of $\boldsymbol{MS}$ is equal to $\boldsymbol{c}_{\mathbf{S}}(\cdot;\overline{\theta})$, then $\boldsymbol{M}=\boldsymbol{DP}$, for some diagonal matrix $\boldsymbol{D}$ and permutation matrix $\boldsymbol{P}$. To achieve separation, we propose to minimize with respect to $\boldsymbol{B}$ some statistical estimate $\widehat{KL}\left(\boldsymbol{c}_{\boldsymbol{Y}},\boldsymbol{c}_{\mathbf{S}}(\cdot;\overline{\theta})\right)$, to be defined below, of the criterion

$$KL\left(\boldsymbol{c}_{\boldsymbol{Y}},\boldsymbol{c}_{\mathbf{S}}(\cdot;\overline{\theta})\right)=\mathbb{E}\left[\log\left(\frac{\boldsymbol{c}_{\boldsymbol{Y}}(F_{Y_1}(Y_1),\ldots,F_{Y_p}(Y_p))}{\boldsymbol{c}_{\mathbf{S}}(F_{Y_1}(Y_1),\ldots,F_{Y_p}(Y_p);\overline{\theta})}\right)\right], \tag{5.8}$$

constructed from the data $\boldsymbol{y}(1),\ldots,\boldsymbol{y}(N)$. The demixing matrix is then estimated by

$$\widehat{\boldsymbol{B}} := \arg\inf_{\boldsymbol{B}} \widehat{KL}\left(\boldsymbol{c}_{\boldsymbol{Y}},\boldsymbol{c}_{\mathbf{S}}(\cdot;\overline{\theta})\right). \tag{5.9}$$

In view of (5.8), using a “plug-in” statistical estimation method, we propose the following estimate of the above criterion $KL\left(\boldsymbol{c}_{\boldsymbol{Y}},\boldsymbol{c}_{\mathbf{S}}(\cdot;\overline{\theta})\right)$

$$\widehat{KL}\left(\boldsymbol{c}_{\boldsymbol{Y}},\boldsymbol{c}_{\mathbf{S}}(\cdot;\overline{\theta})\right) := \frac{1}{N}\sum_{n=1}^{N}\log\left(\frac{\widehat{\boldsymbol{c}}_{\boldsymbol{Y}}(\widehat{F}_{Y_1}(y_1(n)),\ldots,\widehat{F}_{Y_p}(y_p(n)))}{\boldsymbol{c}_{\mathbf{S}}(\widehat{F}_{Y_1}(y_1(n)),\ldots,\widehat{F}_{Y_p}(y_p(n));\overline{\theta})}\right), \tag{5.10}$$

where

$$\widehat{\boldsymbol{c}}_{\boldsymbol{Y}}(\boldsymbol{u}) := \frac{1}{NH_1 \cdots H_p} \sum_{m=1}^{N} \prod_{j=1}^{p} k\left(\frac{\widehat{F}_{Y_j}(y_j(m)) - u_j}{H_j}\right), \forall \boldsymbol{u} \in [0,1]^p, \tag{5.11}$$

is the kernel estimate of the copula density $\boldsymbol{c}_{\boldsymbol{Y}}(\cdot)$, and $\widehat{F}_{Y_j}(x)$, $j = 1, \ldots, p$, is the smoothed estimate of the marginal distribution function $F_{Y_j}(x)$ of the random variable Y_j, at any real value $x \in \mathbb{R}$, defined by

$$\widehat{F}_{Y_j}(x) := \frac{1}{N} \sum_{m=1}^{N} K\left(\frac{y_j(m) - x}{h_j}\right), \forall j = 1, \ldots, p, \tag{5.12}$$

where $K(\cdot)$ is the primitive of a kernel $k(\cdot)$, a symmetric centered probability density. In our forthcoming simulation study, we will use the triangular kernel

$$k(x) = (1 - |x|)\mathbb{1}_{[-1,1]}(x), \forall x \in \mathbb{R}.$$

A more appropriate choice of the kernel $k(\cdot)$, for estimating the copula density, can be done according to [125], which copes with the boundary effect. The bandwidth parameters $H_1, \ldots, H_p$ and $h_1, \ldots, h_p$ in (5.11) and (5.12) will be chosen according to Silverman's rule of thumb, see [138], i.e., for all $j = 1, \ldots, p$, we take

$$H_j = \left(\frac{4}{p+2}\right)^{\frac{1}{p+4}} N^{\frac{-1}{p+4}} \widehat{\Sigma}_j, \quad \text{and} \quad h_j = \left(\frac{4}{3}\right)^{\frac{1}{5}} N^{\frac{-1}{5}} \widehat{\sigma}_j,$$

where $\widehat{\sigma}_j$ and $\widehat{\Sigma}_j$ are, respectively, the empirical standard deviation of the data $y_j(1), \ldots, y_j(N)$ and $\widehat{F}_{Y_j}(y_j(1)), \ldots, \widehat{F}_{Y_j}(y_j(N))$. The solution $\widehat{\boldsymbol{B}}$, the estimate of the demixing matrix, can be computed by a descent gradient algorithm, taking as initial matrix $\boldsymbol{B}_0 = I_p$, the $p \times p$ identity matrix. The gradient in $\boldsymbol{B}$ of $\widehat{KL}\left(\boldsymbol{c}_{\boldsymbol{Y}}, \boldsymbol{c}_{\mathbf{S}}(\cdot;\overline{\theta})\right)$ can be computed from the proper definitions of the estimates as follows

$$\frac{d\widehat{KL}\left(\boldsymbol{c}_{\boldsymbol{Y}}, \boldsymbol{c}_{\mathbf{S}}(\cdot;\overline{\theta})\right)}{d\boldsymbol{B}} = \frac{1}{N} \sum_{n=1}^{N} \frac{\frac{d}{d\boldsymbol{B}} \widehat{\boldsymbol{c}}_{\boldsymbol{Y}}(\widehat{F}_{Y_1}(y_1(n)), \ldots, \widehat{F}_{Y_p}(y_p(n)))}{\widehat{\boldsymbol{c}}_{\boldsymbol{Y}}(\widehat{F}_{Y_1}(y_1(n)), \ldots, \widehat{F}_{Y_p}(y_p(n)))} - \frac{\frac{d}{d\boldsymbol{B}} \boldsymbol{c}_{\mathbf{S}}(\widehat{F}_{Y_1}(y_1(n)), \ldots, \widehat{F}_{Y_p}(y_p(n)); \overline{\theta})}{\boldsymbol{c}_{\mathbf{S}}(\widehat{F}_{Y_1}(y_1(n)), \ldots, \widehat{F}_{Y_p}(y_p(n)); \overline{\theta})},$$

where $\frac{d}{d\boldsymbol{B}} := \left(\frac{\partial}{\partial \boldsymbol{B}_{ij}}\right)_{ij}$, $i, j = 1, \ldots, p$, and

$$\frac{\partial \widehat{\boldsymbol{c}}_{\boldsymbol{Y}}(\widehat{F}_{Y_1}(y_1(n)), \ldots, \widehat{F}_{Y_p}(y_p(n)))}{\partial \boldsymbol{B}_{ij}} = \frac{1}{NH_1 \cdots H_p} \sum_{m=1}^{N} \prod_{j=1, j \neq i}^{p} k\left(\frac{\widehat{F}_{Y_j}(y_j(m)) - \widehat{F}_{Y_j}(y_j(n))}{H_j}\right) \times k'\left(\frac{\widehat{F}_{Y_i}(y_i(m)) - \widehat{F}_{Y_i}(y_i(n))}{H_i}\right) \frac{1}{H_i} \frac{\partial(\widehat{F}_{Y_i}(y_i(m)) - \widehat{F}_{Y_i}(y_i(n)))}{\partial \boldsymbol{B}_{ij}}$$

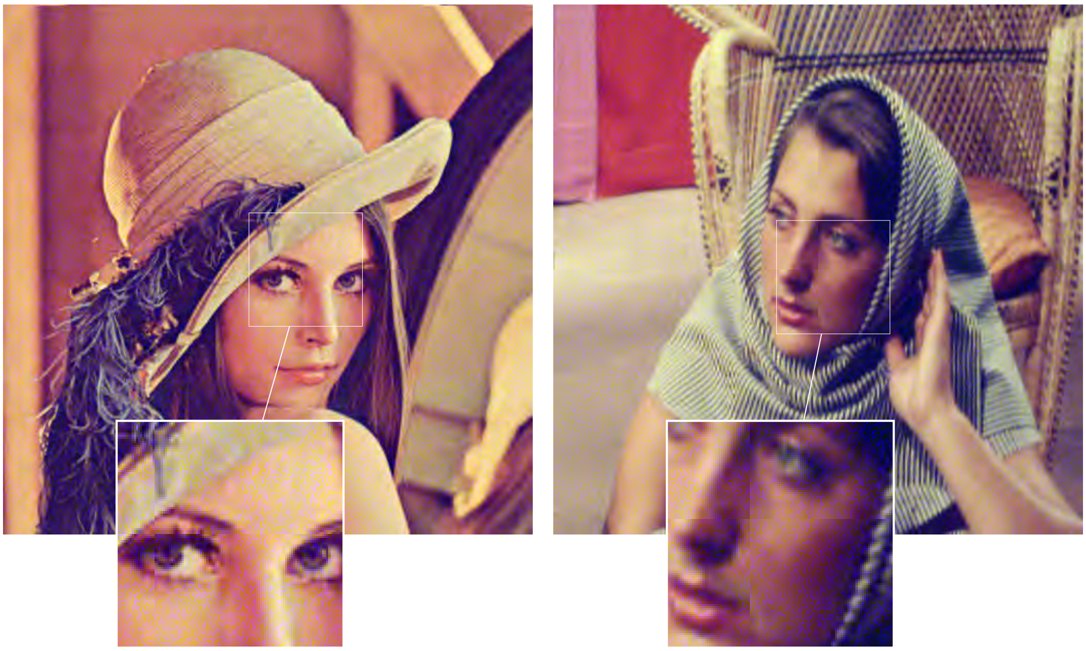
with

$$\frac{\partial(\widehat{F}_{Y_j}(y_j(m))}{\partial \boldsymbol{B}_{ij}} = \frac{1}{Nh_i}\sum_{l=1}^{N} k\left(\frac{y_i(l) - y_i(m)}{h_i}\right)(x_j(l) - x_j(m)), \forall i, j = 1, \ldots, p.$$

Simulation results

We present some simulation results to demonstrate the efficacy of procedure 1 mentioned above. In order to choose a copula density model. We apply the AIC model selection step among seven candidate models : T, Gaussian, FGM, Clayton, AMH, Frank, and Gumbel, in order to select a copula density model $\{\boldsymbol{c}_{\boldsymbol{S}}(F_{S_1}(\cdot), F_{S_2}(\cdot); \theta); \theta \in \Theta \subset \mathbb{R}\}$ for the source vector $\boldsymbol{S}$. The parameter θ will be estimated by $\overline{\theta}$ given in (4.27). The separating matrix $\widehat{\boldsymbol{B}}$, defined in (5.9), is computed using descent gradient algorithm taking the identity matrix for initialization, and descent gradient parameter $\mu = 0.1$. In order to assess the visual quality, part of each image has been zoomed.

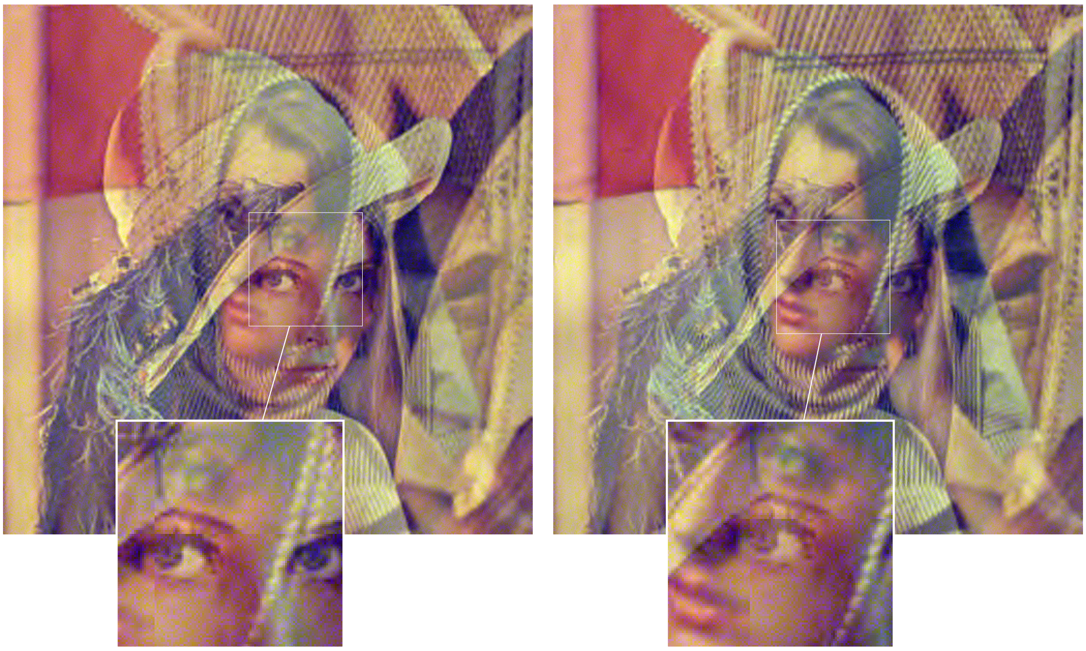
As sources, we consider the two well-known images "Lena" and "Barbara" shown in Figures 5.1a and 5.1b, respectively. The corresponding empirical Kendall's tau is $\widehat{\tau}(\boldsymbol{S}) = -0.1420$. The AIC model selection step provides AMH copula with $\overline{\theta} = -0.7300$ for modeling the dependency structure of the components of $\boldsymbol{S}$. The Kendall's tau of the selected copula is $\tau_{\mathbb{C}}(\overline{\theta}) = -0.1403$. The mixtures $\boldsymbol{x}(n)$, $n = 1, \ldots, N_t$, using the mixing matrix $\boldsymbol{A} = [1\ 0.8; 0.8\ 1]$ are shown in Figures 5.2a and 5.2b. The empirical Kendall's tau, of the mixtures, is $\widehat{\tau}(\boldsymbol{X}) = 0.9999$. Figure 5.3 show the estimated images obtained by applying procedure 1. These results are compared with the ones obtained by MI method, see Figure 5.4. For more assessment, the mean of the SNRs reached by, procedure 1 and MI method, are also plotted; see Figure 5.5a. Figure 5.5b and 5.6a illustrate, respectively, the separation criterion and $\widehat{\tau}(\boldsymbol{Y})$ values vs iterations; we can see that the proposed separation criterion converges to zero, and $\widehat{\tau}(\boldsymbol{Y})$ converges to $\widehat{\tau}(\boldsymbol{S})$, when the separation is achieved. Table 5.1 shows the PSNR and the SNR final values for each method. The proposed one provides better results.



(a) Lena

(b) Barbara

Figure 5.1: The source images: Lena and Barbara



(a) Mixed image 1

(b) Mixed image 2

Figure 5.2: The mixed ones

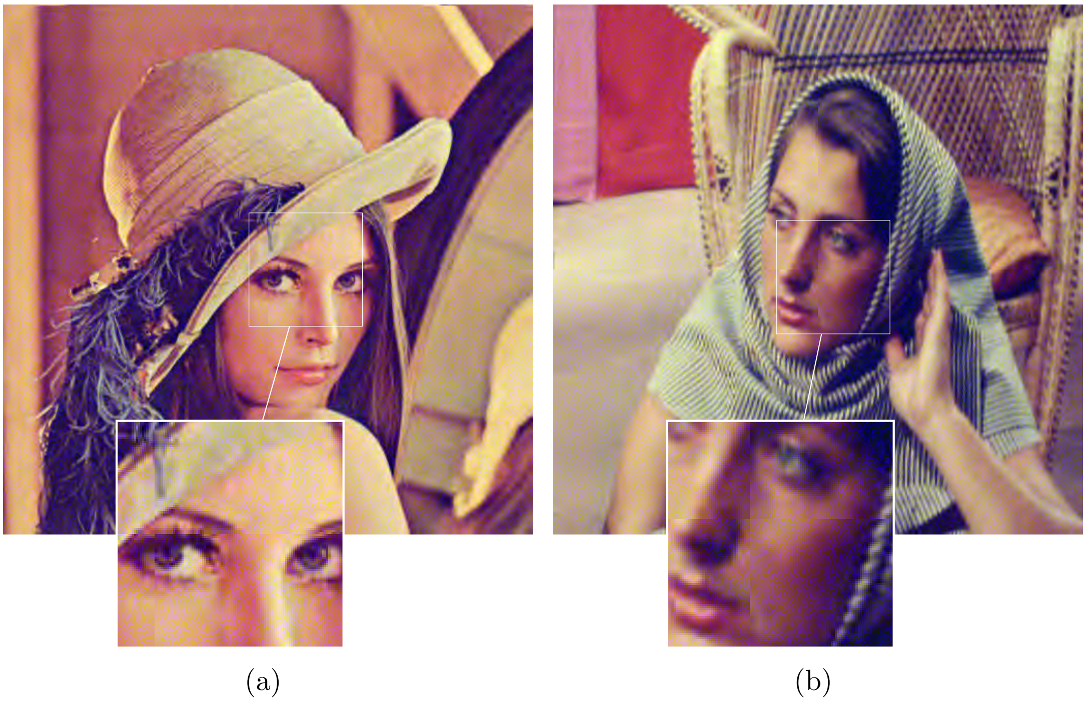


Figure 5.3: Image separation results: Procedure 1 with AMH copula

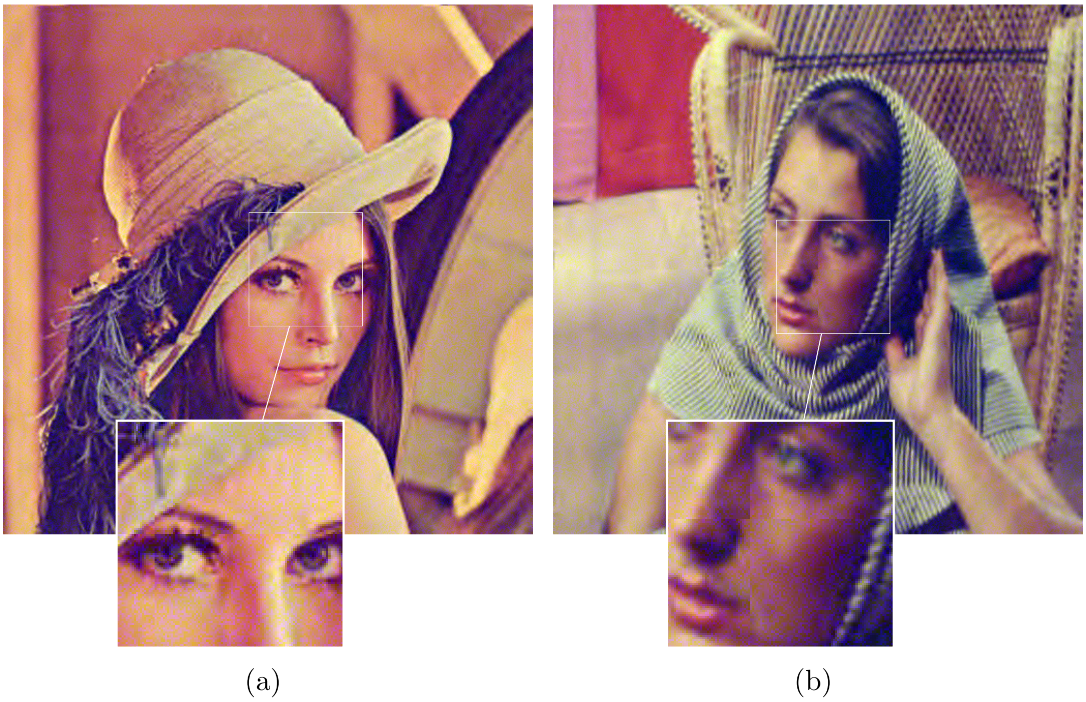
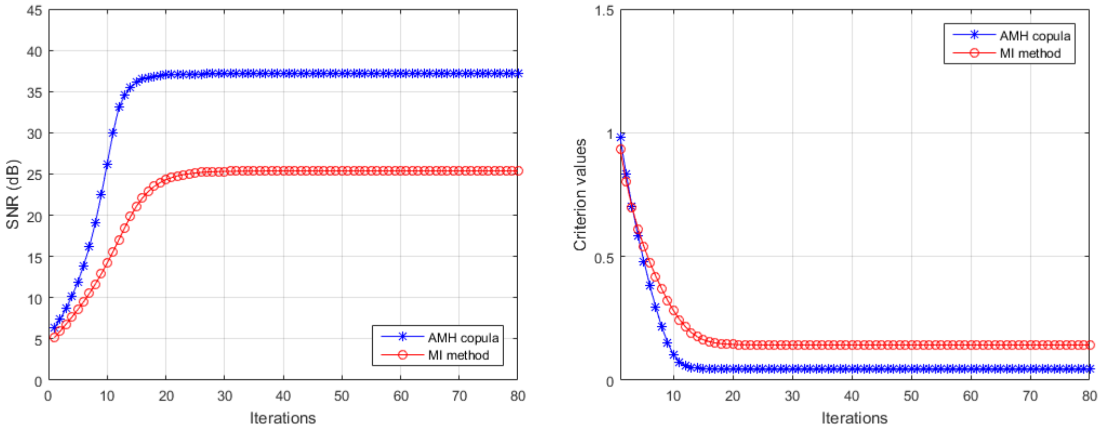
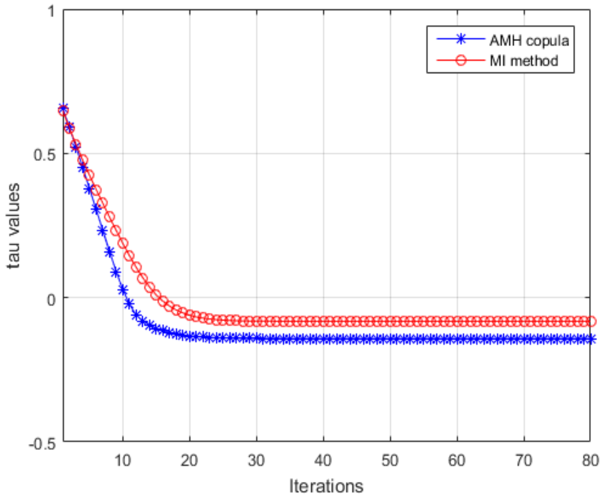


Figure 5.4: Image separation results: MI method



(a) (b)

Figure 5.5: (a) SNRs versus iterations. (b) Criterion values vs iterations



(a)

Figure 5.6: (a) $\widehat{\tau}(\boldsymbol{Y})$ values vs iterations

	SNR (dB)		PSNR (dB)	
MethodsSources	Lena	Barbara	Lena	Barbara
AMH copula	**37.0974**	**37.2727**	**42.7985**	**43.2834**
MI	25.4432	25.2853	35.1233	34.7318
JADE	23.670	24.2969	31.6743	32.9167
RADICAL	26.5644	21.9602	33.7704	31.1324
FastICA	32.6820	14.6023	24.8379	9.68362
InfoMax	6.4597	6.3073	13.0119	11.5812
Sobi	15.6025	15.0689	19.7859	20.5176

Table 5.1: PSNR and final values of SNR for Procedure 1: Lena and Barbara

5.2.2 Procedure 2 : The copula model is known and the parameter is unknown

In this Subsection, we have not at hand training samples of the source vector $\boldsymbol{S}$. We assume that the copula density model of the source components is known with unknown parameter θ; denote it $\{c_{\boldsymbol{S}}(\cdot;\theta);\, \theta \in \Theta \subset \mathbb{R}\}$. Since θ is unknown, we propose to consider, instead of (5.9), the following estimate of the demixing matrix

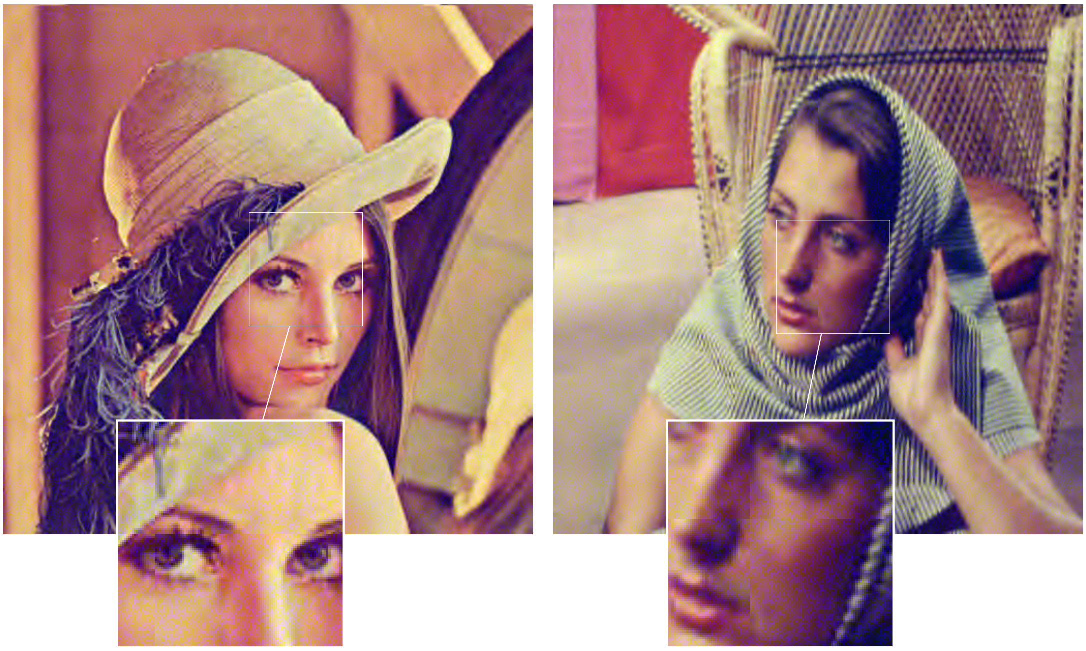
$$\widehat{\boldsymbol{B}} = \arg\inf_{\boldsymbol{B}} \inf_{\theta\in\Theta} \widehat{KL}\left(\boldsymbol{c}_{\boldsymbol{Y}}, \boldsymbol{c}_{\boldsymbol{S}}(\cdot;\theta)\right), \tag{5.13}$$

which can be computed using gradient descent algorithm on both $\boldsymbol{B}$ and θ of the criterion function $(\boldsymbol{B},\theta) \mapsto \widehat{KL}\left(\boldsymbol{c}_{\boldsymbol{Y}}, \boldsymbol{c}_{\boldsymbol{S}}(\cdot;\theta)\right)$. Note that $\boldsymbol{B} \mapsto \inf_{\theta\in\Theta} KL\left(\boldsymbol{c}_{\boldsymbol{Y}}, \boldsymbol{c}_{\boldsymbol{S}}(\cdot;\theta)\right)$ is nonnegative and achieves its minimum value zero iff $\boldsymbol{B} = \boldsymbol{A}^{-1}$ (up to scale and permutation indeterminacies), provided that the copula density model $\{\boldsymbol{c}_{\boldsymbol{S}}(\cdot;\theta);\, \theta \in \Theta \subset \mathbb{R}\}$ of $\boldsymbol{S}$ satisfies the following assumption (A.2): for any regular matrix $\boldsymbol{M}$, if the copula density, of $\boldsymbol{MS}$, belongs to $\{\boldsymbol{c}_{\boldsymbol{S}}(\cdot;\theta);\, \theta \in \Theta \subset \mathbb{R}\}$, then $\boldsymbol{M} = \boldsymbol{DP}$, where $\boldsymbol{D}$ is diagonal and $\boldsymbol{P}$ is a permutation. The estimates of copula density and the marginal distribution functions are defined as before. The gradient in $\boldsymbol{B}$ can be explicitly computed in similar way as in Subsection 5.2.1.

Simulation results

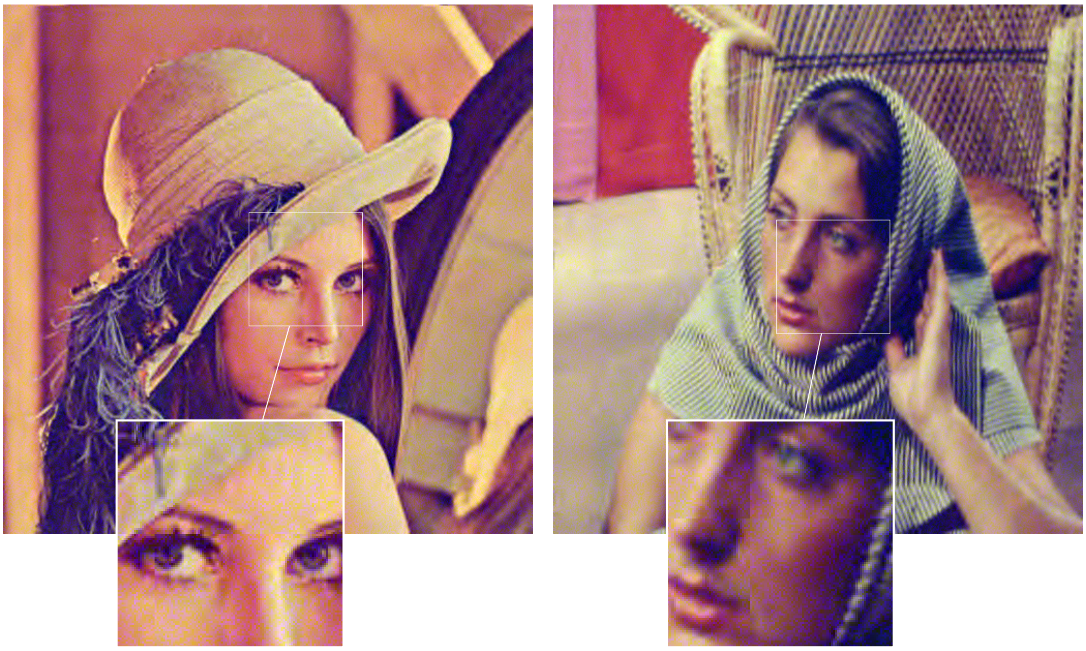
The efficiency of the separation approach described in procedure 2 is illustrated through the example using the same source images as above.

The dependency structure of the sources (Lena and Barbara) is modeled by AMH copula model with unknown parameter $\theta \in \Theta = [-1, 1]$. From Figure 5.10b, we can see that the iterated values of θ, in the descent gradient, converges to the value $\theta = -0.7834$, which is close to $\overline{\theta} = -0.7300$. The separation results are illustrated in Figure 5.7. see Figure 5.9a for the corresponding SNR's. In Table 5.2, are listed, PSNR and SNR final values, for all methods. Our separation criterion, as in procedure 1, converges to 0; see Figure 5.9b.



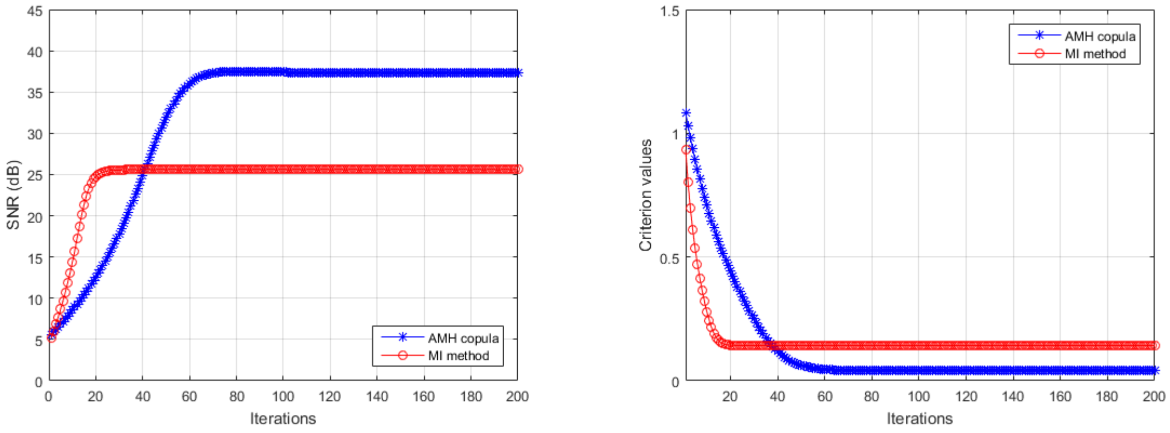
(a) (b)

Figure 5.7: Separation results: procedure 2 with AMH copula model



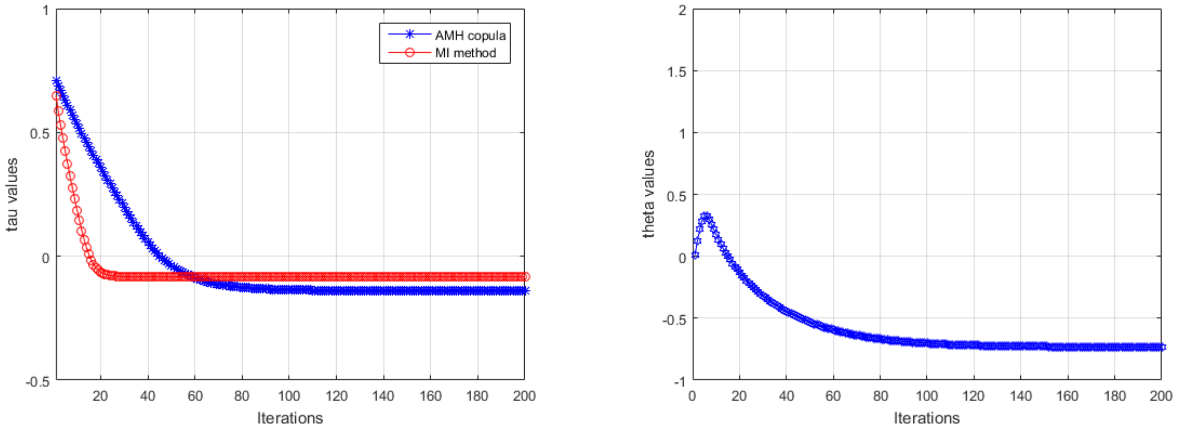
(a) (b)

Figure 5.8: Separation results: MI method



(a) (b)

Figure 5.9: (a) SNRs vs iterations. (b) Criterion values vs iterations



(a) (b)

Figure 5.10: (a) $\widehat{\tau}(\boldsymbol{Y})$ values vs iterations. (b) θ values vs iterations

	SNR (dB)		PSNR (dB)	
MethodsSources	Lena	Barbara	Lena	Barbara
AMH copula	**36.8855**	**37.7672**	**41.5766**	**42.8188**
MI	25.2245	25.9652	34.6609	34.8630
JADE	23.7372	24.3083	31.7770	32.8642
RADICAL	26.5328	22.0524	33.7738	31.1744
FastICA	33.4141	14.7283	21.3899	9.3695
InfoMax	6.4384	6.2929	13.0407	11.6043
Sobi	15.5965	15.0662	19.6044	20.0678

Table 5.2: PSNR and final values of SNR for Procedure 2: Lena and Barbara

5.2.3 Procedure 3 : The model and the parameter are unknown

We consider now the case where there is no prior information neither about the copula density model, of the source components, nor about the parameter θ. We propose the following methodology. We consider a set of K candidate models, for the source copula density,

$$M_k := \{\boldsymbol{c}_k(\cdot;\theta_k);\, \theta_k \in \Theta_k \subset \mathbb{R}\},\ k = 1,\ldots,K. \tag{5.14}$$

Assume that each model M_k, $k = 1,\ldots,K$, satisfies the identifiability condition (A.2) of Subsection 5.2.2. Then the criterion function

$$\boldsymbol{B} \mapsto \min_{k=1,\ldots,K} \inf_{\theta_k\in\Theta_k} KL\left(\boldsymbol{c}_{\boldsymbol{Y}}, \boldsymbol{c}_k(\cdot;\theta_k)\right) \tag{5.15}$$

is nonnegative and achieves its minimum value zero iff $\boldsymbol{B} = \boldsymbol{A}^{-1}$ (up to scale and permutation indeterminacies). As a result, we suggest that for each model, we use the method described in Subsection 5.2.2 and then choose the solution that minimizes the criterion across all models, i.e., to estimate the demixing matrix by

$$\widehat{\boldsymbol{B}} := \arg\inf_{\boldsymbol{B}} \inf_{\theta_{k^*}\in\Theta_{k^*}} \widehat{KL}\left(\boldsymbol{c}_{\boldsymbol{Y}}, \boldsymbol{c}_{k^*}(\cdot;\theta_{k^*})\right), \tag{5.16}$$

where

$$k^* = \arg\min_{k=1,\ldots,K} \inf_{\boldsymbol{B}} \inf_{\theta_k\in\Theta_k} \widehat{KL}\left(\boldsymbol{c}_{\boldsymbol{Y}}, \boldsymbol{c}_k(\cdot;\theta_k)\right). \tag{5.17}$$

Simulation results

The efficiency of procedure 3 will be illustrated through three examples. We consider the same source images as above. Moreover, we will deal with a real case from the recto-verso separation problem.

For separating the mixtures of Lena and Barbara, we apply procedure 3 using the seven candidate models mentioned in Section 5.2. It follows that the best separation has been obtained through AMH copula model with $\theta = -0.7284$. Figure 5.11 shows the restored images using procedure 3 and MI method, while Figure 5.13a illustrate the SNR mean of each copula model compared to the MI one. We can see that all copula models give better results than the MI method, and that the AMH copula model leads to the best separation. Figures 5.13b and 5.14a illustrate, respectively, the convergence to 0 of the criterion value, and the convergence of $\widehat{\tau}(\boldsymbol{Y})$ values to $\widehat{\tau}(\boldsymbol{S})$. Table 5.3 summarizes the PSNR and the SNR final values of each copula model and the other methods. The final separation criterion and $\widehat{\tau}(\boldsymbol{Y})$ values are listed in Table 5.4.

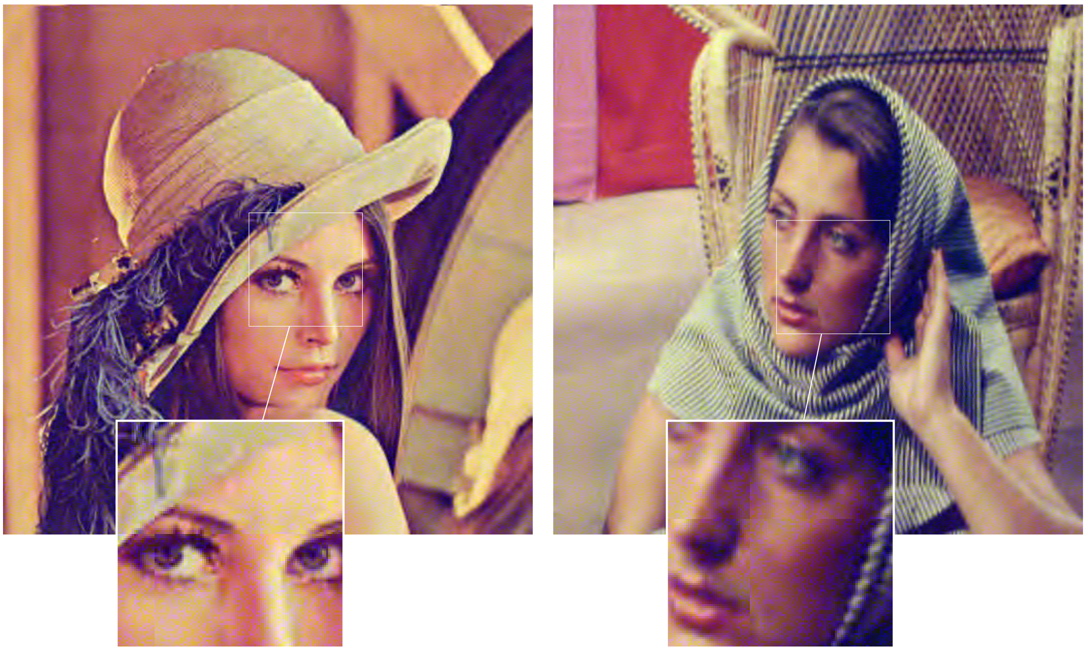


Figure 5.11: Separation results: Procedure 3 with no prior information

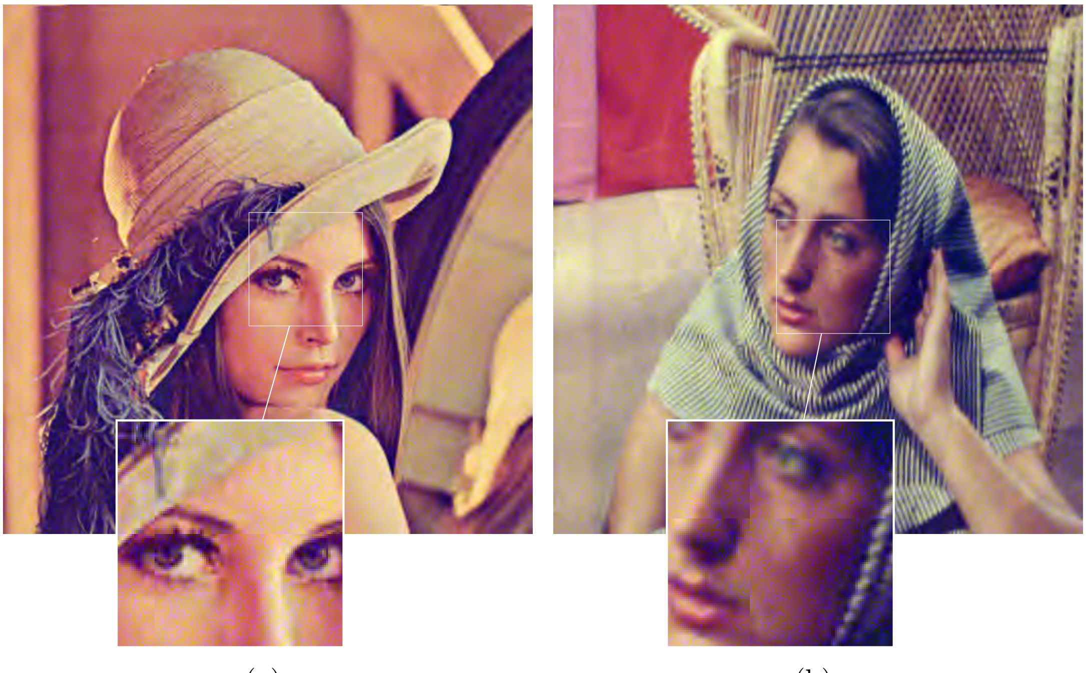
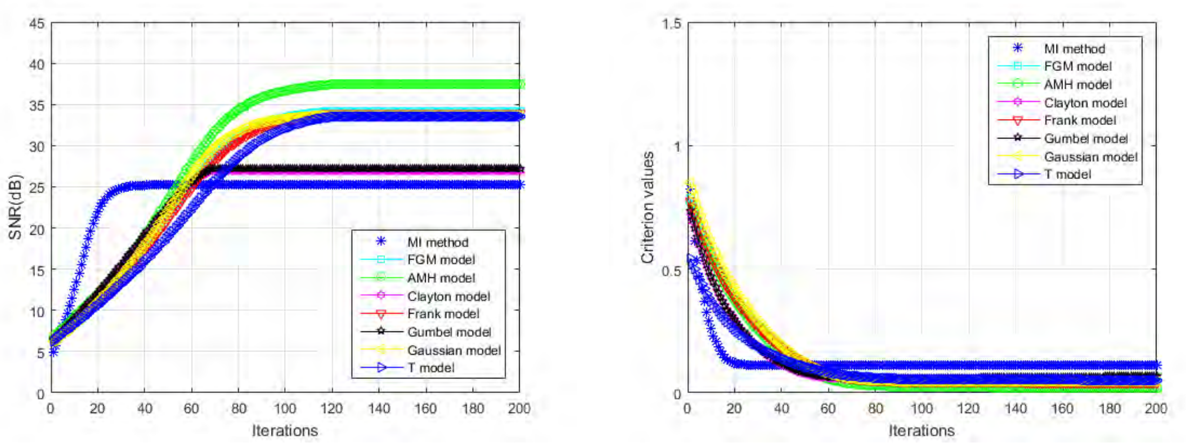
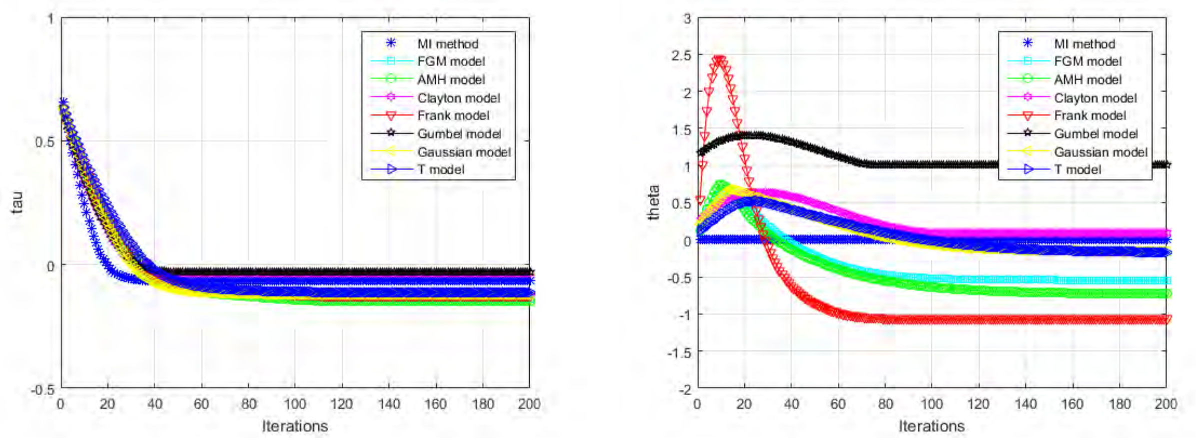


Figure 5.12: Separation results: MI method



(a) (b)

Figure 5.13: (a) SNRs vs iterations. (b) Criterion values vs iterations



(a) (b)

Figure 5.14: (a) $\widehat{\tau}(\boldsymbol{Y})$ values vs iterations. (b) θ values vs iterations

	SNR (dB)		PSNR (dB)	
Methods	Lena	Barbara	Lena	Barbara
MI	25.5387	24.9402	34.6560	34.1124
FGM	34.6091	33.8686	40.1410	40.4089
AMH	**37.6272**	**37.2252**	**42.7882**	**42.7286**
Clayton	27.2660	26.5751	36.7647	36.2616
Frank	34.1103	33.7709	9.9866	38.9037
Gumbel	27.5979	26.7573	35.6664	35.1592
Gaussian	34.0399	33.3673	39.5270	38.1801
T	33.8032	33.1856	38.7655	37.6198
JADE	23.7238	24.3171	31.7751	32.8690
RADICAL	26.5265	22.0898	33.7739	31.17679
FastICA	34.3417	14.9425	21.4409	9.3497
InfoMax	6.4371	6.2938	13.0413	11.6051
Sobi	15.1169	15.4384	19.9217	20.7224

Table 5.3: PSNR and final values of SNR for Procedure 3 : Lena and Barbara

Final values	Criterion	$\widehat{\tau}(\boldsymbol{Y})$
MI	0.1123	-0.0659
FGM	0.0422	-0.1372
AMH	**0.0204**	**-0.1489**
Clayton	0.0575	-0.0340
Frank	0.0422	-0.1261
Gumbel	0.0674	-0.0307
Gaussian	0.0456	-0.1236
T	0.0524	-0.1123

Table 5.4: Final values of the criterion and $\widehat{\tau}(\boldsymbol{Y})$ for each method : Lena and Barbara

5.3 Conclusion

A new method for separating independent and dependent sources has been presented. Iterative algorithms are used to distinguish the copula densities, with the aim of minimizing appropriate criteria. Even when the dependency structure of the source components and/or the relevant parameter are unknown, the proposed algorithms will magnificently separate instantaneous mixtures of independent/dependent source components. The proposed methodology's ability to separate dependent sources is shown by experimental results.

Chapter 6

BSS for noisy mixtures using BTV

6.1 Introduction

The BSS issue BSS in the noisy environment is much more complex and difficult then the noise free environment, therefore some regularization techniques needed to be implemented to overcome this problem. Hence the idea here is to fuse the estimation of Kullback–Leibler divergence between the copula densities and the total variation (TV) regularization, the authors in [69] introduced a new BSS technique to separate statistically dependent or independent sources that were mixed in presence of noise, however, using total variation in order to remove noise has its deficiencies as the sensitivity to noise and easiness to blur. Therefore, in order to overcome this issues, we proposed an extension of the said work in [69] by introducing a method based on the bilateral total variation regularization (BTV) which is generated from the bilateral filter [149]. This filter eliminates the noise completely and retains the edge information. The suggested method relies on denoising the observed signals by using a bilevel minimization where firstly we minimize the observed signals by adding the BTV term then minimizing the Kullback–Leibler divergence between the copula densities.

6.2 Proposed approach

Our method is split into two parts: firstly, we employ the BTV regularization technique in order to denoise the observed $\overline{\boldsymbol{o}}(t)$ signals, and secondly, we use the Kullback–Leibler divergence between copulas densities after the regularization criterion based on BTV to find the approximate $\boldsymbol{z}$ sources. More details are given below regarding these two phases.

6.2.1 Denoising the observed data

Denoising is a signal processing method that extracts signal from a mixture of signal and noise thus preserving the useful information. The commonly used models for denoising problems is stated as follows. We consider

$$\overline{\boldsymbol{o}}(t) := (\overline{o}_1(t), \dots, \overline{o}_p(t))^\top, \quad t \in [0, T], \tag{6.1}$$

denotes the observations contaminated by noise, which takes the following form:

$$\overline{\boldsymbol{o}} := \boldsymbol{o} + \boldsymbol{\nu}, \tag{6.2}$$

where $\boldsymbol{o}(t) := (o_1(t), \dots, o_p(t))^\top, \quad t \in [0, T]$, present the noise-free observed signal, and $\boldsymbol{\nu}$ denotes the additive noise. The aim of this first step is to extract the noise-free signal $\boldsymbol{o}(t)$ from the noise contaminated data $\overline{\boldsymbol{o}}(t)$, under the condition of having the smallest error. To do so, we can use the criterion of the mean squared error, that is to say, we'll reach an approximation of $\boldsymbol{o}(t)$ by solving the least-square problem:

$$\inf_{\boldsymbol{o}} \frac{1}{2} \int_{[0,T]} |\overline{\boldsymbol{o}}(t) - \boldsymbol{o}(t)|^2 \mathrm{d}t, \tag{6.3}$$

As far as (6.3) is an ill-posed problem, obviously the remedy to this issue is to add a regularization that allows us to fulfil the Hadamard's conditions in order of having a well-posed problem. Lately, several regularization technique with the aim of noise free signals as well as images that are contaminated by the noise have been proposed in the literature, namely, the Tikhonov regularization [148] and total variation regularization [26, 134]. We present in this article, a regularization based on BTV-term which has been extracted from the bilateral filter. This last was firstly introduced in [149] as an efficient one-pass filter with the aim of denoising and simultaneously preserving edges. In similar way to Gaussian convolution, it is known as a weighted average of nearby points. The difference resides on the consideration taken by the bilateral filter of the difference in value with the neighbours to preserve edges while smoothing. More precisely the key idea of the bilateral filter is that for a point to influence another point, it should not only occupy a nearby location but also have a similar value. The robust regularizer, called BTV and denoted $BTV(\cdot)$, suggested by [58], is considered in the upcoming lines due to several advantages namely, the ability to smooth away the noise and small variation in a signal while preserving the major edges or discontinuity, and the ability of handling and removing high level noise unlike total variation TV, see for instance [58]. The BTV is then expressed as follows:

$$BTV(\boldsymbol{o}) := \sum_{j=-m}^{m} \alpha^{|j|} \|\boldsymbol{o} - \boldsymbol{G}^j \boldsymbol{o}\|_1, \tag{6.4}$$

where the matrix $\boldsymbol{G}^j$ entails a shift right of j samples and m is the spatial window size. To give a spatial decay effect, one can deploy a scalar weight α $(0 < \alpha < 1)$ to the sum of the regularization terms. One can easily demonstrate that the BTV regularization is a generalization of the other regularizations, for instance if took $m = 1$, with $\alpha = 1$, and define the operator $\boldsymbol{Q} = \boldsymbol{I} - \boldsymbol{G}$ as a first derivative term, then equation (8.17) becomes

$$BTV(\boldsymbol{o}) := \|\boldsymbol{Q}\boldsymbol{o}\|_1, \tag{6.5}$$

which coincides with the total variation regularization. by minimizing the following objective function after applying the BTV regularization, one can easily obtain the noise free signals $\boldsymbol{o}$ (1)

$$\inf_{\boldsymbol{o}} \left\{ \frac{1}{2} \int_{[0,T]} \|\bar{\boldsymbol{o}} - \boldsymbol{o}\|^2 \mathrm{d}t + \lambda \sum_{j=-m}^{m} \alpha^{|j|} \|\boldsymbol{o} - \boldsymbol{G}^j \boldsymbol{o}\|_1 \right\}, \tag{6.6}$$

where λ is the regularization parameter, and also denotes a positive constant that measures the performance of the smoothing effect. The objective function (6.6) is divided into two parts the first one is the fidelity term, while the second one is a smoothing term which has the role of controlling the variations of $\boldsymbol{o}$. Since we use the BTV as a regularizer, the suitable space to minimize the objective function is the space of functions of bounded variations (BV), we refer the readers to [10], for more details and also the modelling theory and the treatment of the problem.

6.2.2 BSS-separation procedure

Once having the noise-free observed signals $\boldsymbol{o}(t) := (o_1(t), \dots, o_p(t))^\top$ deduced from (6.6), our aim now is restoring an estimate for the source signals $\widehat{\boldsymbol{s}}(t) := \widehat{\boldsymbol{W}}\, \boldsymbol{o}(t)$. Therefore, our model consists in finding the separating matrix, thus estimating the source signals, with a minimal noise, by minimizing, in regards to $\boldsymbol{W}$ an approximate with a functional, who has the following form

$$\boldsymbol{W} \mapsto \mathcal{C}(\boldsymbol{W}) := \mathcal{C}_{\text{sep}}(\boldsymbol{W}) + \mathcal{C}_{\text{reg}}(\boldsymbol{z}), \tag{6.7}$$

where $\boldsymbol{z}(t) := \boldsymbol{W}\, \boldsymbol{o}(t) \in \mathbb{R}^p$,, as mentioned above,this method contains two terms, the first one $\mathcal{C}_{\text{sep}}(\cdot)$ is dedicated to the separating part, while the second one is $\mathcal{C}_{\text{reg}}(\boldsymbol{z})$ employed for the regularization part, and this last is applicable in both cases whether we have an independent or dependent source components, and it's expressed as follows:

$$\mathcal{C}_{\text{reg}}(\boldsymbol{z}) := \frac{\gamma}{2T} \int_0^T \|\boldsymbol{z}(t) - \overline{\boldsymbol{z}}(t)\|^2 \, dt + \mu \sum_{j=-m}^{m} \alpha^{|j|} \|\boldsymbol{z}(t) - \boldsymbol{G}^j \boldsymbol{z}(t)\|_1, \; \gamma > 0, \; \mu > 0, \tag{6.8}$$

where $\mathcal{C}_{\text{reg}}(\boldsymbol{z})$ is divided into two terms the first one is a data fidelity term and the second one is a regularization term. The numbers γ and μ presented as regularization parameters that should be adopted carefully by the authors, while the separating term $\mathcal{C}_{\text{sep}}(\cdot)$, in (6.7), measures the dependency between the copula densities, its form relays on whether the source components are dependent or independent. In the upcoming lines, we will study carefully both cases to better understand the BSS process. Firstly, we'll assume the next stochastic modelling of the continuous time signals $\boldsymbol{s}(t) \in \mathbb{R}^p$, $\boldsymbol{o}(t) \in \mathbb{R}^p$ and $\boldsymbol{z}(t) := \boldsymbol{W}\, \boldsymbol{o}(t)$, $t \in [0, T]$. The aforementioned last random procedures are considered to be stationary, thereby the matching sampled versions, with specific time period T_e ,

$$\boldsymbol{s}(i) := \boldsymbol{s}(iT_e), \; \boldsymbol{o}(i) := \boldsymbol{o}(iT_e) \text{ and } \boldsymbol{z}(i) := \boldsymbol{z}(iT_e) = \boldsymbol{W}\, \boldsymbol{o}(i), \; i = 1, \dots, N, \tag{6.9}$$

may be assumed as random vectors's realizations in $\mathbb{R}^p$, that will be denoted respectively:

$$\boldsymbol{S}, \quad \boldsymbol{O} \quad \text{and} \quad \boldsymbol{Z} := \boldsymbol{W}\boldsymbol{O}, \tag{6.10}$$

The estimation of the mixing matrix and the sources for the both cases (dependent / independent), can be easily achieved by having a prior knowledge on the copula density of the random source vector $\boldsymbol{S}$, as it is proven by the authors in [69, 96]. The joint d.f. of the random vector $\boldsymbol{Z} := (Z_1, \dots, Z_p)^\top \in \mathbb{R}^p$, and its marginal d.f.'s, that are continuous, and expressed respectively as follows:

$$\mathbb{F}_{\boldsymbol{Z}}(\cdot) : \; \boldsymbol{z} \in \mathbb{R}^p \mapsto \mathbb{F}_{\boldsymbol{Z}}(\boldsymbol{z}) := \mathbb{F}_{\boldsymbol{Z}}(z_1, \dots, z_p) := \mathbb{P}(Z_1 \leq z_1, \dots, Z_p \leq z_p), \tag{6.11}$$

$$F_{Z_i}(\cdot) : \; z_i \in \mathbb{R} \mapsto F_{Z_i}(z_i) := \mathbb{P}(Z_i \leq z_i), \; \forall i = 1, \dots, p. \tag{6.12}$$

We denote by $f_{\boldsymbol{Z}}(\cdot)$ the probability density of the random variable $\boldsymbol{Z}$, which may exist or not, and also $f_{Z_1}, \ldots, f_{Z_p}$ are the marginal probability densities of $Z_1, \ldots, Z_p$ respectively, then we are allowed to define the mutual information (MI) of $\boldsymbol{Z}$ given as follows:

$$\mathbb{MI}(\boldsymbol{Z}) := \int_{\mathbb{R}^p} \log \frac{f_{\boldsymbol{Z}}(\boldsymbol{z})}{\prod_{i=1}^p f_{Z_i}(z_i)} f_{\boldsymbol{Z}}(\boldsymbol{z})\, d\boldsymbol{z} = \mathbb{E}\left(\log \frac{f_{\boldsymbol{Z}}(\boldsymbol{z})}{\prod_{i=1}^p f_{Z_i}(Z_i)} \right) = \mathbb{K}\left(f_{\boldsymbol{Z}}, \prod_{i=1}^p f_{Z_i} \right), \tag{6.13}$$

where $\mathbb{E}(\cdot)$ denotes the mathematical expectation, and as we can notice, the mutual information equals to the Kullback–Leibler divergence between the joint density $f_{\boldsymbol{Z}}(\cdot)$ and the marginal probability densities Moreover, the term $\mathbb{MI}(\boldsymbol{Z})$ is non-negative and contains a logarithm function that means it reached its minimum value zero if and only if $f_{\boldsymbol{Z}}(\cdot) = \prod\limits_{i=1}^p f_{Z_i}(\cdot)$, that is to say, we reached the independence between the vector $\boldsymbol{Z}$'s elements. In the upcoming lines, we will express the form of the separating term $\mathcal{C}_{\text{sep}}(\cdot)$ of the objective function (6.7) taking into consideration whether the source components are independent or dependent.

The case of independent source components

In this paragraph, we are considering the case where the source components are statistically independent. Hence, we can show that the Mutual Information (MI) of Z, $\mathbb{MI}(\boldsymbol{Z})$, can be written as the Kullback-Leibler divergence between the copula density c_Z of the random vector Z and the copula density $c_\prod$ of independence. In fact, by changing variables in the integral and using the relation (3.8) for the equation (6.13), we can write

$$\begin{aligned}
MI(\boldsymbol{Z}) &:= \int_{\mathbb{R}^p} \log \left(\frac{f_{\boldsymbol{Z}}(\boldsymbol{z})}{\prod\limits_{i=1}^p f_{Z_i}(z_i)} \right) f_{\boldsymbol{Z}}(\boldsymbol{z})\, d\boldsymbol{z} \\
&:= \int_{[0,1]^p} \log \left(\frac{\boldsymbol{c}_{\boldsymbol{Z}}(\boldsymbol{u})}{1} \right) \boldsymbol{c}_{\boldsymbol{Z}}(\boldsymbol{u})\, d\boldsymbol{u} \\
&:= \int_{[0,1]^p} \log \left(\frac{\boldsymbol{c}_{\boldsymbol{Z}}(\boldsymbol{u})}{\boldsymbol{c}_{\prod}(\boldsymbol{u})} \right) \boldsymbol{c}_{\boldsymbol{Z}}(\boldsymbol{u})\, d\boldsymbol{u} \qquad (6.14) \\
&:= KL\left(\boldsymbol{c}_{\boldsymbol{Z}}, \boldsymbol{c}_{\prod}\right) \\
&:= \mathbb{E}\left[\log \left(\boldsymbol{c}_{\boldsymbol{Z}}(F_{Z_1}(Z_1), \ldots, F_{Z_p}(Z_p)) \right) \right], \qquad (6.15)
\end{aligned}$$

with $c_{\boldsymbol{Z}}(\cdot)$ the copula density of $\boldsymbol{Z}$, $c_\prod(\cdot) := \mathbb{1}_{[0,1]^p}(\cdot)$ the copula density of independence. As aforementioned, the Mutual information of a r.v. $\boldsymbol{Z}$ may be viewed as the divergence of Kullback–Leibler between the copula density $c_{\boldsymbol{Z}}(\cdot)$ of the r.v. $\boldsymbol{Z}$ and the copula density $c_\prod(\cdot)$ of independence. Additionally, we have $\mathbb{MI}(\boldsymbol{Z}) = \mathbb{K}\left(c_{\boldsymbol{Z}}, c_\prod\right)$ is always positive, and only reaches the minimum, zero if and only if $c_{\boldsymbol{Z}}(\boldsymbol{v}) = c_\prod(\boldsymbol{v})$, $\forall \boldsymbol{v} \in [0,1]^p$, in other words, if and only if we reach the independence of the elements of the vector $\boldsymbol{Z}$. Consequently the term of separation is expressed in this case as follows:

$$\mathcal{C}_{\text{sep}}(\boldsymbol{W}) := \mathcal{C}_{\text{sep}}^{\text{ind}}(\boldsymbol{W}) := \mathbb{K}\left(c_{\boldsymbol{Z}}, c_{\Pi}\right) \tag{6.16}$$

Hence the function $\boldsymbol{W} \mapsto \mathcal{C}_{\text{sep}}^{\text{ind}}(\boldsymbol{W})$ is also non-negative and reaches its minimum value zero iff the estimate $\boldsymbol{W}$ reaches the inverse of the mixing matrix $\boldsymbol{M}$, in other words $\boldsymbol{W} = \boldsymbol{M}^{-1}$ with indeterminacies of scale and permutation of rows.

The dependent source components case

For the case of statistical dependence between the source components, the prior knowledge about the random source vector $\boldsymbol{S}$ copula density is assumed to be obtained. To do so, one must train the sample $\boldsymbol{S}$ through a model selection procedure, for instance see [27], amid semiparametric copula density models $\{c_\theta(\cdot);\, \theta \in \Theta \subset \mathbb{R}^d\}$, that are well known by a multivariate parameter θ, we refer the readers to [88] and [120] for more details. In order to estimate the multivariate parameter θ, one must use the maximum semiparametric likelihood, see for instance [68] and [151]. We designate by $c_{\widehat{\theta}}(\cdot)$ the copula density that models the dependency between the source elements where $\widehat{\theta}$ is the resulted value of θ. As we did in the previous case where we had the independence structure of the source components, we will replace the copula density of independence (6.15), $c_{\Pi}(\cdot)$ by the semiparametric copula model that estimates the copula density of the signals source $c_{\widehat{\theta}}(\cdot)$ and provide the following separating term $\mathcal{C}_{\text{sep}}(\cdot)$:

$$\mathcal{C}_{\text{sep}}(\boldsymbol{W}) := \mathcal{C}_{\text{sep}}^{\text{dep}}(\boldsymbol{W}) := \int_{[0,1]^p} \log\left(\frac{c_{\boldsymbol{Z}}(\boldsymbol{v})}{c_{\widehat{\theta}}(\boldsymbol{v})}\right) c_{\boldsymbol{Z}}(\boldsymbol{v})\,\mathrm{d}\boldsymbol{v} := \mathbb{E}\left(\log \frac{c_{\boldsymbol{Z}}(F_{Z_1}(Z_1),\ldots,F_{Z_p}(Z_p))}{c_{\widehat{\theta}}(F_{Z_1}(Z_1),\ldots,F_{Z_p}(Z_p))}\right), \tag{6.17}$$

One can easily deduce that the separating term $\mathcal{C}_{\text{sep}}(\cdot)$ equals to the Kullback–Leibler divergence $\mathbb{K}(c_{\boldsymbol{Z}}, c_{\widehat{\theta}})$ between the copula densities $c_{\boldsymbol{Z}}(\cdot)$ and $c_{\widehat{\theta}}(\cdot)$. Therefore, one may easily prove the non-negativity of the function $\boldsymbol{W} \mapsto \mathcal{C}_{\text{sep}}^{\text{dep}}(\boldsymbol{W})$ and reaches its minimum value iff $\boldsymbol{W} = \boldsymbol{M}^{-1}$ with indeterminacies of scale and permutation of rows, provided that the following assumption on the copula density $c_{\widehat{\theta}}(\cdot)$ of the source $\boldsymbol{S}$ is satisfied: if the copula density of the r.v. $\boldsymbol{B}\,\boldsymbol{S}$ for any regular matrix $\boldsymbol{B}$, matches $c_{\widehat{\theta}}(\cdot)$, it means that $\boldsymbol{B} = \boldsymbol{D}\,\boldsymbol{P}$, where $\boldsymbol{D}$ is some diagonal matrix and $\boldsymbol{P}$ a permutation matrix.

The demixing matrix, can be computed using three approaches, depending on our knowledge about the dependency structure of the source components.

- **The model and the parameter are known**

In this case, we estimate $\boldsymbol{W}$ the de-mixing matrix in a straightforward way, as follows:

$$\boldsymbol{W} := \arg\inf_{\boldsymbol{W}} \mathcal{C}_{\text{sep}}^{\text{dep}}(\boldsymbol{W}). \tag{6.18}$$

- **The model is known and the parameter is unknown**

In the second case where the parameter θ of the dependency model is unknown, we propose to estimate the separation matrix $\boldsymbol{W}$ by

$$\boldsymbol{W} = \arg \inf_{\boldsymbol{W}} \inf_{\boldsymbol{\theta} \in \Theta} \mathcal{C}_{\text{sep}}^{\text{dep}}(\boldsymbol{W}). \tag{6.19}$$

- **The model and the parameter are unknown**

Finally, when we have no knowledge about the parameter or the model, we consider $L_1 := \{\mathbf{c}_{\boldsymbol{s}_{\theta_{\mathbf{1}}}}(\cdot);\, \theta_{\mathbf{1}} \in \mathbf{\Theta_1} \subset \mathbb{R}^{\mathbf{d}}\}, \cdots,\, L_T := \{\mathbf{c}_{\boldsymbol{s}_{\theta_{\mathbf{T}}}}(\cdot);\, \theta_{\mathbf{T}} \in \mathbf{\Theta_T} \subset \mathbb{R}^{\mathbf{d}}\}$ to be T models of copula densities of the source components. After that we apply the method described in the second case for each model, and lastly take the model that minimizes the criterion over all considered models, in other words:

$$\boldsymbol{W} = \arg \inf_{W} \inf_{\boldsymbol{\theta} \in \Theta^{k^*}} KL\left(\boldsymbol{c}_{\boldsymbol{s}_{\boldsymbol{\theta k^*}}}, \boldsymbol{c}_{\boldsymbol{Z}}\right). \tag{6.20}$$

where

$$k^* = \arg \min_{k=1\cdots T} \inf_{W} \inf_{\boldsymbol{\theta_k} \in \Theta_k} \mathcal{C}_{\text{sep}}(\boldsymbol{W}). \tag{6.21}$$

We propose to estimate the separation criterion defined by

$$\mathcal{C}_{\text{sep}}(\boldsymbol{W}) := \inf_{\theta \in \Theta} \mathbb{E}\left(\log \frac{c_{\boldsymbol{Z}}(F_{Z_1}(Z_1), \ldots, F_{Z_p}(Z_p))}{c_\theta(F_{Z_1}(Z_1), \ldots, F_{Z_p}(Z_p))}\right). \tag{6.22}$$

6.3 Discretization and Statistical estimation

Having outlined our method earlier, we will explain how to turn it practical in this paragraph. This can be achieved using numerical techniques as well as statistical ones. We will therefore include the discrete versions of statistical estimates of the separating part of both cases (dependent/independent) (6.16), (6.17), and also their regularization part (6.6), (6.8) in the following

6.3.1 Denoising the discrete observed data

In the following, the transformation of the noise-contaminated observations $\bar{\boldsymbol{o}}(i) = (\bar{o}_1(i), \ldots, \bar{o}_p(i))^\top$, $i = 1, \ldots, N$ to a noise-free observed signals $\boldsymbol{o}(i) = (o_1(i), \ldots, o_p(i))^\top$, $i = 1, \ldots, N$ is presented.

This phase is reached by finding the solution of the optimization problem 6.6 in its discrete version. As far as the convergence is hardly guaranteed by using the Euler-Lagrange equation, one must use the primal dual algorithm to overcome this issue, therefore, we set the notations presented bellow:

$$K = \lambda \sum_{j=-m}^{m} \alpha^{|j|}(I - \boldsymbol{G}^j), \tag{6.23}$$

and

$$\mathcal{F}(K\boldsymbol{o}) = \lambda \sum_{j=-m}^{m} \alpha^{|j|} \|\boldsymbol{o} - \boldsymbol{G}^j \boldsymbol{o}\|_1, \tag{6.24}$$

$$\mathcal{G}(\boldsymbol{o}) = \frac{1}{2} \int_{[0,T]} |\boldsymbol{o}(t) - \bar{\boldsymbol{o}}(t)|^2 \mathrm{d}t. \tag{6.25}$$

The problem (6.6), after using the notations cited earlier, is expressed as follows:

$$\inf_{\boldsymbol{o}} \{ \ \mathcal{G}(\boldsymbol{o}) + \mathcal{F}(K\boldsymbol{o})\}. \tag{6.26}$$

After that, to minimize the problem (6.26) one could use the Primal-Dual algorithm, with $\mathcal{F}$ and $\mathcal{G}$ the convex functions and K the linear operator. Hence, we obtain the following primal-dual problem by the use of the saddle point problem:

$$\inf_{\boldsymbol{o}} \sup_{\boldsymbol{p}} \{\langle K\boldsymbol{o}, \boldsymbol{p}\rangle + \mathcal{G}(\boldsymbol{o}) + \mathcal{F}^*(\boldsymbol{p})\}, \tag{6.27}$$

with $\mathcal{F}^*$ representing the Fenchel-Legendre transform of the function $\mathcal{F}$ and it is expressed in the following manner:

$$\mathcal{F}^*(\boldsymbol{p}) = \sup_{\boldsymbol{o}} \langle \boldsymbol{p}, \boldsymbol{o}\rangle - \mathcal{F}(\boldsymbol{o}), \tag{6.28}$$

and $\boldsymbol{p}$ present a dual variable with $\boldsymbol{p} \in \mathbb{R}^p$. After that, we verify that

$$\mathcal{F}^*(\boldsymbol{p}) = \delta_{\boldsymbol{P}}(\boldsymbol{p}) = \begin{cases} 0 & \boldsymbol{p} \in \boldsymbol{P} \\ +\infty & \boldsymbol{p} \notin \boldsymbol{P}, \end{cases} \tag{6.29}$$

where $\boldsymbol{P} = \{\boldsymbol{p} : \|\boldsymbol{p}\|_\infty \leq 1\}$, yet, one must determine the proximity operator functions $\mathcal{F}^*$ and $\mathcal{G}$ before continuing with the Primal-Dual algorithm. First, we present the operator $(I + \sigma\partial\mathcal{F}^*)(\boldsymbol{p})$ by applying a projection on $\boldsymbol{P}$, denoted as $\Pi_{\boldsymbol{P}}$, in the following manner:

$$(I + \sigma\partial\mathcal{F}^*)^{-1}(\boldsymbol{p}) = \Pi_{\boldsymbol{P}}(\boldsymbol{p}),$$

where

$$\Pi_{\boldsymbol{P}}(\boldsymbol{p}) = \frac{\boldsymbol{p}}{\max(||\boldsymbol{p}||_\infty, 1)},$$

and

$$||\boldsymbol{p}||_\infty = \max_{i,j} |\boldsymbol{p}_{i,j}|.$$

By relying on the definition of the function $\mathcal{G}$, one can express the operator $(I + \tau\partial\mathcal{G})^{-1}(\boldsymbol{o})$ as

$$(I + \tau\partial\mathcal{G})^{-1}(\boldsymbol{o}) = \frac{\boldsymbol{o} + \tau\bar{\boldsymbol{o}}}{1 + \tau}. \tag{6.30}$$

After specifying all we need, we are now ready to implement the problem-associated Primal-Dual algorithm (6.26). We detail this algorithm below

```
Data: $\bar{\boldsymbol{o}}$ the noisy observed signal.
Result: $\boldsymbol{z}$ the noise-free obtained signal.
Initialization: Given $\tau$, $\sigma > 0$, $\eta \in [0, 1]$, $(\boldsymbol{p}^0, \boldsymbol{o}^0) \in \mathbb{R}^n \times \mathbb{R}^n$ and set $\boldsymbol{z}^0 = \boldsymbol{o}^0$.
Do:
    $\boldsymbol{p}^{n+1} = (I + \sigma\partial\mathcal{F}^*)^{-1}(\boldsymbol{p}^{\boldsymbol{n}} + \sigma K\boldsymbol{z}^n)$
    $\boldsymbol{o}^{n+1} = (I + \tau\partial\mathcal{G})^{-1}(\boldsymbol{o}^n - \tau K^\intercal\boldsymbol{p}^{n+1})$
    $\boldsymbol{z}^{n+1} = \boldsymbol{o}^{n+1} + \eta(\boldsymbol{o}^{n+1} - \boldsymbol{o}^n)$
```

Algorithm 1: The denoising step using Primal-Dual algorithm.

and we note by $K^{\intercal}$ the adjoint of the operator K presented as the following

$$K^{\intercal} = \lambda \sum_{j=-m}^{m} \alpha^{|j|}(I - \boldsymbol{G}^{-j})$$

After the end of the denoising part, in which we used the primal-dual algorithm, we obtained the de-noised signals $\boldsymbol{o}$ which are assumed to be our observed signals in the BSS part.

6.3.2 Statistical estimates of the separation terms

The case of independent source components

As stated above, the criterion function (6.7) for independent source components is expressed as follows:

$$\mathcal{C}^{\text{ind}}(\cdot) : \boldsymbol{W} \mapsto \mathcal{C}^{\text{ind}}(\boldsymbol{W}) := \mathcal{C}^{\text{ind}}_{\text{sep}}(\boldsymbol{W}) + \mathcal{C}_{\text{reg}}(\boldsymbol{z}), \tag{6.31}$$

where $\mathcal{C}^{\text{ind}}_{\text{sep}}(\boldsymbol{W})$ is given by, see (8.10),

$$\mathcal{C}^{\text{ind}}_{\text{sep}}(\boldsymbol{W}) := \mathbb{K}\left(c_{\boldsymbol{Z}}, c_{\Pi}\right) = \mathbb{E}\left(\log c_{\boldsymbol{Z}}\left(F_{Z_1}(Z_1), \ldots, F_{Z_p}(Z_p)\right)\right)$$

and

$$\mathcal{C}_{\text{reg}}(\boldsymbol{z}) := \frac{\gamma}{2T}\int_0^T \|\boldsymbol{z}(t) - \overline{\boldsymbol{z}}(t)\|^2\,dt + \mu \sum_{j=-m}^{m} \alpha^{|j|}\|\boldsymbol{z}(t) - \boldsymbol{G}^j\boldsymbol{z}(t)\|_1,\ \gamma > 0,\ \mu > 0,$$

with $\boldsymbol{z}(t) = \boldsymbol{W}\,\boldsymbol{o}(t)$ and $\overline{\boldsymbol{z}} = \boldsymbol{W}\,\overline{\boldsymbol{o}}(t)$. By fusing the stochastic modeling (6.9) with the relation (8.10), we estimate the principle (6.31) by

$$\boldsymbol{W} \mapsto \widehat{\mathcal{C}^{\text{ind}}}(\boldsymbol{W}) := \widehat{\mathcal{C}^{\text{ind}}_{\text{sep}}}(\boldsymbol{W}) + \mathcal{C}_{\text{reg,d}}(\boldsymbol{z}), \tag{6.32}$$

where

$$\mathcal{C}_{\text{reg,d}}(\boldsymbol{z}) := \frac{\gamma}{2N}\sum_{i=1}^{N} \mid \overline{\boldsymbol{z}}(i) - \boldsymbol{z}(i) \mid^2 + \mu\sum_{i=1}^{N}\sum_{j=-m}^{m} \alpha^{|j|} \mid \boldsymbol{z}(i) - \boldsymbol{G}^j\boldsymbol{z}(i) \mid, \tag{6.33}$$

denotes the discrete version of $\mathcal{C}_{\text{reg}}(\boldsymbol{z})$, and $\boldsymbol{W} \mapsto \widehat{\mathcal{C}^{\text{ind}}_{\text{sep}}}(\boldsymbol{W})$, the statistical approximate of the separation term $\boldsymbol{W} \mapsto \mathcal{C}^{\text{ind}}_{\text{sep}}(\boldsymbol{W})$, and is defined by:

$$\boldsymbol{W} \mapsto \widehat{\mathcal{C}^{\text{ind}}_{\text{sep}}}(\boldsymbol{W}) := \frac{1}{N}\sum_{i=1}^{N} \log\left(\widehat{c}_Z(\widehat{F}_{Z_1}(z_1(i)), \ldots, \widehat{F}_{Z_p}(z_p(i)))\right), \tag{6.34}$$

where

$$\widehat{c}_Z(\widehat{F}_{Z_1}(z_1(i)), \ldots, \widehat{F}_{Z_p}(z_p(i))) := \frac{1}{NH_1\cdots H_p}\sum_{\ell=1}^{N}\prod_{j=1}^{p} k\left(\frac{\widehat{F}_{Z_j}(z_j(i)) - \widehat{F}_{Z_j}(z_j(\ell))}{H_j}\right),$$

denotes the kernel approximate that correspond to the copula density $c_{\boldsymbol{Z}}(\cdot)$, and $\widehat{F}_{Z_i}(\cdot)$, $\forall i = 1, \ldots, p$, are the smooth estimates of the marginal d.f. $F_{Z_i}(\cdot)$ of the r.v. Z_i, expressed for any number such as $r \in \mathbb{R}$, by

$$\widehat{F}_{Z_i}(r) := \frac{1}{N}\sum_{\ell=1}^{N} K\left(\frac{r - z_i(\ell)}{h_i}\right),$$

where $K(\cdot)$ denotes the kernel $k(\cdot)$'s primitive, which is a symmetric centered probability density.

We picked up the standard Gaussian probability density as our kernel k(.) in the numerical simulations. However, referring to [125] a better selected choice of the kernel can be employed as long as it survives with the boundary effect. By the use of Silverman's rule of thumb, the bandwidth parameters $H_1, \dots, H_p$ and $h_1, \dots, h_p$ will be selected, see for instance [138], i.e., for all $i = 1, \dots, p$, we assume $H_i = \left(\frac{4}{p+2}\right)^{\frac{1}{p+4}} N^{\frac{-1}{p+4}}\,\widehat{\Sigma}_i$, and $h_i = \left(\frac{4}{3}\right)^{\frac{1}{5}} N^{\frac{-1}{5}}\,\widehat{\sigma}_i$, with $\widehat{\Sigma}_i$ and $\widehat{\sigma}_i$ present the empirical standard deviation of the signals, respectively, $\widehat{F}_{Z_i}(z_i(1)), \dots, \widehat{F}_{Z_i}(z_i(N))$ and $z_i(1), \dots, z_i(N)$. one can estimate the source signal vector $\boldsymbol{s}(j), j = 1, \dots, N$, by

$$\widehat{\boldsymbol{s}}(j) = \widehat{\boldsymbol{W}}\,\boldsymbol{o}(j), j = 1, \dots, N,$$

where

$$\widehat{\boldsymbol{W}} := \arg\inf_{\boldsymbol{W}} \widehat{\mathcal{C}^{\text{ind}}}(\boldsymbol{W}),$$

which can be resolved by the use of a gradient descent algorithm. Actually, simple calculus indicate that, the gradient in $\boldsymbol{W}$ of the approximated principle $\boldsymbol{W} \mapsto \widehat{\mathcal{C}^{\text{ind}}}(\boldsymbol{W})$, may be expressed as follows $\frac{d\widehat{\mathcal{C}^{\text{ind}}}(\boldsymbol{W})}{d\boldsymbol{W}} =$

$$\frac{1}{N}\sum_{n=1}^{N}\frac{\frac{d}{d\boldsymbol{W}}\widehat{c}_{\boldsymbol{Z}}(\boldsymbol{v}(n))}{\widehat{c}_{\boldsymbol{Z}}(\boldsymbol{v}(n))} + \frac{\gamma}{N}\sum_{n=1}^{N}(\boldsymbol{z}(n)-\overline{\boldsymbol{z}}(n))(\boldsymbol{o}(n)-\overline{\boldsymbol{o}}(n))^{\top} + \frac{\mu}{N}\sum_{n=1}^{N}\sum_{j=-m}^{m}\alpha^{|j|}\left(I-\boldsymbol{G}^{-j}\right) sign(\boldsymbol{z}(n)-\boldsymbol{G}^{j}\boldsymbol{z}(n \tag{6.35}$$

where $\frac{d}{d\boldsymbol{W}} := \left(\frac{\partial}{\partial \boldsymbol{W}_{ij}}\right)_{ij}$, $i, j = 1, \dots, p$, $\boldsymbol{v}(n) := (\widehat{F}_{Z_1}(z_1(n)), \dots, \widehat{F}_{Z_p}(z_p(n)))^{\top}$ and,

$$\begin{aligned}\frac{\partial\widehat{c}_{\boldsymbol{Z}}(\widehat{F}_{Z_1}(z(n)), \dots, \widehat{F}_{Z_p}(z_p(n)))}{\partial \boldsymbol{W}_{ij}} &= \frac{1}{NH_1\cdots H_p}\sum_{m=1}^{N}\prod_{j=1, j\neq i}^{p} k\left(\frac{\widehat{F}_{Z_j}(z_j(n)) - \widehat{F}_{Z_j}(z_j(m))}{H_j}\right) \\ &\times k'\left(\frac{\widehat{F}_{Z_i}(z_i(m)) - \widehat{F}_{Z_i}(z_i(n))}{H_i}\right)\frac{1}{H_i}\frac{\partial(\widehat{F}_{Z_i}(z_i(m)) - \widehat{F}_{Z_i}(z_i(n)))}{\partial \boldsymbol{W}_{ij}},\end{aligned} \tag{6.36}$$

with

$$\frac{\partial(\widehat{F}_{Z_i}(z_i(m))}{\partial \boldsymbol{W}_{ij}} = \frac{1}{Nh_i}\sum_{n=1}^{N} k\left(\frac{z_i(n) - z_i(m)}{h_i}\right)(o_j(n) - o_j(m)). \tag{6.37}$$

Arriving at this stage, we present the Algorithm of the proposed approach as follows:

```
Data: $\overline{\boldsymbol{o}}$ the noise free observed signal
Result: $\widehat{\boldsymbol{s}}$ the approximated source signal
Initialization: Calculate $\boldsymbol{o} = \overline{\boldsymbol{o}} - \prod_{\lambda G} \overline{\boldsymbol{o}}$ from Algorithm 1, $\boldsymbol{W}^{(0)} = \boldsymbol{I}_p$,
$\boldsymbol{z}^{(0)} = \boldsymbol{W}^{(0)}\, \boldsymbol{o}$.
Given $\varepsilon > 0$, $\nu > 0$.
Do: Update $\boldsymbol{W}$ and $\boldsymbol{z}$:
    $\boldsymbol{W}^{(q+1)} = \boldsymbol{W}^{(q)} - \nu \frac{d\widehat{\mathcal{C}^{\mathrm{ind}}}(\boldsymbol{W})}{d\boldsymbol{W}}$
$\boldsymbol{z}^{(q+1)} = \boldsymbol{W}^{(q+1)}\, \boldsymbol{o}$.
Until $||\boldsymbol{W}^{(q+1)} - \boldsymbol{W}^{(q)}|| < \epsilon$
$\widehat{\boldsymbol{s}} = \boldsymbol{z}^{(q+1)}$.
```

Algorithm 2: BSS using gradient descent algorithm for separation of noise-free independent source components.

The case of dependent source components

Now, we assume that the source elements are dependent, then the criterion (6.7) is given as follows

$$\mathcal{C}^{\mathrm{dep}}(\cdot) : \boldsymbol{W} \mapsto \mathcal{C}^{\mathrm{dep}}(\boldsymbol{W}) := \mathcal{C}^{\mathrm{dep}}_{\mathrm{sep}}(\boldsymbol{W}) + \mathcal{C}_{\mathrm{reg}}(\boldsymbol{z}), \tag{6.38}$$

where $\mathcal{C}^{\mathrm{dep}}(\boldsymbol{W})$ is expressed by, see (8.28),

$$\mathcal{C}^{\mathrm{dep}}_{\mathrm{sep}}(\boldsymbol{W}) := \int_{[0,1]^p} \log\left(\frac{c_{\boldsymbol{Z}}(\boldsymbol{v})}{c_{\widehat{\theta}}(\boldsymbol{v})}\right) c_{\boldsymbol{Z}}(\boldsymbol{v})\, \mathrm{d}\boldsymbol{v} := \mathbb{E}\left(\log \frac{c_{\boldsymbol{Z}}(F_{Z_1}(Z_1), \ldots, F_{Z_p}(Z_p))}{c_{\widehat{\theta}}(F_{Z_1}(Z_1), \ldots, F_{Z_p}(Z_p))}\right).$$

and

$$\mathcal{C}_{\mathrm{reg}}(\boldsymbol{z}) := \frac{\gamma}{2T}\int_0^T \|\boldsymbol{z}(t) - \overline{\boldsymbol{z}}(t)\|^2\, dt + \mu \sum_{j=-m}^{m} \alpha^{|j|} \|\boldsymbol{z}(t) - \boldsymbol{G}^j \boldsymbol{z}(t)\|_1,\ \gamma > 0,\ \mu > 0,$$

By pairing the stochastic modelling (6.9) and the relation (6.17), we suggest to estimate the principle (6.38) as follows:

$$\boldsymbol{W} \mapsto \widehat{\mathcal{C}^{\mathrm{dep}}}(\boldsymbol{W}) := \widehat{\mathcal{C}^{\mathrm{dep}}_{\mathrm{sep}}}(\boldsymbol{W}) + \mathcal{C}_{\mathrm{reg,d}}(\boldsymbol{z}), \tag{6.39}$$

where

$$\mathcal{C}_{\mathrm{reg,d}}(\boldsymbol{z}) := \frac{\gamma}{2N}\sum_{i=1}^{N} \mid \overline{\boldsymbol{z}}(i) - \boldsymbol{z}(i) \mid^2 + \mu \sum_{j=1}^{N}\sum_{j=-m}^{m} \alpha^{|j|} \mid \boldsymbol{z}(n) - \boldsymbol{G}^j \boldsymbol{z}(n) \mid, \tag{6.40}$$

denotes $\mathcal{C}_{\mathrm{reg}}(\boldsymbol{z})$'s discrete version, and $\widehat{\mathcal{C}^{\mathrm{dep}}_{\mathrm{sep}}}(\cdot) : \boldsymbol{W} \mapsto \widehat{\mathcal{C}^{\mathrm{dep}}_{\mathrm{sep}}}(\boldsymbol{W})$ present the statistical approximate, of the principle $\mathcal{C}^{\mathrm{dep}}_{\mathrm{sep}}(\cdot) : \boldsymbol{W} \mapsto \mathcal{C}^{\mathrm{dep}}_{\mathrm{sep}}(\boldsymbol{W})$, which we present as follows

$$\widehat{\mathcal{C}^{\mathrm{dep}}_{\mathrm{sep}}}(\cdot) : \boldsymbol{W} \mapsto \widehat{\mathcal{C}^{\mathrm{dep}}_{\mathrm{sep}}}(\boldsymbol{W}) := \frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{\widehat{c}_{\boldsymbol{Z}}(\widehat{F}_{Z_1}(z(i)), \ldots, \widehat{F}_{Z_p}(z_p(i)))}{\widehat{c}_{\widehat{\theta}}(\widehat{F}_{Z_1}(z(i)), \ldots, \widehat{F}_{Z_p}(z_p(i)))}\right). \tag{6.41}$$

The source signal vector $\boldsymbol{s}(i), i = 1, \ldots, N$, will be approximated as follows:

$$\widehat{\boldsymbol{s}}(i) = \widehat{\boldsymbol{W}}\, \boldsymbol{o}(i), i = 1, \ldots, N,$$

where

$$\widehat{\boldsymbol{W}} := \arg\inf_{\boldsymbol{W}} \widehat{\mathcal{C}^{\text{dep}}}(\boldsymbol{W}),$$

which will be resolvable by relying on a gradient descent algorithm. As a matter of fact, simple calculus indicate that, the gradient of the approximated criterion, with respect to $\boldsymbol{W}$, $\boldsymbol{W} \mapsto \widehat{\mathcal{C}^{\text{dep}}}(\boldsymbol{W})$, is calculated as follows

$$\begin{aligned}\frac{d\widehat{\mathcal{C}^{\text{dep}}}(\boldsymbol{W})}{d\boldsymbol{W}} &= \frac{1}{N}\sum_{n=1}^{N}\left[\frac{\frac{d}{d\boldsymbol{W}}\widehat{c}_{\boldsymbol{Z}}(\boldsymbol{v}(n))}{\widehat{c}_{\boldsymbol{Z}}(\boldsymbol{v}(n))} - \frac{\frac{d}{d\boldsymbol{W}}\widehat{c}_{\widehat{\theta}}(\boldsymbol{v}(n))}{\widehat{c}_{\widehat{\theta}}(\boldsymbol{v}(n))}\right] \\ &+ \frac{\gamma}{N}\sum_{n=1}^{N}(\boldsymbol{z}(n) - \overline{\boldsymbol{z}}(n))(\boldsymbol{o}(n) - \overline{\boldsymbol{o}}(n))^{\top} + + \frac{\mu}{N}\sum_{n=1}^{N}\sum_{j=-m}^{m}\alpha^{|j|}\left(I - \boldsymbol{G}^{-j}\right) sign(\boldsymbol{z}(n) - \boldsymbol{G}^{j}\boldsymbol{z}(n)),\end{aligned} \tag{6.42}$$

where $\boldsymbol{v}(n) := (\widehat{F}_{Z_1}(z(n)), \ldots, \widehat{F}_{Z_p}(z_p(n)))$; the gradients $\frac{d}{d\boldsymbol{W}}\widehat{c}_{\boldsymbol{Z}}(\boldsymbol{v}(n))$ and $\frac{d}{d\boldsymbol{W}}\widehat{c}_{\widehat{\theta}}(\boldsymbol{v}(n))$ can be calculated, in a likewise manner as aforementioned in Subsection 6.3.2. Then, one can easily deduce the following Algorithm 4.

```
Data: $\overline{\boldsymbol{o}}$ the noise free observed signal
Result: $\widehat{\boldsymbol{s}}$ the approximate source signal
Initialization: Calculate $\boldsymbol{o} = \overline{\boldsymbol{o}} - \prod_{\lambda G}\overline{\boldsymbol{o}}$ from Algorithm 1, $\boldsymbol{W}^{(0)} = \boldsymbol{I}_p$, $\boldsymbol{z}^{(0)} = \boldsymbol{W}^{(0)}\,\boldsymbol{o}$.
Given $\varepsilon > 0$, $\nu > 0$.
Do: Update $\boldsymbol{W}$ and $\boldsymbol{z}$:
    $\boldsymbol{W}^{(q+1)} = \boldsymbol{W}^{(q)} - \nu\frac{d\widehat{\mathcal{C}^{\text{dep}}}(\boldsymbol{W})}{d\boldsymbol{W}}$
    $\boldsymbol{z}^{(q+1)} = \boldsymbol{W}^{(q+1)}\boldsymbol{o}$.
Until $||\boldsymbol{W}^{(q+1)} - \boldsymbol{W}^{(q)}|| < \epsilon$
    $\widehat{\boldsymbol{s}} = \boldsymbol{z}^{(q+1)}$.
```

Algorithm 3: BSS algorithm for separating noise-free dependent source components.

6.4 Simulation results

In order to test the performance of the suggested approach and prove its efficiency, we will expose the outcomes of various simulations that were conducted and we'll measure them with those obtained by a TV, BTV regularization versions of [54] (MI), [23] (JADE), [83] (FastICA), [117] (RADICAL) and [17] (InfoMax) under the same conditions, and those obtained by [69].

In this section we will adopt the number of samples at $N = 3000$, while the mixing matrix is given as follows:

$$\boldsymbol{M} := [1\ 0.7\ 0.7\ ;\ 0.7\ 1\ 0.7\ ;\ 0.7\ 0.7\ 1]$$

In addition, we will adopt a Gaussian noise that is centred where its deviation is equal to 0.01 was appended to the normalized mixtures, to gain two different signal-to-noise ratio (SNR) values: $-25dB$ and $-15dB$. $\nu = 0.1$ is the chosen gradient descent parameter, while the denoising part, see the above Algorithm 1, we adopt $\tau = 0.1$, $\sigma = 0.01$ and $\eta = 0.01$.

Meantime in the other part, we take $\gamma = 0.01$, $\mu = 0.01$ and $\epsilon = 0.001$, see Algorithm 2 or Algorithm 3, while the number of iterations is settled at 100 iterations for all the simulations, and we adopt the following signal-to-residue-ratio criterion, also called the signal-to-noise-ratio, in order to calculate the effectiveness of the source estimation, presented as follows:

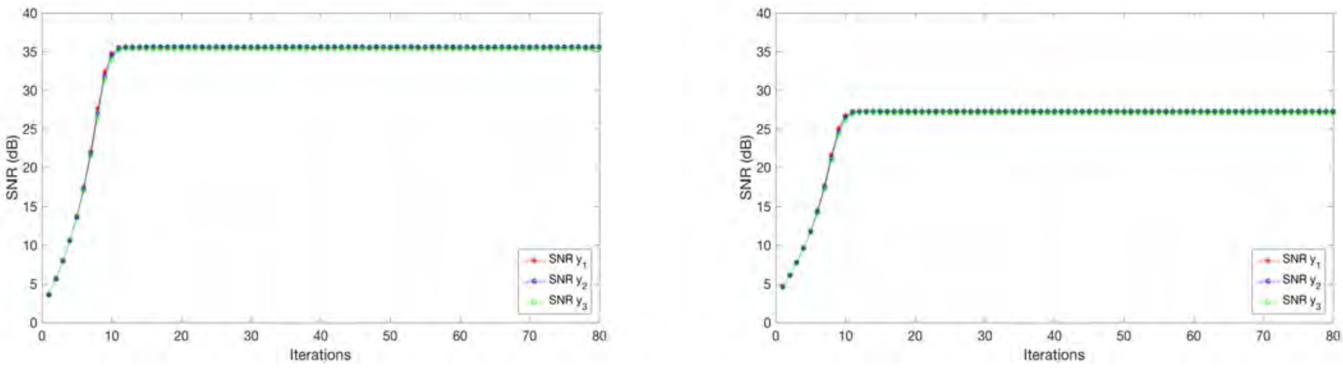
$$SNR_j := 10 \log_{10} \frac{\sum_{m=1}^{N} z_j(m)^2}{\sum_{m=1}^{N} (\widehat{z}_j(m) - z_j(m))^2}, \; j = 1, 2, 3. \tag{6.43}$$

With the aim of exposing the competence of our proposed method, we accomplished numerical simulations for four test signals:

- Uniform i.i.d sources with independent components.
- Uniform i.i.d sources with dependent elements extracted from AMH copula with $\theta = 0.75$.
- Uniform i.i.d sources with dependent elements extracted from Clayton copula, with $\theta = 1.5$
- Uniform i.i.d sources with dependent elements extracted from Frank copula, with $\theta = 2$.

Fig.6.1 shows that our proposed approach (obtained by using algorithm 1 and algorithm 2) is a noise removal technique and well estimating the sources in the independent components case, accompanied with delightful performance at different noise level. When the separation is successfully reached, Fig.6.2 shows that the criterion values (of $\widehat{\mathcal{C}^{\text{ind}}}(\cdot)$) converge to the minimum 0.

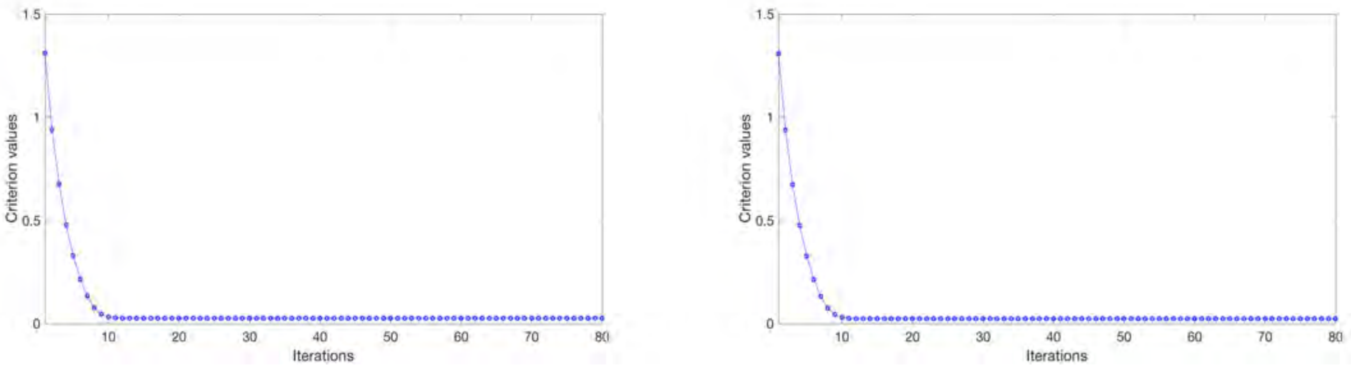
The results of our method are compared with the Copula-TV algorithm presented in [69] and also compared with those of [54], [23] (JADE), [83] (FastICA), [117] (RADICAL) and [17] (InfoMax) penalized by the same BTV and TV-regularization term. Table 6.1 shows the resulted SNR gained for each source



(a) Noise value -25

(b) Noise value -15

Figure 6.1: SNRs values versus iteration number for independent source components



(a) Noise value -25

(b) Noise value -15

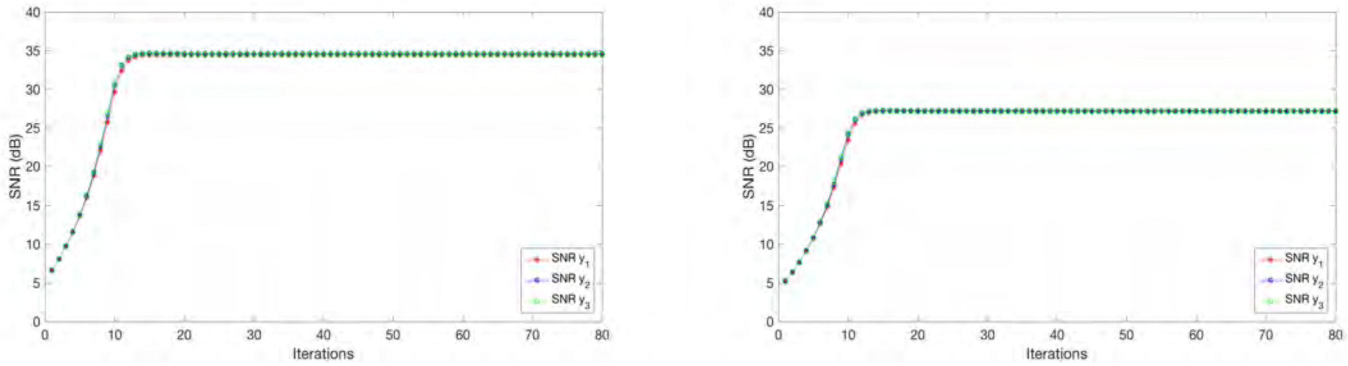
Figure 6.2: Criterion values versus iteration number for independent source components

Noise	-25			-15		
Sources	**S1**	**S2**	**S3**	**S1**	**S2**	**S3**
Our method	**36.5471**	**36.5954**	**36.4158**	**28.2438**	**28.2780**	**28.1510**
Copula-TV	34.8081	34.8543	34.8827	26.7540	26.7878	26.6622
MI-BTV	34.7684	34.7297	34.7244	26.3500	26.3231	26.3194
MI-TV	33.7045	33.6673	33.6622	25.8572	25.8306	25.8269
FastICA-BTV	33.4641	33.8923	30.8436	26.1685	25.7622	24.6918
FastICA-TV	32.9945	32.8525	29.5444	26.0232	25.2849	24.2389
JADE-BTV	34.6533	34.2516	33.4077	26.9526	26.9433	26.9651
JADE-TV	33.6222	33.2407	33.1620	26.1526	26.0425	26.9641
RADICAL-BTV	34.9930	33.5704	34.0972	26.2299	25.9483	26.0727
RADICAL-TV	34.0718	32.9523	34.2743	25.8552	25.7189	25.3581
InfoMax-BTV	35.4062	34.8745	34.9221	26.7912	26.4683	26.3730
InfoMax-TV	34.4247	34.1689	34.5282	26.1014	25.8752	26.0800

Table 6.1: Output SNR's for independent source components

On the other hand, even with the noised dependent source components, our method (obtained by using algorithm 1 and algorithm 3) shows its efficiency by denoising and separating the mixtures dependent source components from Fig.6.3 to Fig.6.5. The convergence of the principle $\widehat{\mathcal{C}^{\text{dep}}}(\cdot)$ to 0 when we obtain the separation is shown in Fig.6.6.

We also compare our experimental result with the result of [69] and those of [54], [23] (JADE), [83] (FastICA), [117] (RADICAL) and [17] (InfoMax), penalized by the same BTV and TV-regularization versions. The comparison results obtained by each method for each source are summarized in Table 6.2 to Table 6.4 .



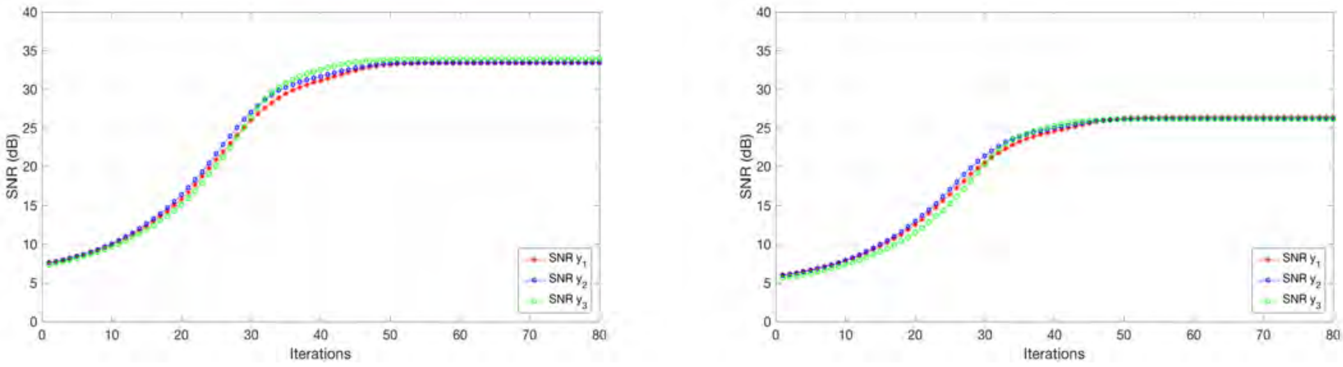
(a) Noise value -25

(b) Noise value -15

Figure 6.3: SNRs values versus iteration number for dependent source components (AMH copula)

Noise	-25			-15		
Sources	**S1**	**S2**	**S3**	**S1**	**S2**	**S3**
Our method	**35.4457**	**35.6201**	**35.5215**	**28.1162**	**28.2098**	**28.1162**
Copula-TV	33.7363	33.7695	33.7507	26.1224	26.2838	26.1926
MI-BTV	13.0724	13.0872	13.0774	10.6702	10.8095	10.6419
MI-TV	12.8616	12.8759	12.8664	10.1909	10.2717	10.2179
FastICA-BTV	32.3677	8.3796	6.9618	25.6886	8.2546	6.5094
FastICA-TV	31.6124	8.0572	6.7939	24.5366	7.9771	6.4478
JADE-BTV	12.8532	12.2388	12.3677	11.0552	10.9488	10.5114
JADE-TV	12.5622	12.1087	12.0597	10.6551	10.4486	10.2116
RADICAL-BTV	12.3677	12.3796	12.9618	11.3589	11.7601	11.2248
RADICAL-TV	12.6124	12.0572	12.7939	11.0552	11.4581	11.0046
InfoMax-BTV	8.3558	8.3269	8.3479	7.8050	7.8691	7.8910
InfoMax-TV	8.3478	8.3189	8.3400	7.7164	7.6805	7.7024

Table 6.2: Output SNR's (dependent components generated from AMH copula)



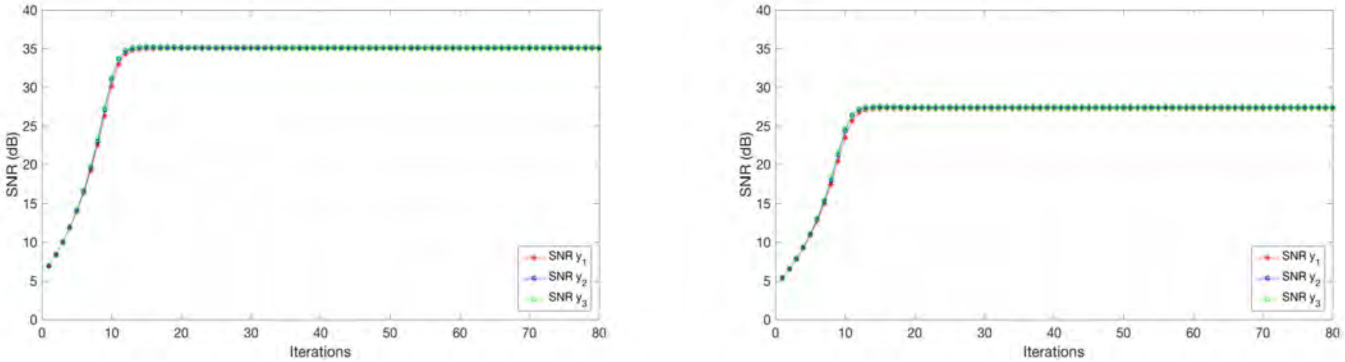
(a) Noise value -25

(b) Noise value -15

Figure 6.4: SNRs values versus iteration number for dependent source components (Clayton copula)

Noise	**-25**			**-15**		
Sources	**S1**	**S2**	**S3**	**S1**	**S2**	**S3**
Our method	**34.4175**	**34.4566**	**34.9370**	**27.3750**	**27.2740**	**27.1677**
Copula-TV	32.7252	32.7326	32.8239	25.3364	25.3726	25.8170
MI-BTV	9.3787	9.5195	9.4826	7.8156	8.4903	8.1958
MI-TV	9.1836	9.3203	9.2845	7.2549	8.0250	7.8233
FastICA-BTV	24.3500	6.3672	3.9238	22.6489	5.4379	3.7233
FastICA-TV	23.9612	5.6319	3.6257	22.6293	5.3838	3.4151
JADE-BTV	9.0835	9.0632	9.7930	8.3348	8.3037	8.4995
JADE-TV	8.3835	8.4631	8.8930	8.1349	8.0037	8.2994
RADICAL-BTV	8.8629	8.3840	9.9952	8.0636	8.7793	8.4024
RADICAL-TV	8.3323	7.9013	9.7054	7.7358	8.4084	8.1157
InfoMax-BTV	2.3558	2.3269	2.3479	2.0396	2.0490	2.0841
InfoMax-TV	2.3478	2.3189	2.3400	2.0318	2.0412	2.0762

Table 6.3: Output SNR's (dependent components generated from Clayton copula)



(a) Noise value -25

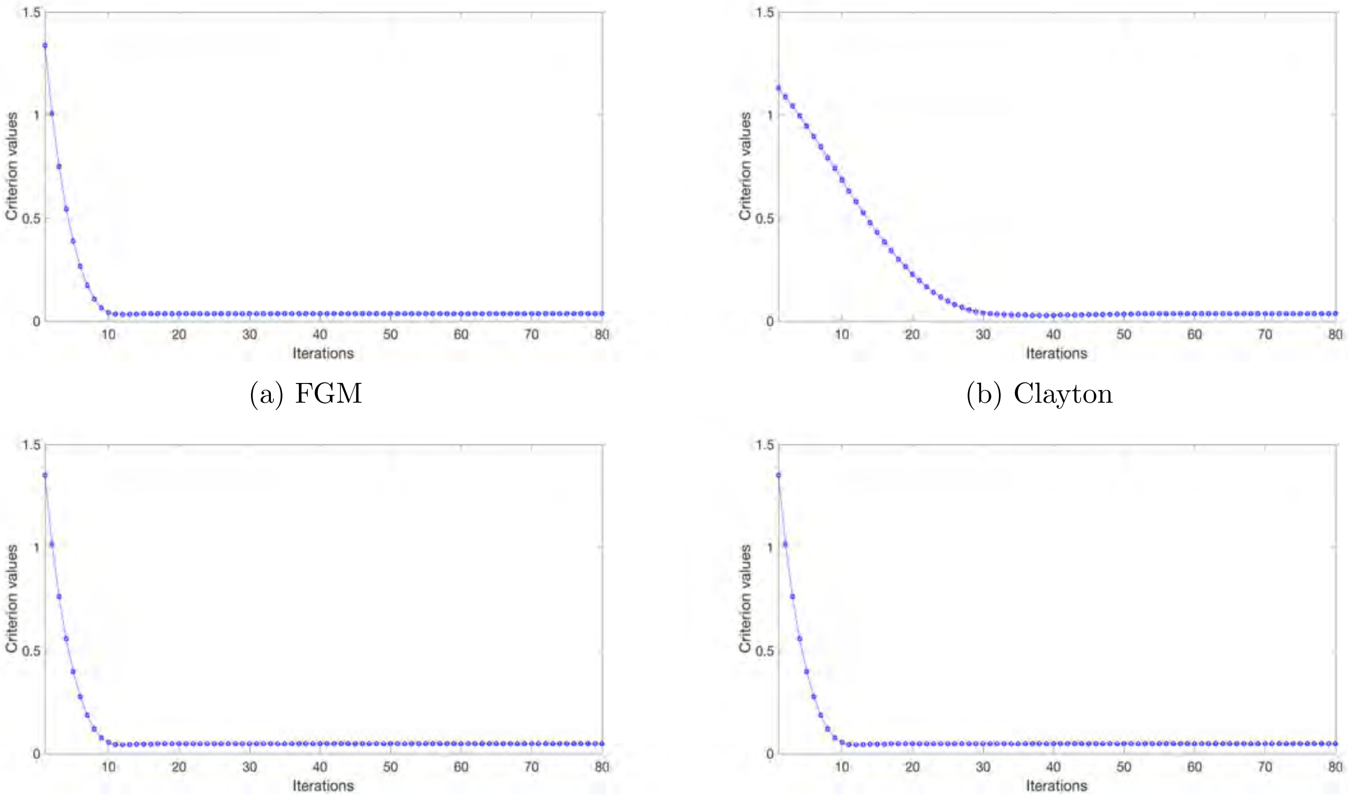
(b) Noise value -15

Figure 6.5: SNRs values versus iteration number for dependent source components (Frank copula)

Noise	**-25**			**-15**		
Sources	**S1**	**S2**	**S3**	**S1**	**S2**	**S3**
Our method	**36.0128**	**36.1876**	**36.0888**	**28.2641**	**28.4388**	**28.3400**
Copula-TV	34.0558	34.0891	34.0703	26.3192	26.3061	26.3135
MI-BTV	15.6846	15.7053	15.6915	12.4442	12.5147	12.3668
MI-TV	15.4543	15.4744	15.4610	11.8674	11.8589	11.8645
FastICA-BTV	34.9595	10.5616	8.3832	24.9059	9.5946	8.1635
FastICA-TV	34.5843	10.1445	7.9430	24.3964	9.7748	7.7252
JADE-BTV	14.3312	14.4466	14.7004	13.2157	13.2842	12.9108
JADE-TV	13.5313	13.6468	13.8002	13.2157	13.2841	12.9108
RADICAL-BTV	14.8603	14.4040	14.6389	12.5809	14.3538	12.6422
RADICAL-TV	13.9417	13.7839	14.0200	13.1350	14.0523	12.4450
InfoMax-BTV	9.6526	9.5702	9.6092	9.4765	9.3793	9.4124
InfoMax-TV	9.5392	9.4571	9.4959	9.4633	9.3664	9.3994

Table 6.4: Output SNR's (dependent components generated from Frank copula)

Fig.6.6 shows that the criterion values (of $\widehat{\mathcal{C}^{\text{dep}}}(\cdot)$) converge to the minimum 0.



(a) FGM

(b) Clayton

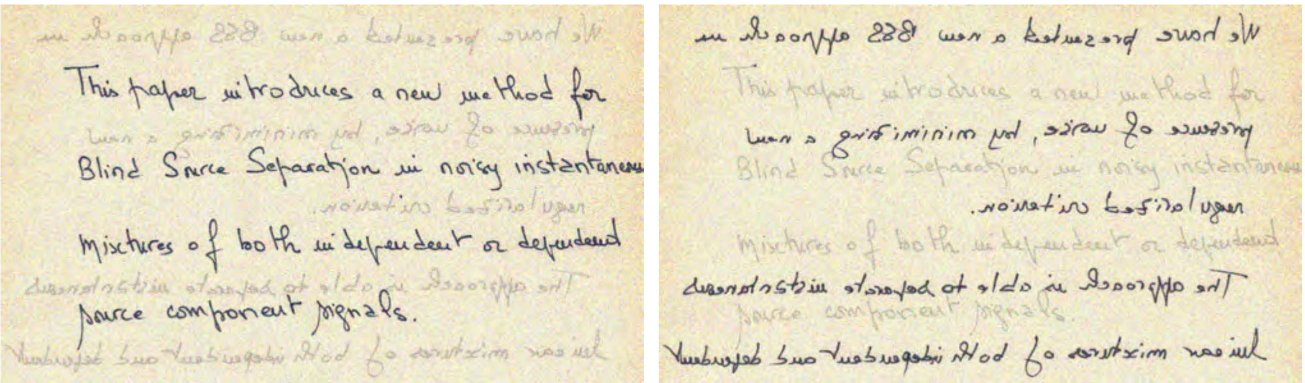
(c) AMH

(d) Frank

Figure 6.6: Criterion values versus iteration number for Dependent source components

The figures Fig.6.3 to Fig.6.6 and the tables Table 6.2 to Table 6.4 show clearly that our approach can perfectly handle the dependent sources case with good performance.

We would like to end this section by giving a real example to showcase the performance of our approach . In this example, we deal with the recto-verso separation problem. The observations ("image1" and "image2"), see Fig.6.7, are obtained by scanning a recto-verso document (at which we have added centered gaussian noise with standard deviation $\sigma = 0.01$). These observations are obviously unknown mixtures of the recto (source 1) and the verso (source 2).

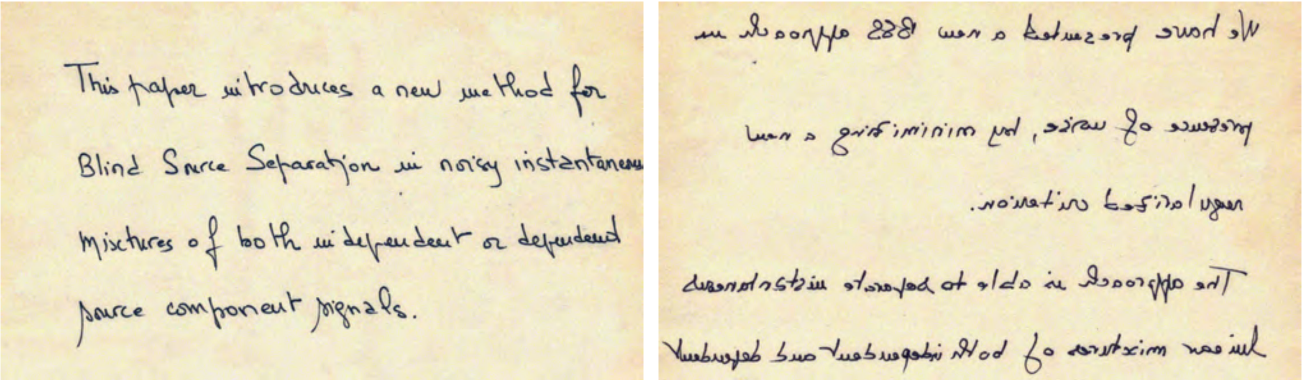

This paper introduces a new method for
Blind Source Separation in noisy instantaneous
mixtures of both independent or dependent
source component signals.
We have presented a new BSS approach in
presence of noise, by minimizing a new
regularized criterion.
The approach is able to separate instantaneous
mixtures of both independent and dependent

(a) image 1

(b) image 2

Figure 6.7: The scanned recto and verso of a document contaminated by a Gaussian noise

Procedures 1 and 2 can not be applied in this case, since the dependency structure of the sources is unknown and no training samples of the sources is available. Therefore, we will apply procedure 3 using the same three candidate copula models as before. The optimal model in this case follows the AMH copula with $\theta = 0.7$. Fig.6.8 present the restored images, using procedure 3, the noise was reduced and we can say that the separation was successful.


This paper introduces a new method for
Blind Source Separation in noisy instantaneous
mixtures of both independent or dependent
source component signals.
We have presented a new BSS approach in
presence of noise, by minimizing a new
regularized criterion.
The approach is able to separate instantaneous
mixtures of both independent and dependent

(a) Estimated source 1

(b) Estimated source 2

Figure 6.8: The estimated sources (recto-verso) using the BTV approach

6.5 Conclusion

In this paper a new BSS approach is applied for noisy environments, with a new regularized criterion being minimised. This technique will distinguish and eliminate noise from instantaneous linear mixtures of independent and dependent source components. The method consists of minimizing a criterion that incorporates two types of information: a separation part and a regularization part. The first one is given by Kullback-Leibler divergence between copula densities for separating the observed signals. The second one is the BTV for de-noising the observed data. Various literature's methods are evaluated against our proposed approach, and this last performs far better than the other methods, as experimental results show. In addition, the proposed framework can be enlarged to function in future interchanges with convolutive mixtures, which are independent / dependent.

Chapter 7

Copula based BSS using Alpha-divergence

7.1 Introduction

In this chapter, we will be detailing on our new approach that uses α-divergence as a cost function for the blind source separation problem. As we saw in the previous chapter, minimizing the Kullback-Leibler divergence between copula densities have shown magnificent results, however the Kullback-Leibler divergence is just a special case of the Alpha-divergence. Consequently, we come up with the idea to replace the Kullback-Leibler divergence by the alpha-divergence as our cost function to minimize, considering its superiority to handle noisy data as well as its ability to converge faster. We test our approach for various values of alpha and give a comparative study between the proposed methodology and other existing methods. This method outperformed the others in terms of quality and accuracy, especially when α is set to $\frac{1}{2}$, which corresponds to the Hellinger divergence.

7.2 The proposed approach

As stated in the State of art chapter, in section 2.3, the goal of BSS consists in constructing a separation system, denoted B, which allows the best possible estimation of the initial sources s from the sole knowledge of the observations (source mixtures). The ideal solution for this problem is when $W = A^{-1}$, where A is the mixing matrix, in other words, finding the demixing matrix W. To achieve a solution that is as close as possible to A^{-1} we will use the α-divergence as our cost function. Applying the change variable formula for multiple integrals, using the copula formula (3.8), the alpha divergence $D_\alpha\left(\prod_{i=1}^{p} f_{Z_i}, f_{\boldsymbol{Z}}\right)$ could be defined as follows

$$\begin{aligned} D_\alpha\left(\prod_{i=1}^{p} f_{Z_i}, f_{\boldsymbol{Z}}\right) &:= \int_{[0,1]^p} \varphi_\alpha\left(\frac{1}{\boldsymbol{c}_{\boldsymbol{Z}}(\boldsymbol{v})}\right) \boldsymbol{c}_{\boldsymbol{Z}}(\boldsymbol{v}) d\boldsymbol{v}, \\ &:= \mathbb{E}\left[\varphi_\alpha\left(\frac{1}{\boldsymbol{c}_{\boldsymbol{Z}}(F_{Z_1}(Z_1), \ldots, F_{Z_p}(Z_p))}\right)\right]. \end{aligned} \tag{7.1}$$

where the α-divergence between the product density of the marginal densities and the joint density of the random vector $\boldsymbol{Z}$ is equivalent to the α-divergence between the copula density of independent $\boldsymbol{c}_{\Pi}$, and copula density $\boldsymbol{c}_{\boldsymbol{Z}}$ of the random vector $\boldsymbol{Z}$.

$$D_{\alpha}\left(\prod_{i=1}^{p} f_{Z_i}, f_{\boldsymbol{Z}}\right) := D_{\alpha}\left(\boldsymbol{c}_{\Pi}, \boldsymbol{c}_{\boldsymbol{Z}}\right) \tag{7.2}$$

In the following, we study the case of mixture of independent source components and mixture of dependent source components, separately.

7.2.1 Independent source components

The α-divergence $D_{\alpha}\left(\boldsymbol{c}_{\Pi}, \boldsymbol{c}_{\boldsymbol{Z}}\right)$ of independent source components, is always nonnegative and reaches its minimum value zero if and only if $\boldsymbol{W} = \boldsymbol{A}^{-1}$ with indeterminacies up to scale and permutation:

$$\boldsymbol{A}^{-1} := \arg \inf_{\boldsymbol{W}} D_{\alpha}\left(\boldsymbol{c}_{\Pi}, \boldsymbol{c}_{\boldsymbol{Z}}\right). \tag{7.3}$$

Minimizing upon $\boldsymbol{W}$ a nonparametric estimate $\widehat{D}_{\alpha}\left(\boldsymbol{c}_{\Pi}, \boldsymbol{c}_{\boldsymbol{Z}}\right)$ from $\boldsymbol{Z}(1) \cdots \boldsymbol{Z}(N)$, will lead to estimate the source signals $\widehat{\boldsymbol{Z}}(n) := \widehat{W}\boldsymbol{Y}(i), \forall i = 1 \cdots N$. where $\widehat{\boldsymbol{W}} := \arg \inf_{\boldsymbol{W}} \widehat{D}_{\alpha}\left(\boldsymbol{c}_{\Pi}, \boldsymbol{c}_{\boldsymbol{Z}}\right)$, is the estimated unmixing matrix. We suggest estimating the $\alpha-$divergence $D_{\alpha}\left(\boldsymbol{c}_{\Pi}, \boldsymbol{c}_{\boldsymbol{Z}}\right)$ by:

$$\widehat{D}_{\alpha}\left(\boldsymbol{c}_{\Pi}, \boldsymbol{c}_{\boldsymbol{Z}}\right) := \frac{1}{N}\sum_{i=1}^{N} \varphi_{\alpha}\left(\frac{1}{\hat{\boldsymbol{c}}_{\boldsymbol{Z}}(\hat{F}_{Z_1}(Z_1(i)), \ldots, \hat{F}_{Z_p}(Z_p(i)))}\right), \tag{7.4}$$

where the kernel estimate of the copula density is calculated as follows,

$$\widehat{c}_{\boldsymbol{Z}}(\boldsymbol{v}) := \frac{1}{NH_1 \cdots H_p}\sum_{m=1}^{N}\prod_{j=1}^{p} k\left(\frac{\widehat{F}_{Z_j}(Z_j(m)) - v_j}{H_j}\right), \forall \boldsymbol{v} \in [0,1]^p, \tag{7.5}$$

and $\widehat{F}_{Z_j}(v)$, are the smoothed estimate of the marginal distribution functions of the random variables Z_j for all $j = 1, \ldots, p$, which are defined as shown below

$$\widehat{F}_{Z_j}(u) := \frac{1}{N}\sum_{m=1}^{N} K\left(\frac{Z_j(m) - u}{h_j}\right), \forall j = 1, \ldots, p\, v \in \mathbb{R}, \tag{7.6}$$

with $K(.)$ a symmetric and centered probability density and the primitive of a kernel $k(.)$. For our computational study we chose the standard Gaussian density as the kernel $k(.)$. Further information on the kernel option $k(.)$ can be found in [125] to approximate the copula density, which copes with the boundary effect. To determine the parameters of the bandwidth $H_1, \ldots, H_p$ and $h_1, \ldots, h_p$ which are present in (8.12), for every $j = 1, \ldots, p$, the Silverman's rule of thumb [138] is adopted :

$$\begin{cases} H_j = \left(\dfrac{4}{p+2}\right)^{\frac{1}{p+4}} N^{\frac{-1}{p+4}} \widehat{\Sigma}_j, \\ h_j = \left(\dfrac{4}{3}\right)^{\frac{1}{5}} N^{\frac{-1}{5}} \widehat{\sigma}_j, \end{cases} \tag{7.7}$$

where $\widehat{\Sigma}_j$ and $\widehat{\sigma}_j$ are, the empirical standard deviation of $\widehat{F}_{Z_j}(Z_j(1)), \ldots, \widehat{F}_{Z_j}(Z_j(N))$ and $\boldsymbol{Z}_j(1), \ldots, \boldsymbol{Z}_j(N)$, respectively. The estimate of the de-mixing matrix $\widehat{\boldsymbol{W}}$ is computed using a descent gradient algorithm, where the initial matrix $\boldsymbol{W}_0 = I_p$, is the $p \times p$ identity matrix.

In the following we detail the computations of the gradient in $\boldsymbol{W}$ of $\widehat{D}_\alpha\left(\boldsymbol{c}_\Pi, \boldsymbol{c}_{\boldsymbol{Z}}\right)$

$$\begin{aligned}
\frac{d\widehat{D}_\alpha\left(\boldsymbol{c}_\Pi, \boldsymbol{c}_{\boldsymbol{Z}}\right)}{dW} &:= \frac{1}{N}\sum_{n=1}^{N} \frac{d}{dW}\left(\varphi_\alpha\left(\frac{1}{\hat{\boldsymbol{c}}_{\boldsymbol{Z}}(\hat{F}_{Z_1}(Z_1(n)), \ldots, \hat{F}_{Z_p}(Z_p(n)))}\right)\right), \\
&:= \frac{-1}{N}\sum_{n=1}^{N} \varphi'_\alpha\left(\frac{1}{\hat{\boldsymbol{c}}_{\boldsymbol{Z}}(\hat{F}_{Z_1}(Z_1(n)), \ldots, \hat{F}_{Z_p}(Z_p(n)))}\right) \\
&\times \frac{\frac{d}{dW}(\hat{\boldsymbol{c}}_{\boldsymbol{Z}}(\hat{F}_{Z_1}(Z_1(n)), \ldots, \hat{F}_{Z_p}(Z_p(n))))}{\hat{\boldsymbol{c}}_{\boldsymbol{Z}}(\hat{F}_{Z_1}(Z_1(n)), \ldots, \hat{F}_{Z_p}(Z_p(n)))} \times \frac{1}{\hat{\boldsymbol{c}}_{\boldsymbol{Z}}(\hat{F}_{Z_1}(Z_1(n)), \ldots, \hat{F}_{Z_p}(Z_p(n)))},
\end{aligned} \tag{7.8}$$

where $d/dW = \partial/\partial W_{ij}; i, j = 1 \cdots p$

$$\begin{aligned}
\frac{\partial\hat{\boldsymbol{c}}_{\boldsymbol{Z}}(\hat{F}_{Z_1}(Z_1(n)), \ldots, \hat{F}_{Z_p}(Z_p(n)))}{\partial W_{ij}} &:= \frac{1}{NH_1 \cdots H_p}\sum_{m=1}^{N} \prod_{j=1, j\neq i}^{p} k\left(\frac{\widehat{F}_{Z_j}(Z_j(m)) - \widehat{F}_{Z_j}(Z_j(n))}{H_j}\right) \\
&\times k'\left(\frac{\widehat{F}_{Z_j}(Z_j(m)) - \widehat{F}_{Z_j}(Z_j(n))}{H_j}\right)\frac{1}{H_i}\frac{\partial(\widehat{F}_{Z_j}(Z_j(m)) - \widehat{F}_{Z_j}(Z_j(n)))}{\partial W_{ij}}
\end{aligned} \tag{7.9}$$

To summarize, we have the following algorithm

```
• Data: $\boldsymbol{x}$ the observed source signals
• Result: $\boldsymbol{z}$ the estimated source signals
• Initialization $\boldsymbol{W}^0 = \boldsymbol{I}_p$, $z^0 = \boldsymbol{W}^0\boldsymbol{x}$, $\mu > 0$, $\varepsilon > 0$.
• Do Update $\boldsymbol{W}$ and $\boldsymbol{z}$
        $\boldsymbol{W}^{q+1} = \boldsymbol{W}^q - \mu\frac{d\widehat{D}_\alpha(\boldsymbol{c}_\Pi, \boldsymbol{c}_{\boldsymbol{Z}})}{dW}$        (7.8).
        $\boldsymbol{z}^{q+1} = \boldsymbol{W}^{q+1}\ \boldsymbol{y}$.
•  Repeat until $|W^{q+1} - W^q| < \varepsilon$
$\boldsymbol{z} = \boldsymbol{z}^{q+1}$
```

Algorithm 4: Separation of independent source components using α-divergence

7.2.2 Dependent source components

In this section, we will analyse the case where the component sources $\boldsymbol{S}$ may be dependent, in this instance we model the dependency by a certain structure which is represented by an unknown semiparametric copula $\{\mathbb{C}_{\boldsymbol{S}}(\cdot, \boldsymbol{\theta}); \boldsymbol{\theta} \in \Theta \subset \mathbb{R}^d\}$ where its density is denoted by $\boldsymbol{c}_{\boldsymbol{s}_{\boldsymbol{\theta}}}$. It can be seen that the copula density $\boldsymbol{c}_{\boldsymbol{s}_{\boldsymbol{\theta}}}$ differ from the density copula of independence $\boldsymbol{c}_\Pi$. So as a criterion function we take $\boldsymbol{W} \to D_\alpha\left(\boldsymbol{c}_{\boldsymbol{s}_{\boldsymbol{\theta}}}, \boldsymbol{c}_{\boldsymbol{Z}}\right)$ instead of $\boldsymbol{W} \to D_{\varphi_\alpha}\left(\boldsymbol{c}_\Pi, \boldsymbol{c}_{\boldsymbol{Z}}\right)$, this criterion function is always positive and is only equals to its minimum value zero if and only if $\boldsymbol{W} = \boldsymbol{A}^{-1}$ with indeterminacies up to scale and permutation:

$$\boldsymbol{A}^{-1} = \arg\inf_{\boldsymbol{W}} D_\alpha\left(\boldsymbol{c}_{\boldsymbol{s}_{\boldsymbol{\theta}}}, \boldsymbol{c}_{\boldsymbol{Z}}\right). \tag{7.10}$$

Hence the α-divergence in the case of dependency takes the form of

$$\begin{aligned} D_\alpha\left(\boldsymbol{c}_{\boldsymbol{s}_{\boldsymbol{\theta}}}, \boldsymbol{c}_{\boldsymbol{Z}}\right) &:= \int_{[0,1]^p} \varphi_\alpha\left(\frac{\boldsymbol{c}_{\boldsymbol{s}_{\boldsymbol{\theta}}}(\boldsymbol{v})}{\boldsymbol{c}_{\boldsymbol{Z}}(\boldsymbol{v})}\right)\boldsymbol{c}_{\boldsymbol{Z}}(\boldsymbol{v})d\boldsymbol{v}, \\ &:= \mathbb{E}\left[\varphi_\alpha\left(\frac{\boldsymbol{c}_{\boldsymbol{s}_{\boldsymbol{\theta}}}(F_{Z_1}(Z_1), \ldots, F_{Z_p}(Z_p))}{\boldsymbol{c}_{\boldsymbol{Z}}(F_{Z_1}(Z_1), \ldots, F_{Z_p}(Z_p))}\right)\right]. \end{aligned} \tag{7.11}$$

To estimate this new criterion, three approaches are used depending on our knowledge about the dependency structure of the source components. In the following we will describe these three approaches:

- **The model and the parameter are known**

When both the model and the parameter are known, we propose to adapt the above criterion and estimate $\boldsymbol{W}$ the de-mixing matrix as follows

$$\boldsymbol{W} := \arg\inf_{\boldsymbol{W}} D_\alpha\left(\boldsymbol{c}_{\boldsymbol{s}_{\boldsymbol{\theta}}}, \boldsymbol{c}_{\boldsymbol{Z}}\right). \tag{7.12}$$

- **The model is known and the parameter is unknown**

In the case where the parameter θ of the dependency model is unknown the separation matrix $\boldsymbol{W}$ will be computed by

$$\boldsymbol{W} = \arg\inf_{\boldsymbol{W}} \inf_{\boldsymbol{\theta}\in\Theta} D_{\varphi_\alpha}\left(\boldsymbol{c}_{\boldsymbol{s}_{\boldsymbol{\theta}}}, \boldsymbol{c}_{\boldsymbol{Z}}\right). \tag{7.13}$$

- **The model and the parameter are unknown**

The last case where the model in unknown, we consider $L_1 := \{\mathbf{c}_{\mathbf{s}_{\theta_\mathbf{1}}}(\cdot);\, \theta_\mathbf{1} \in \mathbf{\Theta_1} \subset \mathbb{R}^\mathbf{d}\}, \cdots,\ L_T := \{\mathbf{c}_{\mathbf{s}_{\theta_\mathbf{T}}}(\cdot);\, \theta_\mathbf{T} \in \mathbf{\Theta_T} \subset \mathbb{R}^\mathbf{d}\}$ to be T models of copula densities of the source components. Then we apply the method described in the second case for each model, and finally choose the model that minimizes the criterion over all considered models, that means

$$\boldsymbol{W} = \arg\inf_{W} \inf_{\boldsymbol{\theta}\in\Theta^{k^*}} D_\alpha\left(\boldsymbol{c}_{\boldsymbol{s}_{\boldsymbol{\theta}^{\boldsymbol{k}^*}}}, \boldsymbol{c}_{\boldsymbol{Z}}\right). \tag{7.14}$$

where

$$k^* = \arg\min_{k=1\cdots T} \inf_{W} \inf_{\boldsymbol{\theta_k}\in\Theta_k} D_\alpha\left(\boldsymbol{c}_{\boldsymbol{s}_{\boldsymbol{\theta}^{\boldsymbol{k}}}}, \boldsymbol{c}_{\boldsymbol{Z}}\right). \tag{7.15}$$

We propose to estimate the separation criterion defined in 7.11 by

$$\widehat{D}_\alpha\left(\boldsymbol{c}_{\boldsymbol{s}_{\boldsymbol{\theta}}}, \boldsymbol{c}_{\boldsymbol{Z}}\right) := \frac{1}{N}\sum_{n=1}^{N} \varphi_\alpha\left(\frac{\hat{\boldsymbol{c}}_{\boldsymbol{s}_{\boldsymbol{\theta}}}(\hat{F}_{Z_1}(Z_1(n)), \ldots, \hat{F}_{Z_p}(Z_p(n)))}{\hat{\boldsymbol{c}}_{\boldsymbol{Z}}(\hat{F}_{Z_1}(Z_1(n)), \ldots, \hat{F}_{Z_p}(Z_p(n)))}\right). \tag{7.16}$$

The definitions of the marginal distribution functions and the estimates of the copula density are exactly similar to the previous section and the solution $\boldsymbol{W}$ is computed by a descent gradient algorithm.

The gradient in $\boldsymbol{W}$ of $\widehat{D}\alpha\left(\boldsymbol{c}_{\Pi},\boldsymbol{c}_{\boldsymbol{Z}}\right)$ is calculated as follows:

$$\begin{aligned}\frac{d\widehat{D}_{\alpha}\left(\boldsymbol{c}_{\boldsymbol{s}_{\boldsymbol{\theta}}},\boldsymbol{c}_{\boldsymbol{Z}}\right)}{dW} &:= \frac{1}{N}\sum_{n=1}^{N}\frac{d}{dW}\left(\varphi_{\alpha}\left(\frac{\hat{\boldsymbol{c}}_{\boldsymbol{s}_{\boldsymbol{\theta}}}(\hat{F}_{Z_1}(Z_1(n)),\ldots,\hat{F}_{Z_p}(Z_p(n)))}{\hat{\boldsymbol{c}}_{\boldsymbol{Z}}(\hat{F}_{Z_1}(Z_1(n)),\ldots,\hat{F}_{Z_p}(Z_p(n)))}\right)\right),\\ &:= \frac{1}{N}\sum_{n=1}^{N}\varphi_{\alpha}'\left(\frac{\hat{\boldsymbol{c}}_{\boldsymbol{s}_{\boldsymbol{\theta}}}(\hat{F}_{Z_1}(Z_1(n)),\ldots,\hat{F}_{Z_p}(Z_p(n)))}{\hat{\boldsymbol{c}}_{\boldsymbol{Z}}(\hat{F}_{Z_1}(Z_1(n)),\ldots,\hat{F}_{Z_p}(Z_p(n)))}\right)\\ &\times\left[\frac{\frac{d[\hat{\boldsymbol{c}}_{\boldsymbol{s}_{\boldsymbol{\theta}}}(\hat{F}_{Z_1}(Z_1(n)),\ldots,\hat{F}_{Z_p}(Z_p(n)))]}{dW}\hat{\boldsymbol{c}}_{\boldsymbol{Z}}(\hat{F}_{Z_1}(Z_1(n)),\ldots,\hat{F}_{Z_p}(Z_p(n)))}{\hat{\boldsymbol{c}}_{\boldsymbol{Z}}(\hat{F}_{Z_1}(Z_1(n)),\ldots,\hat{F}_{Z_p}(Z_p(n)))^2}\right.\\ &\left.-\frac{\frac{d[\hat{\boldsymbol{c}}_{\boldsymbol{Z}}(\hat{F}_{Z_1}(Z_1(n)),\ldots,\hat{F}_{Z_p}(Z_p(n)))]}{dW}\hat{\boldsymbol{c}}_{\boldsymbol{s}_{\boldsymbol{\theta}}}(\hat{F}_{Z_1}(Z_1(n)),\ldots,\hat{F}_{Z_p}(Z_p(n)))}{\hat{\boldsymbol{c}}_{\boldsymbol{Z}}(\hat{F}_{Z_1}(Z_1(n)),\ldots,\hat{F}_{Z_p}(Z_p(n)))^2}\right]\end{aligned}\tag{7.17}$$

where $d/dW = \partial/\partial W_{ij}; i,j = 1\cdots p$.
The derivation $\frac{\partial\hat{\boldsymbol{c}}\boldsymbol{Z}(\hat{F}Z_1(Z_1(n)),\ldots,\hat{F}Z_p(Z_p(n)))}{\partial Wij}$ is calculated exactly as (7.9) and $\frac{d[\hat{\boldsymbol{c}}\boldsymbol{s}_{\boldsymbol{\theta}}(\hat{F}Z_1(Z_1(n)),\ldots,\hat{F}Z_p(Z_p(n)))]}{dW}$ is calculated by a direct derivation from the well known copulas. Finally $\boldsymbol{W}$ is estimated by $\widehat{\boldsymbol{W}} = \arg\inf_{\boldsymbol{W}}\widehat{D}_{\alpha}\left(\boldsymbol{c}_{\boldsymbol{s}_{\boldsymbol{\theta}}},\boldsymbol{c}_{\boldsymbol{Z}}\right)$, eventually leading to estimation of the source signals $\widehat{\boldsymbol{Z}}(n) = \widehat{W}\boldsymbol{Y}(n), \forall n = 1\cdots N$. To summarize the method we have the following algorithm

```
• Data: $\boldsymbol{x}$ the observed source signals
• Result: $\boldsymbol{z}$ the estimated source signals
• Initialization $\boldsymbol{W}^0 = \boldsymbol{I}_p$, $z^0 = \boldsymbol{W}^0\boldsymbol{x}$, $\mu > 0$ , $\varepsilon > 0$.
• Do Update $\boldsymbol{W}$ and $\boldsymbol{z}$
        Compute $\widehat{D}_{\varphi_\alpha}\left(\boldsymbol{c}_{\boldsymbol{s}_{\boldsymbol{\theta}}},\boldsymbol{c}_{\boldsymbol{Z}}\right)$      (7.17).
        $\boldsymbol{W}^{q+1} = \boldsymbol{W}^q - \mu\frac{d\widehat{D}_{\alpha}\left(\boldsymbol{c}_{\boldsymbol{s}_{\boldsymbol{\theta}}},\boldsymbol{c}_{\boldsymbol{Z}}\right)\cdot}{dW}$.
        $\boldsymbol{z}^{q+1} = \boldsymbol{W}^{q+1}\ \boldsymbol{y}$.
• Repeat until $||W^{q+1} - W^q|| < \varepsilon$
$\boldsymbol{z} = \boldsymbol{z}^{q+1}$
```

Algorithm 5: Separation of dependent source components using the α-divergence

7.3 Simulation results

In the following, we present the results of various simulations of our proposed approach. The experiments are done for 3 observation mixtures of 3 sources to be estimated. These sources are mixed through a mixing matrix $M := [1\,0.5\,0.7;\ 0.5\,1\,0.7;\ 0.5\,0.7\,1]$, where for each source we have N = 2000 samples, and all simulations are repeated 80 times with $\mu = 0.1$ as the gradient descent parameter. We illustrate the performance of BSS-copula approach, first for independent sources then the dependent sources and finally the noise-contaminated sources using both the signal-noise-ratio (SNR) which is defined as follows:

$$SNR_j = 10\log_{10}\left(\frac{\sum_{k=1}^{N}\widehat{z_j}(k)}{\sum_{k=1}^{N}(\widehat{z_j}(k) - s_j(k)^2)}\right),\ j = 1,2,3 \tag{7.18}$$

and the Performance Index [28] which is defined as such

$$PI = \frac{1}{N(n-1)} \sum_{i}^{N} \{ (\sum_{k=1}^{N} \frac{|u_{ik}|}{max_j u_{ij}} - 1) + (\sum_{k=1}^{N} \frac{|u_{ki}|}{max_j u_{ji}} - 1) \} \tag{7.19}$$

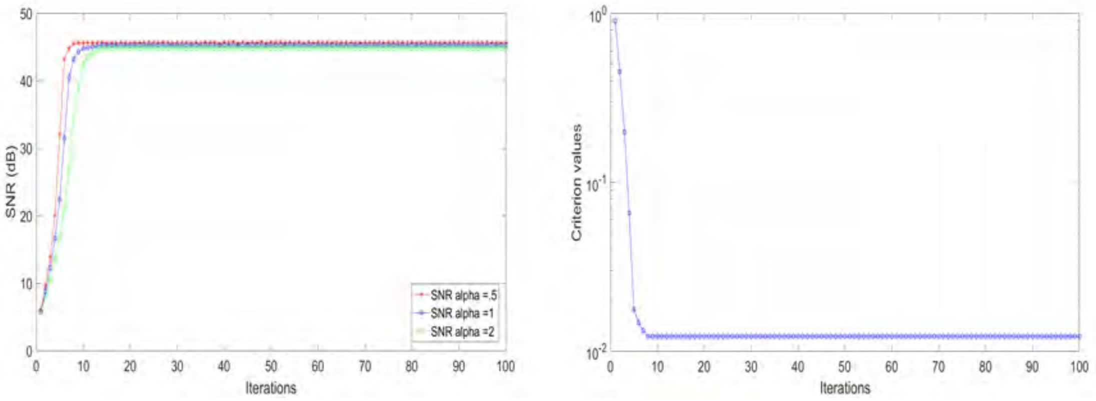
where u_{ij} is the (i,j)-element of the global system matrix U=AW, $max_j u_{ij}$ represents the maximum value among the elements in the i^{th} row vector of U and $max_j u_{ji}$ represents the maximum value among the elements of the i^{th} column vector of U. When the perfect separation is achieved, the performance index is zero, index when the index is close to 0 we can say that the quality of the separation is good. We also compare our approach with various existing methods for the BSS problem.

7.3.1 Independent source components

We consider in this experiment three mixed signals from a uniform i.i.d with independent components.
First we test our approach for independent components using three values of α:

- $\alpha = 0$ which is equivalent to the modified Kullback-Leibler divergence.
- $\alpha = 0.5$ which is equivalent to the Hellinger divergence.
- $\alpha = 2$ which is equivalent to the χ^2-divergence.

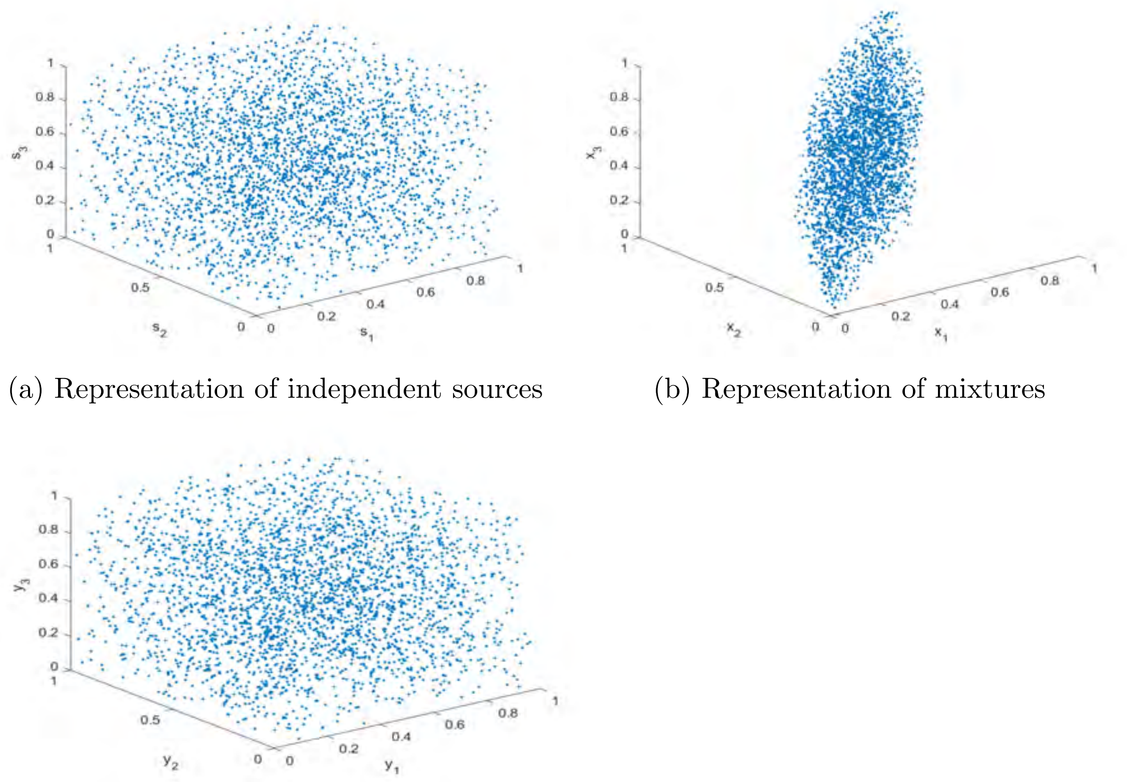
Figure. 7.1a presents the average SNR output in function of the number of iterations, using our approach for $\alpha = 0, 0.5, 2$. It is noticeable that the Hellinger divergence, e.g. $\alpha = 0.5$, is more accurate than the other α-divergences. On the other hand, Figure. 7.1b, represents the criterion value vs iterations. We can see that the separation is achieved when our criterion converges to its minimum value 0.
Figure. 7.2a represents the shape of the original source signals, figure. 7.2b shows their shape after being mixed by the mixing matrix A and finally figure. 7.2c, showcases the shape of the estimated original data after the separation.



(a) Average output SNRs versus iteration number

(b) The criterion value vs iterations for $\alpha = 0.5$

Figure 7.1: Uniform independent sources for the three values of α



(a) Representation of independent sources

(b) Representation of mixtures

(c) Representation of estimated sources

Figure 7.2: Uniform independent sources for $\alpha = 0.5$

We compare our method with the following methods: [128](MI), [23] (JADE), [83] (FastICA), [117] (RADICAL) and [17] (InfoMax). Table 7.1 represents the SNR output of the sources for the different approaches, we can see that our approach when $\alpha = \frac{1}{2}$ which is the Hillenger diverence is the one with the highest SNR. Also, table 7.2 exhibit the superiority of the Hellinger divergence compared to the other methods because its Performance Index is the one closer to 0.

Sources	**S1**	**S2**	**S3**
Copula $\alpha = 0$	44.9898	44.4859	44.4188
Copula $\alpha = 0.5$	**46.6395**	**46.7664**	**46.3895**
Copula $\alpha = 2$	44.5414	44.0996	44.0484
MI	43.9801	43.7431	43.4067
FastICA	43.0801	43.7431	43.4067
JADE	43.6369	43.4810	43.0588
RADICAL	43.4713	43.8964	43.5151
InfoMax	43.8154	43.5161	43.2699

Table 7.1: Output SNR's for independent source components

Approach	**PI**	**Approach**	**PI**
Copula $\alpha = 0$	0.0171	**FastICA**	0.0259
Copula $\alpha = 0.5$	**0.0122**	**JADE**	0.0230
Copula $\alpha = 2$	0.0199	**RADICAL**	0.0241
MI	0.0202	**InfoMax**	0.0251

Table 7.2: Output PI's for independent source components

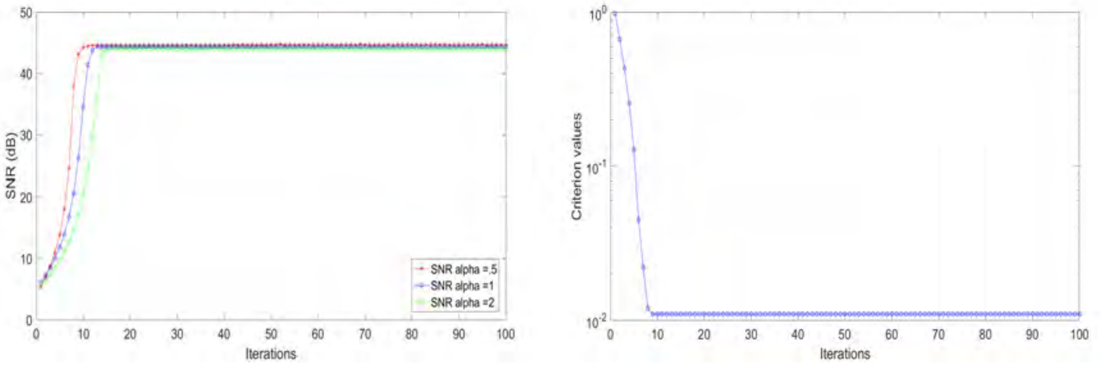
7.3.2 Dependent source components

We present within this subsection, the potential of our proposed approach (algorithm 5 for dependent sources) and its capability to separate mixtures of three dependent signals successfully. We treated instant mixtures of four types of sample sources:

- i.i.d (with uniform marginals) dependent source components generated from AMH copula with $\theta = 0.4$.
- i.i.d (Binary Phase-Shift Keying (BPSK)) dependent source components generated from Clayton copula, with $\theta = 1.3$
- i.i.d (with uniform marginals) dependent source components generated from Frank copula, $\theta = 2.5$.
- i.i.d (with uniform marginals) dependent source components generated from Gaussian copula, $\theta = 0.2$.

We test our approach for the four different dependent samples introduced above for three values of α as done before for the independent case.
Figures. 7.3a, 7.5a, 7.7a, 7.9a, show the SNRs for dependent sources from AMH, Clayton, Frank and Gaussian copulas, respectively in function of the number of iterations, using our approach for the three values of α: $0, 0.5, 2$. From the simulation results, it is noticeable that in this case also the Hellinger has the highest quality of the separation. Moreover, Figures. 7.3b, 7.5b, 7.7b, 7.9b represent the criterion value vs iterations for the said copulas. We can see that the separation is achieved when our criterion converges to its minimum value 0.

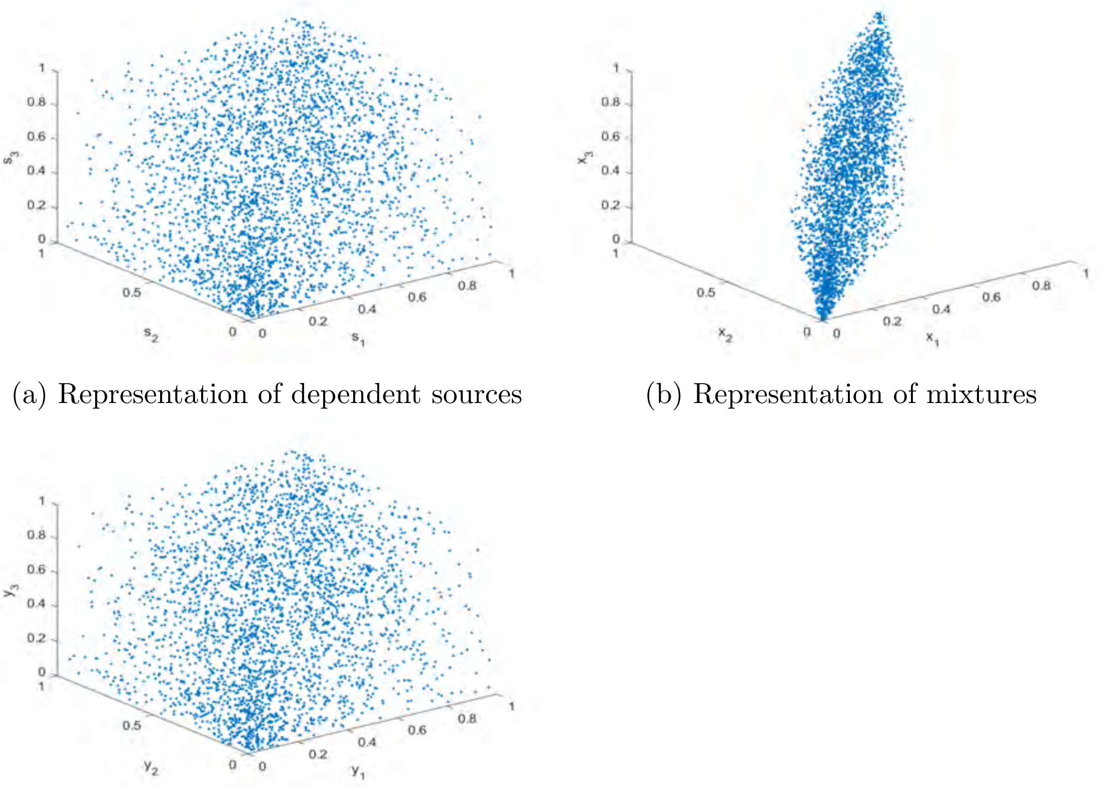
Figures. 7.4a, 7.6a, 7.8a,7.10a represents the shape of the original source signals from AMH, Clayton, Frank and Gaussian copulas respectively. Figures. 7.4b,7.6b, 7.8b, 7.10b show their shape after being mixed by the mixing matrix A and finally figures. 7.4c,7.6c, 7.8c, 7.10c, showcase the shape of the estimated original sources after the separation.



(a) Average output SNRs versus iteration number

(b) The criterion value vs iterations for $\alpha = 0.5$

Figure 7.3: Uniform dependent sources from AMH copula ($\theta = 0.4$) for the three values of α

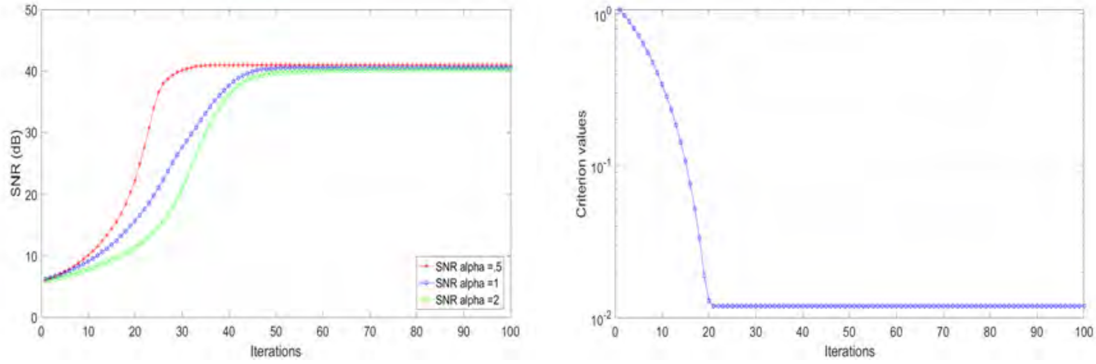


(a) Representation of dependent sources

(b) Representation of mixtures

(c) Representation of estimated sources

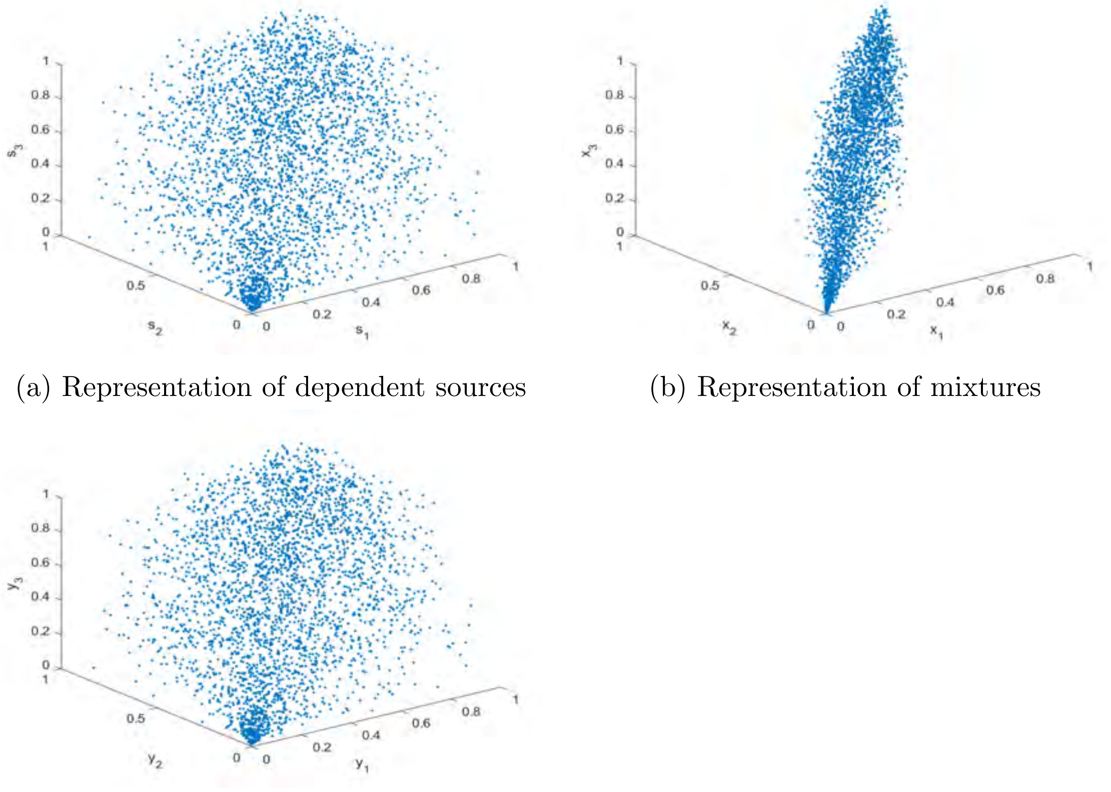
Figure 7.4: Uniform dependent sources from AMH copula ($\theta = 0.4$) for $\alpha = 0.5$



(a) Average output SNRs versus iteration number

(b) The criterion value vs iterations for $\alpha = 0.5$

Figure 7.5: BPSK dependent sources from Clayton copula ($\theta = 1.3$)for the three values of α

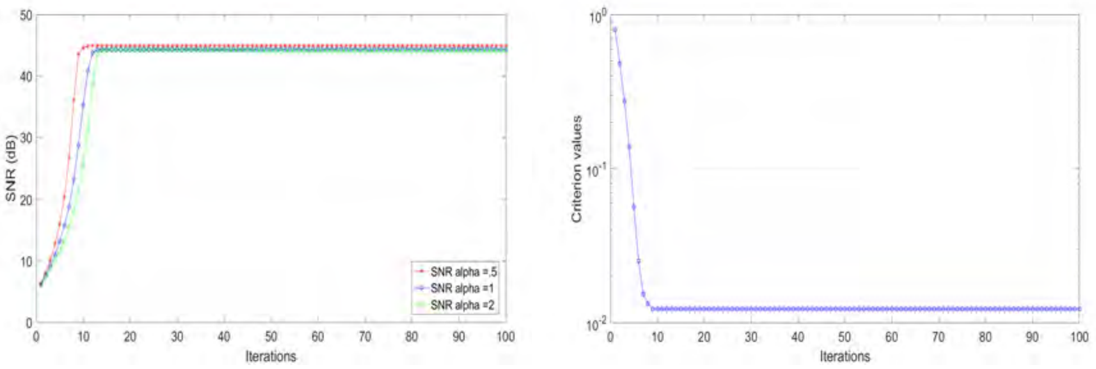


(a) Representation of dependent sources

(b) Representation of mixtures

(c) Representation of estimated sources

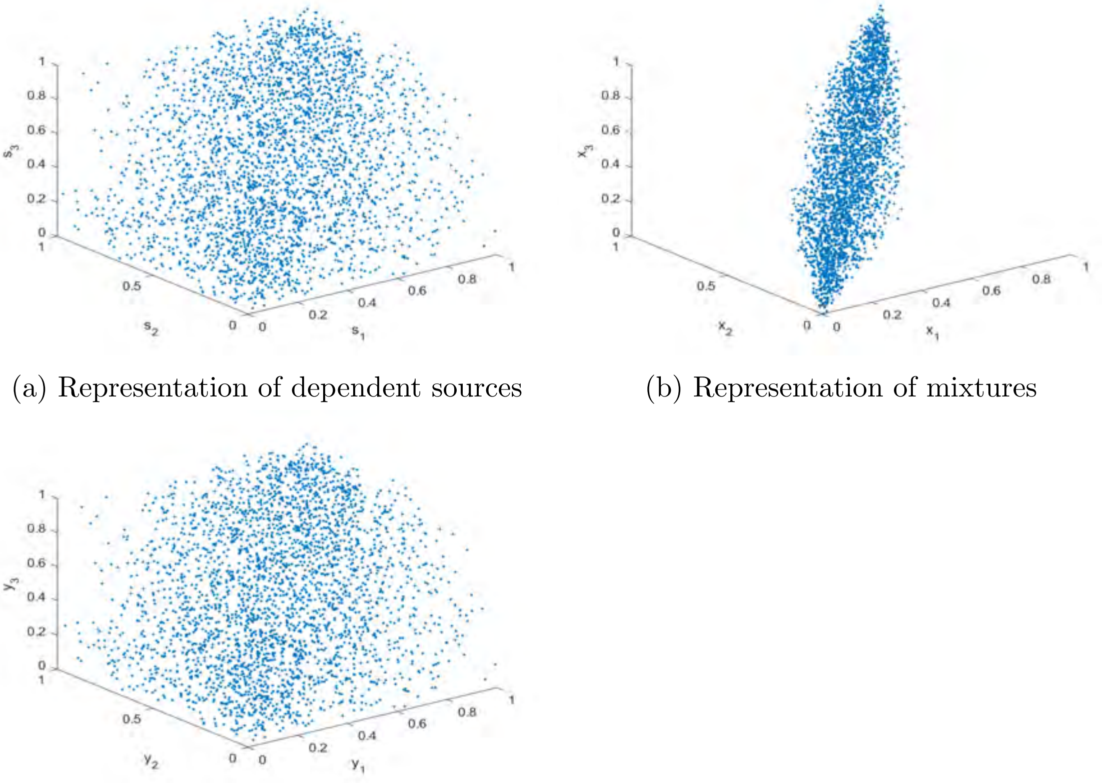
Figure 7.6: BPSK dependent sources from Clayton copula($\theta = 1.3$) for $\alpha = 0.5$



(a) Average output SNRs versus iteration number

(b) The criterion value vs iterations for $\alpha = 0.5$

Figure 7.7: Uniform dependent sources from Frank copula ($\theta = 2.5$) for the three values of α

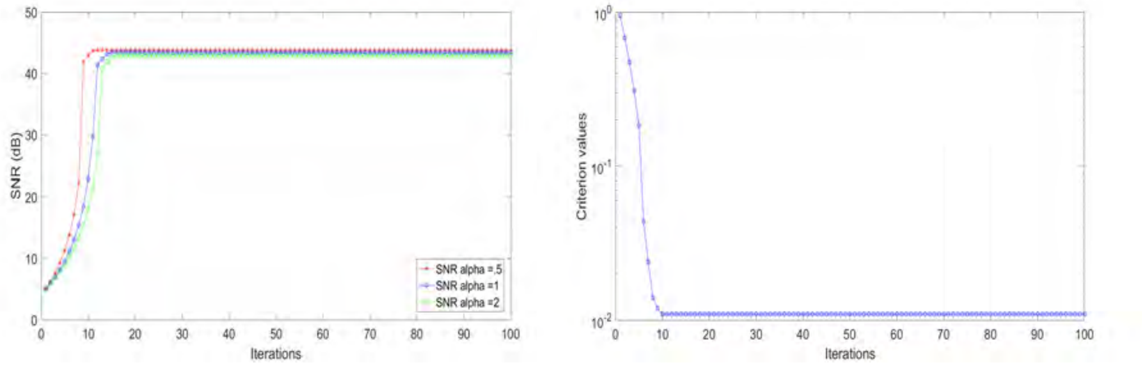


(a) Representation of dependent sources

(b) Representation of mixtures

(c) Representation of estimated sources

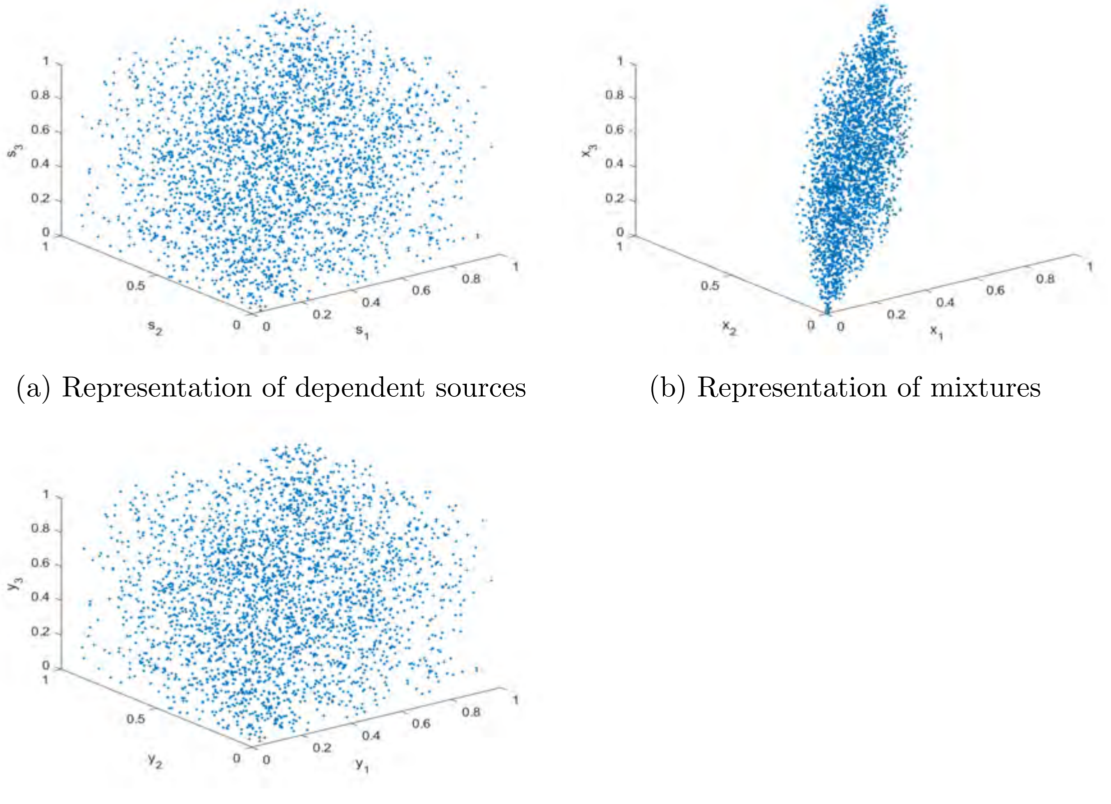
Figure 7.8: Uniform dependent sources from Frank copula ($\theta = 2.5$) for $\alpha = 0.5$



(a) Average output SNRs versus iteration number

(b) The criterion value vs iterations for $\alpha = 0.5$

Figure 7.9: Uniform dependent sources from Gaussian copula ($\theta = 0.2$) for the three values of α



(a) Representation of dependent sources

(b) Representation of mixtures

(c) Representation of estimated sources

Figure 7.10: Uniform dependent sources from Gaussian copula ($\theta = 0.2$) for $\alpha = 0.5$

Tables 7.3 and 7.4 exhibit the superiority of the proposed approach especially for $\alpha = \frac{1}{2}$ compared to the existing approaches mentioned in the previous section for the two evaluation metrics: SNR and the Performance Index.

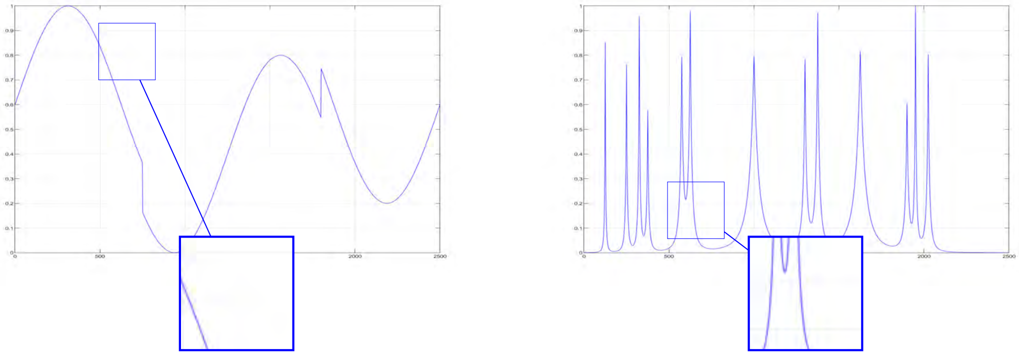
Type	**AMH**			**Clayton**		
Sources	**S1**	**S2**	**S3**	**S1**	**S2**	**S3**
Copula $\alpha = 0$	43.7564	43.1242	43.3107	39.3411	39.5134	38.9531
Copula $\alpha = 0.5$	**45.3499**	**45.5485**	**45.2925**	**41.7242**	**41.2638**	**41.8273**
Copula $\alpha = 2$	43.6874	43.0231	43.1316	39.1823	39.2651	38.5204
MI	22.3045	22.1221	22.1873	16.1335	16.6321	16.3413
FastICA	40.9576	16.8551	15.7032	20.2045	11.4156	11.3645
JADE	21.0242	21.1523	21.5396	15.3368	15.3520	15.7654
RADICAL	20.5861	21.0112	20.4135	15.8944	15.9326	15.9075
InfoMax	15.9627	16.0018	16.0121	12.0985	12.1062	12.1419
Type	**Frank**			**Gaussian**		
Sources	**S1**	**S2**	**S3**	**S1**	**S2**	**S3**
Copula $\alpha = 0$	44.6787	44.4692	44.8902	43.7743	43.1553	43.3212
Copula $\alpha = 0.5$	**46.3790**	**46.1961**	**46.0499**	**45.0324**	**45.2317**	**45.1545**
Copula $\alpha = 2$	43.8468	44.0643	44.5799	42.5768	42.7816	43.1019
MI	22.1426	22.2120	22.3240	22.56292	22.8021	21.9356
FastICA	41.2526	11.8120	12.6400	41.0267	11.8322	11.8233
JADE	19.6022	19.6687	19.7018	18.9111	18.3361	18.1552
RADICAL	17.9526	17.7357	17.8292	17.1106	17.1933	17.1542
InfoMax	11.9645	11.9963	11.0047	10.9003	10.7977	10.8522

Table 7.3: Output SNR's for dependent source components

	AMH	**Clayton**	**Frank**	**Gaussian**
Copula $\alpha = 0$	0.0167	0.0965	0.0128	0.0086
Copula $\alpha = 0.5$	**0.0159**	**0.0912**	**0.0117**	**0.0070**
Copula $\alpha = 2$	0.0207	0.0986	0.0143	0.0093
MI	0.2887	0.3147	0.3002	0.2213
FastICA	0.3071	0.3374	0.3155	0.2156
JADE	0.3467	0.4005	0.3504	0.2331
RADICAL	0.3467	0.4743	0.3949	0.2340
InfoMax	0.3992	0.4322	0.3723	0.2342

Table 7.4: Output PI's for dependent source components

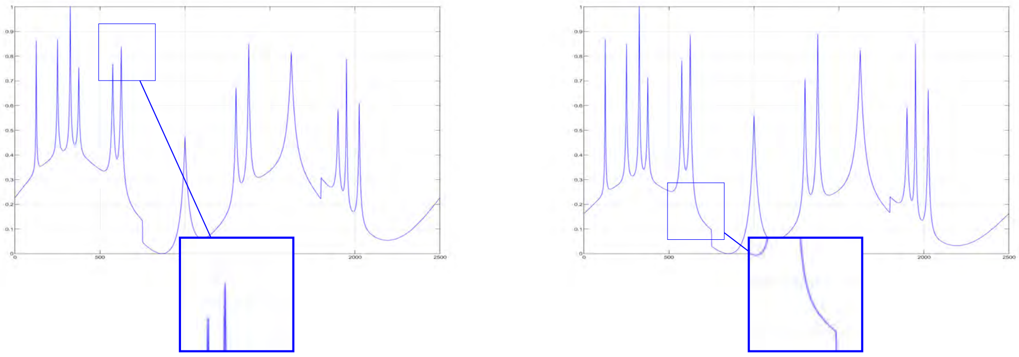
To test our approach for the deterministic case, we took two types of sources. The first one is the "Waves" signals, which combine two waves and it is represented on the far right of the figure. 7.11. The second one is a "Bumps" source signal represented by the second image in figure. 7.11. The number of samples of the sources is fixed on N=2500. These two sources are lightly dependent (AMH copula with $\theta = 0.4$) as their Kendall's Tau is equal to 0.102 and they were mixed linearly by the same mixing matrix A as in the previous simulations. The mixtures are shown in the pictures in figure. 7.12. To separate these two sources, we applied our approach in section 7.2.2 without assuming the statistical dependency. The two images in Figure. 7.13 are the estimated sources using the Hellinger divergence and figures in 7.14 are the estimated sources using the MI. Figure. 7.17 represents different SNR outputs for the dependent sources from the AMH copula using the Hellinger divergence and the MI, illustrating the superiority of the separation using the Hellinger divergence.



(a) ”Waves signal”

(b) ”Bumps signal”

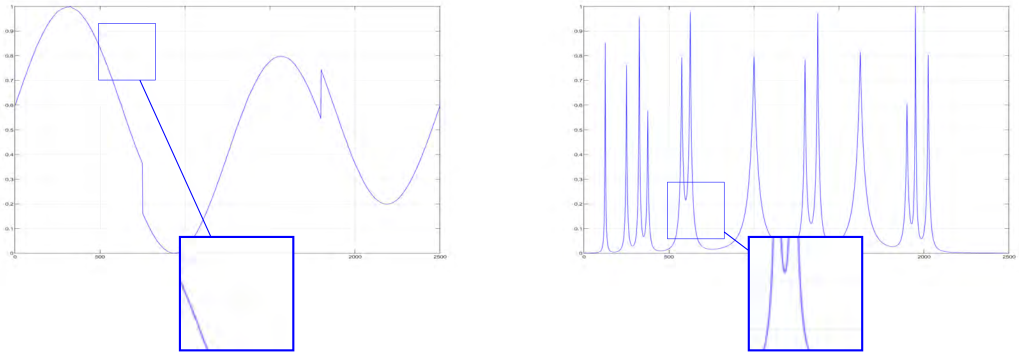
Figure 7.11: Sources ”Waves” and ”Bumps” and their mixtures



(a) Mixture 1

(b) Mixture 2

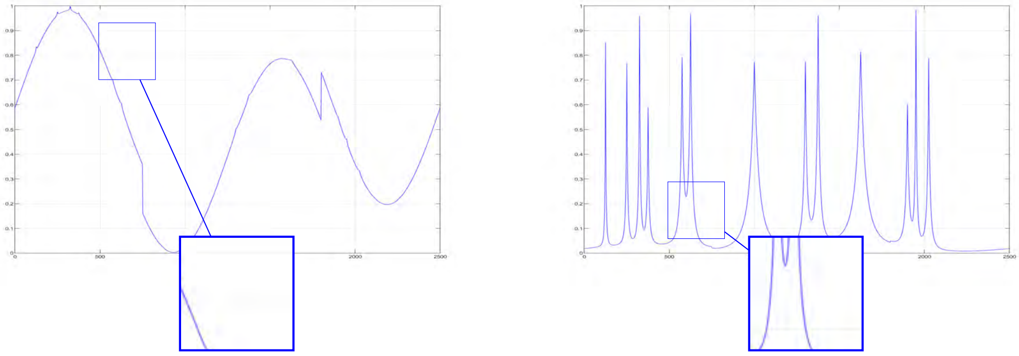
Figure 7.12: Sources ”Waves” and ”Bumps” and their mixtures



(a) ”Waves” estimated via proposed approach($\alpha = .5$)

(b) ”Bumps” estimated via proposed approach($\alpha = .5$)

Figure 7.13: Estimated dependent sources from AMH copula ($\theta = 0.4$) using the Hellinger divergence and the MI method



(a) "Waves" estimated via MI method (b) "Bumps" estimated via MI method

Figure 7.14: Estimated dependent sources from AMH copula ($\theta = 0.4$) using the Hellinger divergence and the MI method

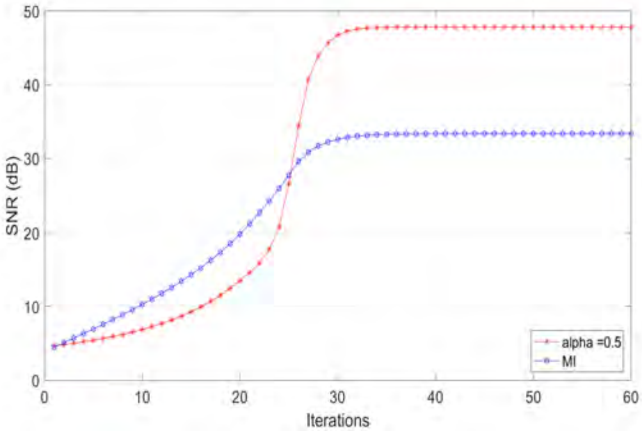


Figure 7.15: The SNR output for the Hellinger divergence and the MI

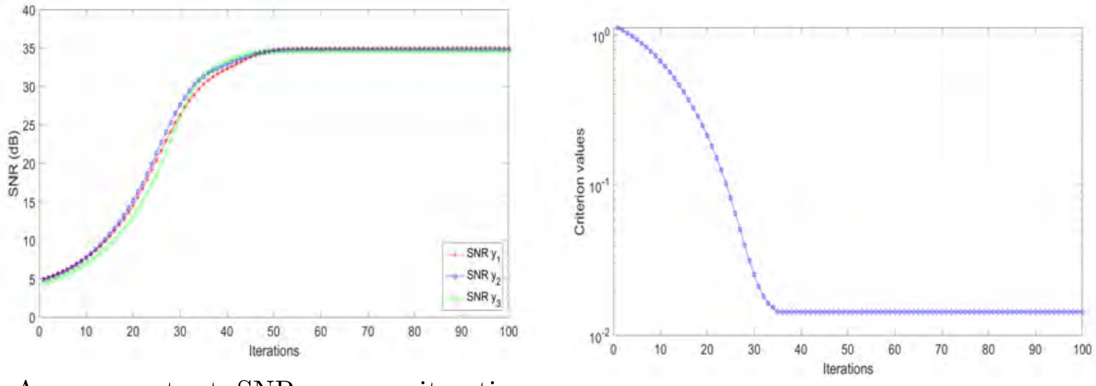
7.3.3 Noisy source components

In this subsection, we test the accuracy of our approach when a bit of noise is present during the mixing process. We work with the same source signals used above and the same conditions and we add white Gaussian noise to the observed signals. We take $SNR = -25dB$.

Figure. 7.16a illustrates the SNR of the dependent sources from Clayton copula for the three values of α, it can be seen that the proposed approach is able to separate even noisy dependent sources with good performance and that the hellinger divergence is again the faster to convergence. Moreover Figure. 7.16b shows that when the separation is achieved our criterion converges to its minimum 0.

Figure. 7.17a represents the shape of the original source signals, figure. 7.17b shows their shape after being mixed by the mixing matrix A and finally figure. 7.17c, showcases the shape of the estimated original data after the separation for the noisy dependent sources from the Clayton copula.

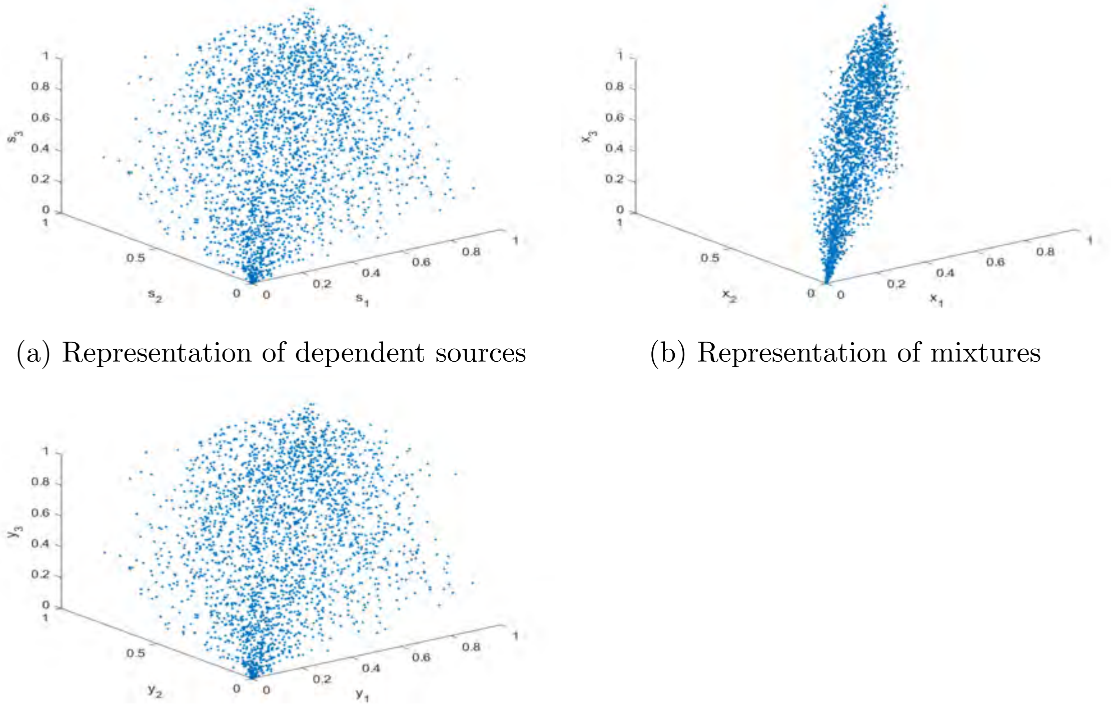
Table 7.5 presents the output SNR values of the estimated sources using the α-divergence and the other methods, we can see that the approaches are equivalent, with superiority of our method, in case of noise-contaminated independent source components. On the other hand, our approach is apt to separate even noisy mixtures of dependent source components with higher accuracy. Table 7.7 exhibits the output Performance Index of our approach, showcasing its superiority especially for $\alpha = \frac{1}{2}$.



(a) Average output SNRs versus iteration number

(b) The criterion value vs iterations

Figure 7.16: Uniform dependent noisy sources from Clayton-copula ($\theta = 1.4$) for $\alpha = 0.5$



(a) Representation of dependent sources

(b) Representation of mixtures

(c) Representation of estimated sources

Figure 7.17: Uniform dependent sources from Clayton copula ($\theta = 1.4$) for $\alpha = 0.5$

Copulas	**Independence**			**Clayton**		
Sources	**S1**	**S2**	**S3**	**S1**	**S2**	**S3**
Copula $\alpha = 0$	31.3267	31.4681	30.9519	29.1470	29.2910	29.5691
Copula $\alpha = 0.5$	**36.1882**	**36.2384**	**36.0518**	**34.9658**	**34.8174**	**34.6611**
Copula $\alpha = 2$	31.0214	31.0657	30.7131	28.9377	29.1850	29.1672
MI	30.4531	30.4178	30.4130	7.1605	7.0444	6.6586
FastICA	29.2118	29.6796	28.2773	25.5767	3.0302	3.7841
JADE	30.2331	30.2209	30.2495	6.9056	6.8649	7.1214
RADICAL	29.9161	29.9472	30.1102	6.5064	7.4440	6.9503
InfoMax	30.0881	30.6651	30.5402	5.3751	5.3627	5.3168

Table 7.5: Output SNR's for independent and dependent noisy source components

Copulas	**Frank**		
Sources	**S1**	**S2**	**S3**
Copula $\alpha = 0$	30.9543	30.7120	30.6045
Copula $\alpha = 0.5$	**36.2770**	**36.5339**	**36.3887**
Copula $\alpha = 2$	30.4445	30.4273	30.4370
MI	12.2857	12.3780	12.1843
FastICA	28.5473	8.4895	6.6148
JADE	13.2814	13.3712	12.8820
RADICAL	12.4281	14.7506	12.5084
InfoMax	8.3977	8.2703	8.3137

Table 7.6: Output SNR's for dependent noisy source components of Frank copula

Copulas	**Independence**	**Clayton**	**Frank**
Copula $\alpha = 0$	0.0137	0.0159	0.0171
Copula $\alpha = 0.5$	**0.0101**	**0.0115**	**0.0133**
Copula $\alpha = 2$	0.0191	0.0211	0.0202
MI	0.03541	0.4058	0.4440
FastICA	0.0211	0.4796	0.4773
JADE	0.0231	0.4219	0.4435
RADICAL	0.0461	0.4342	0.1102
InfoMax	0.0685	0.4561	0.4523

Table 7.7: Output PI's for independent and dependent noisy source components

7.4 Conclusion

A new approach for Blind Source Separation (BSS), based on minimizing the α-divergences between the copula densities of the sources have been presented in this chapter for both independent and dependent sources as a generalization of the mutual information approach. We minimize the estimate of the α-divergence for different values of α. In this study, we deducted that the Hellinger divergence which is equivalent to $\alpha = 0.5$ is the distance that gives the highest accuracy compared to other values and other approaches as it was illustrated in section "Simulation results" for 3 × 3 mixture-sources of independent, dependent and even noise contaminated sources. The efficiency and the accuracy of the algorithms was evaluated through both the signal-to-noise-ratio criterion and the performance index.

Chapter 8

Applications

8.1 Introduction

In this chapter we will be introducing variety of applications of BSS that we applied. First we used BSS to denoise digital images, our approach was used for different types of noise (gaussian, poisson and so on). We also used BSS for separating images of mixed fingerprints which could be useful in a crime investigation to separate a mixture of two fingerprints and finally we used BSS to remove the bleed-through/show-through effects which is detected in the antique documents with the aim of improving text readability and optical character recognition (OCR) efficiency.

8.2 Image denoising

Due to the influence of the environment during acquisition and transmission, images are contaminated by additive noise, leading to distortion and loss of image information. The main aim of image denoising is to remove noise from a noisy image, so as to restore the original image while preserving the image features. In fact, image denoising is a fundamental challenge in computer vision especially in image restoration and has been widely studied in the image processing community.

During the last decade multiple algorithms were proposed for denoising images. The wavelet transform was a popular image-denoising approach, it was first introduced by [114] and since, it was heavily investigated in image denoising [36, 45, 66, 76, 87, 102, 154, 156]. Since the wavelet transform has many good characteristics, such as sparseness and multi-scaling, it is still an active area of research in image denoising. However, the wavelet transform heavily relies on the selection of wavelet bases. If the selection is inappropriate, image shown in the wavelet domain cannot be well represented, which causes poor denoising effect. Therefore, this method is not adaptive. More recently, a growing number of researchers have shifted their attention to nonlocal image restoration techniques, first was [20], motivated by the key observation that many important structures in natural images, including textures and edges, are characterized by abundant self-repeating patterns. From here, many improved versions have been proposed. Some studies focused on the acceleration of the algorithm [42, 147], while others focused on how to enhance the performance of the algorithm [57, 72, 132]. We also have the sparsity-based regularization that has led to a variety of promising results for image restoration problems including image denoising [51, 56, 112, 158]. For the data adaptive transforms we have Independent component analysis (ICA) [37] and Principal component Analysis (PCA) [166]. Among them, the ICA method has been successfully implemented for image denoising [77, 167].

Most denoising methods assume that the noise is white Gaussian or have prior information about its properties. However, since noise, texture, and edge, are high-frequency components, it is difficult to distinguish them in the process of denoising and the denoised images could inevitably lose some details. In our previous work [96], we used copula models to better estimate the mixing matrix along with the sources using solely the mixtures for both independent and dependent source components. This gives us an opportunity to omit the independence assumption that was used for years and model the BSS problem for statistically dependent source components. Hence in this chapter, we first start from a denoising problem to a BSS problem by creating a second observation image from the noisy one and then applying the copula-based BSS to separate the clean image from the noise. Then we introduce a TGV regularization term into our cost function to eliminate the apparent blur in the estimated image from the noise-free BSS approach.

8.2.1 The proposed approach

Our denoising approach is based on blind source separation (BSS) using copulas (7.2). Let us first consider that the random (vector) processes $\boldsymbol{s}(t)$, $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t) := \boldsymbol{W}\,\boldsymbol{x}(t)$, $t \in [0, T]$, are stationary, so that the corresponding discretized versions, with certain time period say T_e is,

$$\boldsymbol{s}(m) := \boldsymbol{s}(mT_e),\ \boldsymbol{x}(m) := \boldsymbol{x}(mT_e) \tag{8.1}$$

and

$$\boldsymbol{y}(m) := \boldsymbol{y}(mT_e) = \boldsymbol{W}\,\boldsymbol{x}(m),\ m = 1, \ldots, N_t, \tag{8.2}$$

can be viewed as realizations of random vectors in $\mathbb{R}^2$, which will be denoted, respectively, by

$$\boldsymbol{S}, \quad \boldsymbol{X} \quad \text{and} \quad \boldsymbol{Y} := \boldsymbol{W}\boldsymbol{X}. \tag{8.3}$$

Notice that the random vector $\boldsymbol{Y}$, as well as the associated samples $\boldsymbol{y}(m)$, $m = 1, \ldots, N_t$, depends on the deterministic (unknown) matrix $\boldsymbol{W}$. Let $i_1, \ldots, i_N$ be N i.i.d. realizations (in decreasing order) generated, independently, from the uniform discrete probability distribution on the set index $\{1, \ldots, N_t\}$, with $N \leq N_t$. For simplicity of notation, we will simply denote $1, \ldots, N$ instead of $i_1, \ldots, i_N$. Therefore, each of $\boldsymbol{s}(m)$, $\boldsymbol{x}(m)$ and

$$\boldsymbol{y}(m) := \boldsymbol{W}\,\boldsymbol{x}(m),\ m = 1, \ldots, N, \tag{8.4}$$

can be viewed as N i.i.d. realizations, respectively, of the random vectors $\boldsymbol{S}$, $\boldsymbol{X}$ and $\boldsymbol{Y} := \boldsymbol{W}\boldsymbol{X}$. We will use in what follows this i.i.d. realization (8.4), for estimating the separating matrix, instead of the "original" data (8.2) which my be (temporally) dependent. In the other hand, in practice N_t may be large, and the i.i.d. data (8.4), with $N << N_t$, may be enough for separating the sources with good performance.

8.2.2 Denoising Independent noise

In this section, we will tackle down two cases of signal independent noises, the Gaussian noise and the combination of Salt & pepper and Gaussian noise.

Gaussian noise

Gaussian noise is often used to model the noise statistics in many applications, such as in medical imaging, as a simple approximation of more complicated noise models. The noise is assumed to be an additive random component n, Gaussian-distributed with zero mean and variance σ^2 determining the noise intensity. In this case the denoising problem can be simply written as

$$\boldsymbol{I} = \boldsymbol{I_0} + \boldsymbol{n} \tag{8.5}$$

where $\boldsymbol{I}$ is the noise-corrupted image, $\boldsymbol{I_0}$ the clean image that we want to estimate and $\boldsymbol{n}$ the additive Gaussian noise. Our approach uses BSS to separate $\boldsymbol{I_0}$ from $\boldsymbol{n}$. Nonetheless, to enable BSS to work, the number of observations needs to be equal to the number of the sources. To tackle this problem we propose to create a second observation $\boldsymbol{I}^{'}$. The idea is to use bilateral filter [127] to decrease the noise level in the image hence constructing a second observation with different weights for the clear image and the noise component. The following Figure 8.1 describes our denoising process

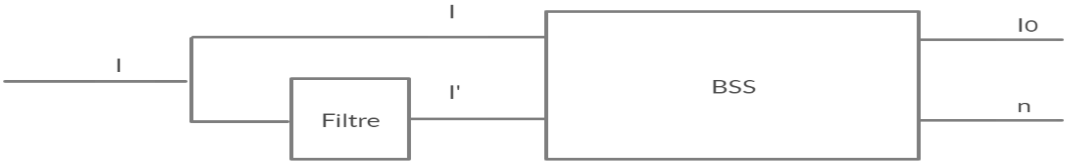


Figure 8.1: The BSS process for denoising images

where $\boldsymbol{I}$ and $\boldsymbol{I}^{'}$ are the observations, and $(\boldsymbol{I_0}, \boldsymbol{n})$ are the sources, the current BSS model is written as follows

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{I}^{'} \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \times \begin{bmatrix} \boldsymbol{I}_0 \\ \boldsymbol{n} \end{bmatrix} = \boldsymbol{M}\,\boldsymbol{S} \tag{8.6}$$

with $\boldsymbol{M}$ the mixing matrix where $m_{11} = m_{12} = 1$. The goal is to estimate $y = (y_1, y_2)$ of $(\boldsymbol{I_0}, \boldsymbol{n})$ by finding a demixing matrix W that should be as close as M^{-1}. To separate the mixture we minimize a criterion of this form:

$$\mathcal{J} : \boldsymbol{W} \to \mathcal{J}(\boldsymbol{W}) := \mathcal{J}_{sep}(\boldsymbol{W}) + \mathcal{J}_{reg}(\boldsymbol{y}) \tag{8.7}$$

where $\boldsymbol{y}(t) = \boldsymbol{W}\boldsymbol{x}(t) \in \mathbb{R}^2$, $\mathcal{J}_{sep}(.)$ is the separating criterion, while $\mathcal{J}_{reg}(.)$ is a regularization term.

As shown in the previous chapters the Mutual information of a random variable $\boldsymbol{Y}$ can be regarded as the Kullback-Leibler divergence between the copula density $c_{\boldsymbol{Y}}(\cdot)$ of $\boldsymbol{Y}$ and the copula density of independence c_0 (7.2). Moreover, $\mathbb{MI}(\boldsymbol{Y}) = \mathbb{KL}\left(c_{\boldsymbol{Y}}||c_0\right)$ is non-negative and achieves its minimum value zero iff $c_{\boldsymbol{Y}}(\boldsymbol{u}) = c_0(\boldsymbol{u})$, $\forall \boldsymbol{u} \in$

$[0,1]^2$, namely, iff the components of the vector $\boldsymbol{Y}$ are independent. The Kullback-Leibler divergence is calculated as follows:

$$\begin{aligned}\mathbb{KL}\left(c_{\boldsymbol{Y}}||c_0\right) &:= \int_{[0,1]^2} -\log\left(\frac{1}{c_{\boldsymbol{Y}}(\boldsymbol{u})}\right) c_{\boldsymbol{Y}}(\boldsymbol{u})\, d\boldsymbol{u} \qquad (8.8)\\ &= \mathbb{E}\left[\log c_{\boldsymbol{Y}}\left(F_{Y_1}(Y_1), F_{Y_2}(Y_2)\right)\right] \qquad (8.9)\end{aligned}$$

The separating term is then

$$\mathscr{J}_{\text{sep}}(\boldsymbol{W}) := \mathbb{KL}\left(c_{\boldsymbol{Y}}||c_0\right) \qquad (8.10)$$

The statistical estimate $\widehat{\mathscr{J}_{\text{sep}}}$ of the separating term $\mathscr{J}_{\text{sep}}$ is defined by

$$\widehat{\mathscr{J}_{\text{sep}}}(\boldsymbol{W}) := \mathbb{KL}\widehat{\left(c_{\boldsymbol{Y}}||c_0\right)} = \frac{1}{N}\sum_{m=1}^{N} \log\left(\widehat{c}_{\boldsymbol{Y}}(\widehat{F}_{Y_1}(y_1(m)), \widehat{F}_{Y_2}(y_2(m)))\right). \qquad (8.11)$$

where, $\forall \boldsymbol{u} \in [0,1]^2$,

$$\widehat{c}_{\boldsymbol{Y}}(\boldsymbol{u}) := \frac{1}{NH_1H_2}\sum_{m=1}^{N} k\left(\frac{\widehat{F}_{Y_1}(y_1(m)) - u_1}{H_1}\right) k\left(\frac{\widehat{F}_{Y_2}(y_2(m)) - u_2}{H_2}\right), \qquad (8.12)$$

is the kernel estimate of the copula density $c_{\boldsymbol{Y}}(\cdot)$, and $\widehat{F}_{Y_j}(x)$, $j = 1, 2$, is the smoothed estimate of the marginal distribution function $F_{Y_j}(x)$ of the random variable Y_j, at any real value $x \in \mathbb{R}$, defined by

$$\widehat{F}_{Y_j}(x) := \frac{1}{N}\sum_{m=1}^{N} K\left(\frac{y_j(m) - x}{h_j}\right), \qquad (8.13)$$

where $K(\cdot) : x \in \mathbb{R} \mapsto K(x) := \int_{-\infty}^{x} k(t)\, dt$, is the primitive of a kernel $k(\cdot)$, a symmetric centered probability density. In our forthcoming simulation study, we will use the triangular kernel

$$k(x) := (1 - |x|)\mathbb{1}_{[-1,1]}(x), \forall x \in \mathbb{R}.$$

A more appropriate choice of the kernel $k(\cdot)$, for estimating the copula density, can be done according to [125], which copes with the boundary effect. The bandwidth parameters H_1, H_2 and h_1, h_2 in (8.12) and (8.13) will be chosen according to Silverman's rule of thumb, see [138], i.e., for $j = 1, 2$, we take $H_j = \left(\frac{4}{2+2}\right)^{\frac{1}{2+4}} N^{\frac{-1}{2+4}}\,\widehat{\Sigma}_j$, and $h_j = \left(\frac{4}{3}\right)^{\frac{1}{5}} N^{\frac{-1}{5}}\,\widehat{\sigma}_j$, where $\widehat{\sigma}_j$ and $\widehat{\Sigma}_j$ are, respectively, the empirical standard deviation of the data $y_j(1), \ldots, y_j(N)$ and $\widehat{F}_{Y_j}(y_j(1)), \ldots, \widehat{F}_{Y_j}(y_j(N))$. The minimum, $\widehat{\boldsymbol{W}}$, can be computed using gradient descent algorithm on $\boldsymbol{W}$ of the criterion function $\mathscr{J}_{\text{sep}}(\boldsymbol{W})$. where $\frac{d\widehat{\mathscr{J}_{sep}}}{d\boldsymbol{W}}$ is calculated as follows:

$$\frac{d\widehat{\mathscr{J}_{sep}}}{d\boldsymbol{W}} = \frac{1}{N}\sum_{i=1}^{N} \frac{\frac{d}{d\boldsymbol{W}}\widehat{c}_{\boldsymbol{Y}}(\boldsymbol{u}(i))}{\widehat{c}_{\boldsymbol{Y}}(\boldsymbol{u}(i))} \qquad (8.14)$$

with $\frac{d}{d\boldsymbol{W}} := \left(\frac{\partial}{\partial \boldsymbol{W}_{l,j}}\right)$, $l, j = 1, 2$, $\boldsymbol{u}(i) := (\widehat{F}_{Y_1}(y_1(i)), \widehat{F}_{Y_2}(y_2(i)))^\top$ and

$$\frac{\partial \widehat{c}_{\boldsymbol{Y}}(\widehat{F}_{Y_1}(y(n)), \widehat{F}_{Y_2}(y_2(n)))}{\partial \boldsymbol{W}_{l,j}} = \frac{1}{NH_1H_2}\sum_{i=1}^{N}\prod_{j=1,j\neq l}^{2} k\left(\frac{\widehat{F}_{Y_j}(y_j(i)) - \widehat{F}_{Y_j}(y_j(n))}{H_j}\right)$$
$$\times\, k'\left(\frac{\widehat{F}_{Y_l}(y_l(i)) - \widehat{F}_{Y_l}(y_l(n))}{H_l}\right)\frac{1}{H_l}\frac{\partial(\widehat{F}_{Y_l}(y_l(i)) - \widehat{F}_{Y_l}(y_l(n)))}{\partial \boldsymbol{W}_{l,j}}, \tag{8.15}$$

with

$$\frac{\partial(\widehat{F}_{Y_l}(y_l(i))}{\partial \boldsymbol{W}_{l,j}} = \frac{1}{Nh_l}\sum_{n=1}^{N} k\left(\frac{y_l(n) - y_l(m)}{h_l}\right)(x_j(n) - x_j(i)). \tag{8.16}$$

The second term of the criterion is a regularisation term, we chose the BTV-type proposed by [58] which is derived from the bilateral filter. The idea of the bilateral filter was first proposed in [149] as a very effective one-pass filter for denoising purposes while keeping sharp edges. The idea behind this regularization is that the bilateral TV is able to smooth away the noise and small variation in a signal while preserving the major edges or discontinuity. Also, it is more robust to remove a high level of noise, better than the total variation TV, see for example [58]. The bilateral TV regularization term takes the following form:

$$BTV(\boldsymbol{y}) := \sum_{j=-m}^{m} \alpha^{|j|}\|\boldsymbol{y} - \boldsymbol{G}^j\boldsymbol{y}\|_1, \tag{8.17}$$

where the matrix $\boldsymbol{G}^j$ implies a right shift of j samples. The scalar weight α ($0 < \alpha < 1$), is applied to give a spatially decaying effect to the summation of the regularization terms, and m is the spatial window size. It is easy to show that this regularization method is a generalization of other popular regularization methods. If we limit set $m = 1$, with $\alpha = 1$, and define the operator $\boldsymbol{Q} = \boldsymbol{I} - \boldsymbol{G}$ as a first derivative term, then equation (8.17) becomes

$$BTV(\boldsymbol{y}) := \|\boldsymbol{Q}\boldsymbol{y}\|_1, \tag{8.18}$$

which coincides with the total variation regularization.

Using the bilateral TV regularization (8.17), the second term of the criterion to minimize will be of the form:

$$\mathcal{J}_{\text{reg}}(\boldsymbol{y}) := \mu\sum_{j=-m}^{m} \alpha^{|j|}\|\boldsymbol{y}(t) - \boldsymbol{G}^j\boldsymbol{y}(t)\|_1,\ \mu > 0, \tag{8.19}$$

where μ are regularization parameters that should be wisely chosen by the user. The demixing matrix $\boldsymbol{W}$ can be computed using a gradient descent type algorithm. In fact, the gradient in $\boldsymbol{W}$ of the estimated criterion $\boldsymbol{W} \to \widehat{\mathcal{J}}(\boldsymbol{W})$ is calculated as follows:

$$\frac{d\mathcal{J}}{d\boldsymbol{W}} = \frac{d\mathcal{J}_{sep}}{d\boldsymbol{W}} + \frac{\mu}{N}\sum_{n=1}^{N}\sum_{j=-m}^{m} \alpha^{|j|}\left(I - \boldsymbol{G}^{-j}\right) sign(\boldsymbol{y}(n) - \boldsymbol{G}^j\boldsymbol{y}(n)) \tag{8.20}$$

We can then derive the following Algorithm:

```
**Data:** $\boldsymbol{x}$ the two observed images
**Result:** $\widehat{\boldsymbol{s}}$ the estimated source signal
**Initialization:** $\boldsymbol{W}^{(0)} = \boldsymbol{I}_p$, $\boldsymbol{y}^{(0)} = \boldsymbol{W}^{(0)}\,\boldsymbol{x}$.
Given $\varepsilon > 0$, $\nu > 0$.
**Do**: Update $\boldsymbol{W}$ and $\boldsymbol{y}$:
    $\boldsymbol{W}^{(q+1)} = \boldsymbol{W}^{(q)} - \nu \frac{d\widehat{\mathcal{J}}(\boldsymbol{W})}{d\boldsymbol{W}}$
$\boldsymbol{y}^{(q+1)} = \boldsymbol{W}^{(q+1)}\,\boldsymbol{x}$.
**Until** $||\boldsymbol{W}^{(q+1)} - \boldsymbol{W}^{(q)}|| < \epsilon$
$\widehat{\boldsymbol{s}} = \boldsymbol{y}^{(q+1)}$.
```

Algorithm 6: BSS algorithm for independent source components.

Salt & pepper and Gaussian noise

Th second type of noise that we will tackle is the combination of Salt & pepper and Gaussian noise, the new model is written as follows:

$$\boldsymbol{I} = \boldsymbol{I_0} + \boldsymbol{n} + \boldsymbol{s} \tag{8.21}$$

where $\boldsymbol{n}$ is the additive Gaussian noise and $\boldsymbol{s}$ is the additive Salt & pepper noise. In this case to use BSS we need two other observations, due to having three sources to be separated $(\boldsymbol{I_0}, \boldsymbol{n}, \boldsymbol{s})$. The idea here is to use two different filters to get two different observations $\boldsymbol{I_1}, \boldsymbol{I_2}$. The following figure 8.2 shows our denoising process:

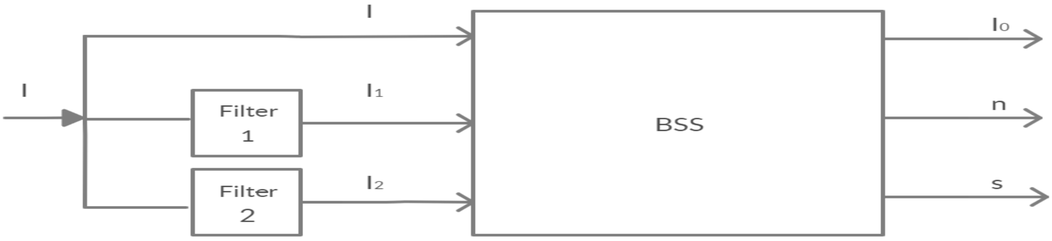


Figure 8.2: The BSS process for denoising Salt & pepper and Gaussian noise contaminated images

Hence, the current BSS model is written as follows:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{I}_1 \\ \boldsymbol{I}_2 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \times \begin{bmatrix} \boldsymbol{I}_0 \\ \boldsymbol{n} \\ \boldsymbol{s} \end{bmatrix} = \boldsymbol{M}\,\boldsymbol{S} \tag{8.22}$$

with $\boldsymbol{M}$ the mixing matrix where $m_{11} = m_{12} = m_{13} = 1$. The goal is to estimate $\boldsymbol{y} = (y_1, y_2, y_3)$ of $(\boldsymbol{I_0}, \boldsymbol{n}, \boldsymbol{s})$ by finding a demixing matrix $\boldsymbol{W}$ that should be as similar as possible to $\boldsymbol{M}^{-1}$. To separate the mixture we minimize a criterion of this form:

$$\mathcal{J} : \boldsymbol{W} \to \mathcal{J}(\boldsymbol{W}) := \mathcal{J}_{sep}(\boldsymbol{W}) + \mathcal{J}_{reg}(\boldsymbol{y}) \tag{8.23}$$

The regularization term is the same as the previous section, and the separation term take the following form:

$$\begin{aligned}\mathcal{J}_{sep}(\boldsymbol{W}) := \mathbb{KL}\left(c_{\boldsymbol{Y}} || c_0\right) &:= \int_{[0,1]^3} - \log\left(\frac{1}{c_{\boldsymbol{Y}}(\boldsymbol{u})}\right) c_{\boldsymbol{Y}}(\boldsymbol{u})\, d\boldsymbol{u} \qquad (8.24)\\ &= \mathbb{E}\left[\log c_{\mathbf{Y}}\left(F_{Y_1}(Y_1), F_{Y_2}(Y_2), F_{Y_3}(Y_3)\right)\right] \qquad (8.25)\end{aligned}$$

The gradient of the two previous terms can be explicitly computer using a similar approach to (8.15) and (8.20) above.

8.2.3 Poison noise

In this case, we consider the problem of a signal-dependent noise component. In particular, we will focus on a Poisson distributed component $\boldsymbol{z}$ having u as Poisson parameter. In other words, we will consider the following denoising model (8.26) with

$$\boldsymbol{I} = \boldsymbol{I_0} + \boldsymbol{z} \;\; where\, \boldsymbol{z} \sim \;Pois(u) \qquad (8.26)$$

As in section 8.2.2, the idea is to create a second observation from $\boldsymbol{I}$ denoted $\boldsymbol{I'}$. Due to the fact that Poison noise is signal dependent then the two source components $\boldsymbol{S} = (\boldsymbol{I}_0, \boldsymbol{z})^\top$ are statistically dependent, hence we can't use the independent copula density as in the previous section. Denote by $c_{\boldsymbol{S}}(\cdot)$ the unknown copula density of $\boldsymbol{S}$. The source copula density $c_{\boldsymbol{S}}(\cdot)$ is unknown. We assume that it belongs to a set of L candidate semiparametric models, say,

$$M_l := \{c_l(\cdot; \theta_l);\, \theta_l \in \Theta_l \subset \mathbb{R}\},\, l = 1, \dots, L. \qquad (8.27)$$

We refer to [88] for many examples of semiparametric copula density models, more details were given in the "Introduction to copula" chapter. We assume that each "semiparametric" model M_l, $l = 1, \dots, L$, satisfies the following identifiability condition : for any regular matrix $\boldsymbol{M}$, if the copula density, of $\boldsymbol{MS}$, belongs to $\{c_l(\cdot; \theta_l);\, \theta_l \in \Theta_l \subset \mathbb{R}\}$, then $\boldsymbol{M} = \boldsymbol{DP}$, where $\boldsymbol{D}$ is diagonal and $\boldsymbol{P}$ is a permutation. We introduce then the following objective function

$$\boldsymbol{W} \mapsto \mathcal{J}(\boldsymbol{W}) := \mathcal{J}_{\text{sep}}(\boldsymbol{W}) + \mathcal{J}_{reg}(\boldsymbol{y}),$$

where $\mathcal{J}_{reg}(\boldsymbol{y})$ is the same as in the previous section. The new separating term of the form:

$$\mathcal{J}_{\text{sep}}(\boldsymbol{W}) := \mathbb{KL}\left(c_{\boldsymbol{Y}} || c_l(\cdot; \theta_l)\right) \qquad (8.28)$$

This term is nonnegative and achieves its minimum value zero iff $\boldsymbol{W} = \boldsymbol{M}^{-1}$ (up to scale and permutation indeterminacies). Therefore, we propose to estimate, from the i.i.d. data (8.4), the demixing matrix by

$$\widehat{\boldsymbol{W}} := \arg\inf_{\boldsymbol{W}} \inf_{\theta_{l^*} \in \Theta_{l^*}} \widehat{\mathcal{J}}(\boldsymbol{W}), \qquad (8.29)$$

where

$$l^* = \arg\min_{l=1,\dots,L} \inf_{\boldsymbol{W}} \inf_{\theta_l \in \Theta_l} \widehat{\mathbb{KL}}\left(c_{\boldsymbol{Y}}, c_l(\cdot; \theta_l)\right), \qquad (8.30)$$

and

$$\widehat{\mathbb{KL}}\left(c_{\boldsymbol{Y}}||c_l(\cdot;\theta_l)\right) := \frac{1}{N}\sum_{n=1}^{N}\log\left(\frac{\widehat{c}_{\boldsymbol{Y}}(\widehat{F}_{Y_1}(y_1(n)),\widehat{F}_{Y_2}(y_2(n)))}{c_l(\widehat{F}_{Y_1}(y_1(n)),\widehat{F}_{Y_2}(y_2(n));\theta_l)}\right) \tag{8.31}$$

The minimum, $\widehat{\boldsymbol{W}}$, in (8.31) can be computed using gradient descent algorithm on both $\boldsymbol{W}$ and θ of the criterion function $(\boldsymbol{W},\theta) \mapsto \widehat{\mathbb{KL}}\left(c_{\boldsymbol{Y}}||c_l(\cdot;\theta_l)\right) + \mathcal{J}_{reg}(\boldsymbol{y})$ for each model and then choose the solution minimizing the criterion over all considered models where

$$\frac{d\widehat{\mathcal{J}_{sep}}}{d\boldsymbol{W}} = \frac{1}{N}\sum_{i=1}^{N}\frac{\frac{d}{d\boldsymbol{W}}\widehat{c}_{\boldsymbol{Y}}(\boldsymbol{u}(i))}{\widehat{c}_{\boldsymbol{Y}}(\boldsymbol{u}(i))}\,\frac{\widehat{c}_{\widehat{\theta}}(\boldsymbol{u}(i))}{\frac{d}{d\boldsymbol{W}}\widehat{c}_{\widehat{\theta}}(\boldsymbol{u}(i))} \tag{8.32}$$

with $\boldsymbol{u}(i) := (\widehat{F}_{Y_1}(y_1(i)),\widehat{F}_{Y_2}(y_2(i)))^\top$, and $\frac{d\widehat{\mathcal{J}_{reg}}}{d\boldsymbol{W}}$ is the same as in section 8.2.2. The two gradients $\dfrac{d}{d\boldsymbol{W}}\widehat{c}_{\boldsymbol{Y}}(\boldsymbol{u}(i))$ and $\dfrac{d}{d\boldsymbol{W}}\widehat{c}_{\widehat{\theta}}(\boldsymbol{u}(i))$ can be explicitly computer using a similar approach to (8.15) and (8.20) above. We obtain the following estimates of the sources, see (8.2),

$$\widehat{\boldsymbol{s}}(m) := \widehat{\boldsymbol{W}}\,\boldsymbol{x}(m),\; m = 1,\ldots,N_t. \tag{8.33}$$

We obtain then the following Algorithm 7.

```
Data: $\boldsymbol{x}$ the two observed images
Result: $\widehat{\boldsymbol{s}}$ the estimated source signal
Initialization: $\boldsymbol{W}^{(0)} = \boldsymbol{I}_p$, $\boldsymbol{y}^{(0)} = \boldsymbol{W}^{(0)}\,\boldsymbol{x}$.
Given $\varepsilon > 0$, $\nu > 0$.
Do: Update $\boldsymbol{W}$, $\boldsymbol{\theta}$ and $\boldsymbol{y}$:
    $\boldsymbol{\theta}^{(q+1)} = \boldsymbol{\theta}^{(q)} - \nu\dfrac{d\widehat{\mathcal{J}^{\text{dep}}_{\text{sep}}(\boldsymbol{W})}}{d\boldsymbol{\theta}}$
    $\boldsymbol{W}^{(q+1)} = \boldsymbol{W}^{(q)} - \nu\dfrac{d\widehat{\mathcal{J}^{\text{dep}}_{\text{sep}}(\boldsymbol{W})}}{d\boldsymbol{W}}$
    $\boldsymbol{y}^{(q+1)} = \boldsymbol{W}^{(q+1)}\boldsymbol{x}$.
Until $||\boldsymbol{W}^{(q+1)} - \boldsymbol{W}^{(q)}|| < \epsilon$
    $\widehat{\boldsymbol{s}} = \boldsymbol{y}^{(q+1)}$.
```

Algorithm 7: BSS algorithm for Poison noise.

8.2.4 Results

In this section, we give simulation results for the proposed method. We dealt with various types of images corrupted with different levels of noises. To evaluate our approach, we use both the structural similarity (SSIM) index and the classical peak signal to noise ratio (PSNR) measure. Although PSNR is widely used as a comparative index between two images, many works have pointed out that PSNR is not a good fit to measure the perceptual similarity between two images like the SSIM. The PSNR is defined as:

$$PSNR = 10log_{10}\frac{MAX^2}{MSE} \tag{8.34}$$

where the MSE (Mean Square Root) of two monochrome images u and v of size $m \times n$ is calculated as follows

$$MSE = \frac{1}{m \times n}\sum_{i=1}^{m} -1 \sum_{j=1}^{n} -1 \parallel v(i,j) - u(i,j) \parallel^2 \tag{8.35}$$

and the MAX is the maximum possible pixel value of the image, and because we are using only gray-scaled image this MAX value is equal to 255.

The SSIM index is calculated on various windows of an image. The measure between two windows x and y is

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{8.36}$$

where μ_y and μ_x are the average value of window x and y, respectively. σ_x and σ_y are the variance of x and y, and σ_{xy} is the covariance of x and y. $c_1 = (k_1R)^2$ and $c_2 = (k_2R)^2$ are two variables to stabilize the division with k_1 and k_2 set respectively as 0.01 and 0.03 by default and R is the the dynamic range and it is equal to 255 if the image is 8-bit. The SSIM has a maximum value of 1. The maximum value of 1 indicates that the two signals are perfectly structurally similar while a value of 0 indicates no structural similarity.

In the following we present the results of denoising multiple images corrupted by various types of noise.

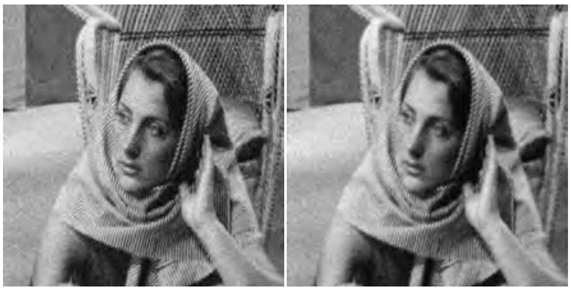


Figure 8.3

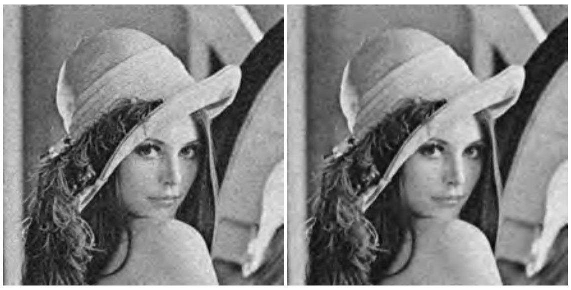


Figure 8.4

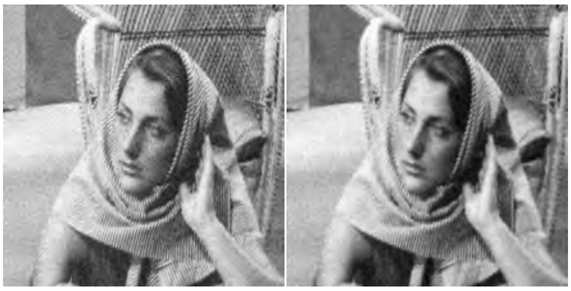


Figure 8.5

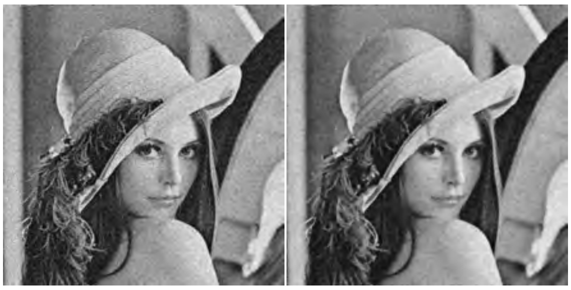


Figure 8.6

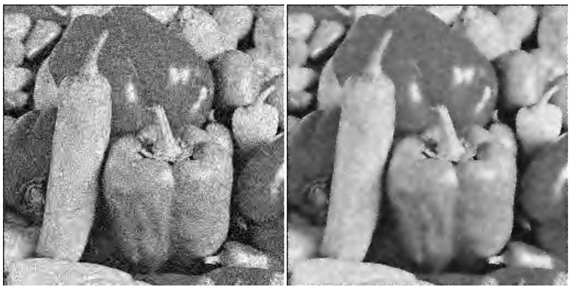


Figure 8.7

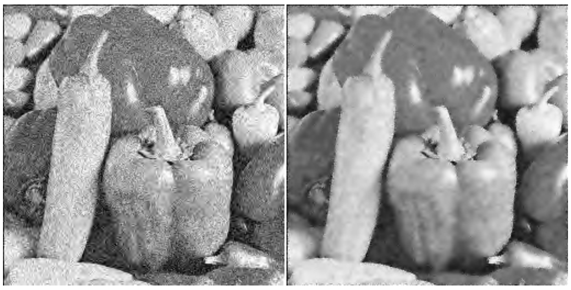


Figure 8.9

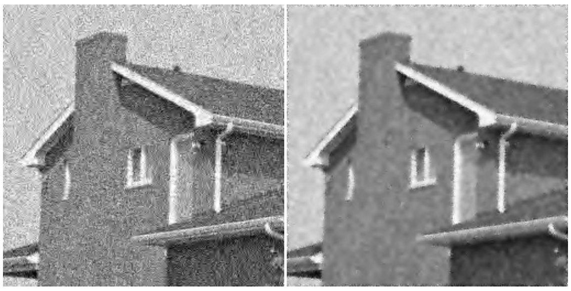


Figure 8.8

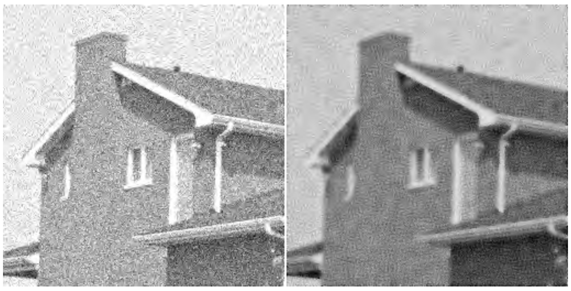


Figure 8.10

8.3 Separation of fingerprints

A second application in the field of digital images is the separation of mixtures of images of fingerprints. The uniqueness of a fingerprint is one of the biological features of humans. Fingerprints are one of these characteristics which is not changed or changed slowly by time and is permanent moreover is it easy to test, which mean, we can measure the similarity of two fingerprints by simple skills or equipment.

Fingerprint recognition has become an important tool in recent years, and it is used for identification purposes for protection and privacy on a regular basis. It does, however, have a problem when fingerprints pattern are on the same position. The problem happens in crime scene investigation normally. Therefore, researchers were more interested in how to repair or reconstruct overlapping fingerprints. Our proposed method focus on separating and reconstructing two fingerprints mixtures.

Our idea is to use BSS as the tool to separate those fingerprint mixtures. We take the case of two fingerprints. This problem can be modelled as follows:

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \times \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \tag{8.37}$$

where $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ is the mixing matrix and I_1, I_2 are the two fingerprint images that we want to get. We only have their mixed images X_1 and X_2, For estimating the demixing matrix and the original fingerprints, each image will be converted to grayscale, and will be considered as 1d-signal of dimension. We use the algorithm proposed in chapter 6 to achieve the separation. To test the accuracy of our approach we don't use the PSNR in this case as it can contribute to some misleading results especially because we are talking about fingerprints. Instead we used a similarity Convolutive Neural Network (CNN) method to test the similarity. This Image Similarity algorithm compares two images and returns a value that tells you how visually similar they are. The lower the the score, the more contextually similar the two images are with a score of zero being identical.

8.3.1 Simulation Results

The two original fingerprints were mixed by $A = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$ mixing matrix. Figure.8.11 present those two mixtures. Using our algorithm on these observations leads to the figure.8.12, where we estimated the original mixtures. And by deploying the CNN image similarity algorithm we got the following results.

- **First Fingerprint**
  - Similarity between original first fingerprint and the first mixture: 8
  - Similarity between original first fingerprint and the BSS-retrieved fingerprint: 0
- **Second Fingerprint**
  - Similarity between original second fingerprint and the second mixture: 7
  - Similarity between original second fingerprint and the BSS-retrieved fingerprint: 0

We can say that the separation was a success as we got a score of zero which mean perfect similarity.

(a) First Mix

(b) Second Mix

Figure 8.11: The mixture of the two fingerprints

(a) The 1st FP After applying BSS (b) The 2nd FP After applying BSS

Figure 8.12: The reconstituted fingerprints

8.4 Digital document image restoration

Another application of BSS is removing the bleed-through/show-through effects which is detected either in the antique documents due to the process of rewriting the erased parts and the intensity of the ink that appears in the reverse side of the document, or in the double-side scanned documents, due to the transparency of the paper. It is obvious that such intervening strokes, will get in the way if an optical character recognition (OCR) system is to function effectively. To remedy to this problem, we suggest a restoration method based on our approach of blind source separation which is based on copula theory that models the dependency structure, with the aim of improving text readability and OCR efficiency.

8.4.1 Proposed approach

In this section we assume that a document which is affected by the bleed-through/show-through effect is viewed as the superposition of three sources called respectively background, underwriting, and overwriting, consequently, In our BSS problem, we have three different sources that in one way or another fused to provide the studied image. Simultaneously, we may believe that there are three observed charts that were divided into red, green and blue components. As stated above, we assume having an equality between the number of sources and observations, we may also denote that the color of each of the three sources is indexed as follow (r_1, g_1, b_1) for the background, (r_2, g_2, b_2) for the overwriting, and (r_3, g_3, b_3) for the underwriting. Therefore, the BSS problem is stated as follows :

$$\begin{bmatrix} u_r(t) \\ u_g(t) \\ u_b(t) \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 \\ g_1 & g_2 & g_3 \\ b_1 & b_2 & b_3 \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \end{bmatrix}$$

where $u(t)$ denotes the observations, $D = \begin{bmatrix} r_1 & r_2 & r_3 \\ g_1 & g_2 & g_3 \\ b_1 & b_2 & b_3 \end{bmatrix}$ the mixing matrix that belongs to $\mathbb{R}^{3\times3}$, and $s(t)$ is the source that we want to deduce in the end of this process by the next equation :

$$\mathbf{w}(t) = \mathbf{Z}\mathbf{u}(t),$$

where $w(t)$ denotes the estimate value of $s(t)$. Notice that W, U and S are three random variables of $\mathbb{R}^3$ whom realizations are respectively w, u and s.

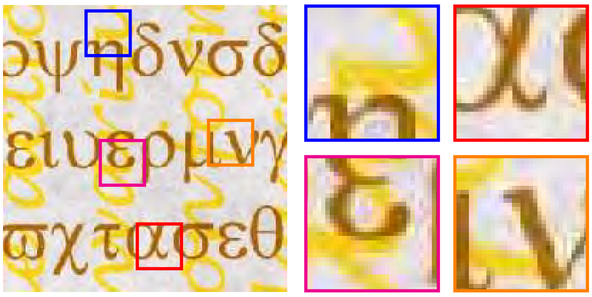
Under the dependency condition of the source components, we denote by $c_{\boldsymbol{S}}(\cdot)$ the copula density of the random variable S which is assumed unknown. To retrieve the de-mixing matrix which is assumed to be as close as possible to D^{-1}, one must introduce the following objectif function $\boldsymbol{Z} \mapsto KL\,(c_W, c_S)$, where

$$\begin{aligned} KL\,(c_W, c_S) : &= \int_{[0,1]^3} \log\left(\frac{c_W(\boldsymbol{y})}{c_S(\boldsymbol{y})}\right) c_W(\boldsymbol{y}) d\boldsymbol{y} \\ &= \mathbb{E}\left[\log \frac{c_W\left(F_{W_1}\left(W_1\right), \ldots, F_{W_3}\left(W_3\right)\right)}{\mathrm{c_S}\left(F_{W_1}\left(W_1\right), \ldots, F_{W_3}\left(W_3\right)\right)}\right], \end{aligned}$$

where KL denotes the Kullback-Liebler divergence between the copula density of the source components and the copula density of the observations. We use the procedure explained in chapter ”Instantaneous BSS via copulas”.

8.4.2 Simulation results

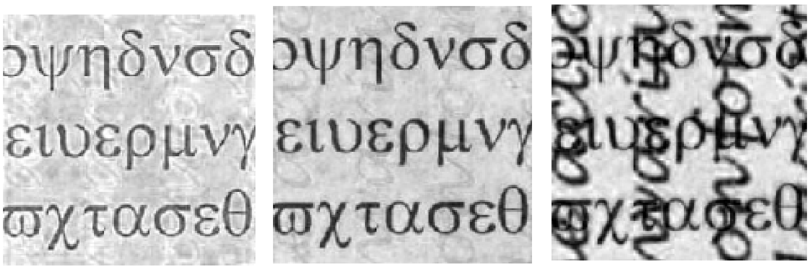
In this section, we will show the efficiency of our approach dealing with the bleed-through/show-through effect [150] by exposing the results using the BSS process based on copulas. To do so, we used a real image of ancient document as shown in figure 8.13.



(a) (b)

Figure 8.13: (a) Image 1, (b) Zoom on four regions

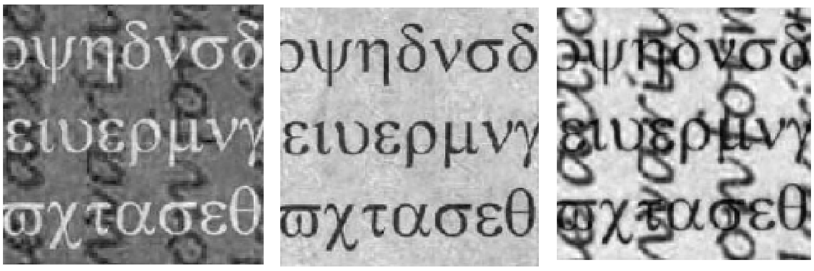
As we can notice on the four zooming regions, the image is truly affected by the bleed-through effect, this image is the superposition of three observations u_r, u_g and u_b given by figure 8.14.



(a) (b) (c)

Figure 8.14: (a) Image corresponding to u_r, (b) Image corresponding to u_g, (c) Image corresponding to u_b

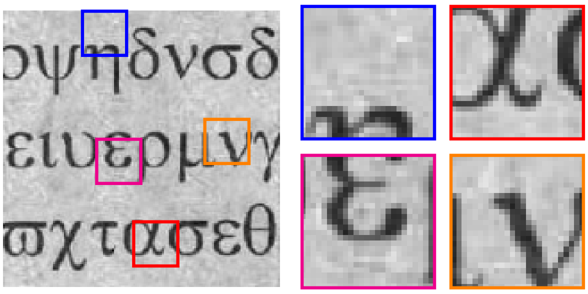
As stated above in section 8.4.1 the dependency structure is unknown between the source elements, that's why we will use our approach with some copulas examples namely : Student T, Gaussian, FGM, Clayton, AMH, Frank, and Gumbel copula, we refer the readers to [88] for the definitions of these copula models. After simulations, the perfect candidate that allows us to reach the separation with $\theta = 1.5$ is Gumbel copula. By applying our approach, we obtain the results presented in figure 8.15.



(a) (b) (c)

Figure 8.15: Restored images by our approach :(a) Image corresponding to s_r, (b) Image corresponding to s_g, (c) Image corresponding to s_b

As we can notice in figure 8.15, the bleed-through effect is well removed from the image 8.15b and the image is well restored, and as a proof we zoom on four regions in figure 8.16.



(a) (b)

Figure 8.16: (a) Restored image, (b) Zoom on four regions of the restored image

8.5 Conclusion

In this chapter we presented some of the applications of Blind Source Separation that we conducted during our research. As we saw all of them were a success in term of quality of results. However, they are much more advanced applications where BSS can be very useful such as hearing aid applications and its real-time implementation on smartphone and so on, and our goal in the future is to implement our approach in those advanced applications and improve them even more to give perfect results.

Chapter 9

Conclusion and perspectives

In this thesis, We have provided algorithms for blind source separation for linear instantaneous mixtures . The idea behind all of the algorithms is to use the steepest descent gradient algorithm to iteratively minimize appropriate criteria based on copula densities. The first approach uses the mutual information (MI) which is equivalent to the Kullback-Leibler divergence between the copula of the estimated sources and the particular copula of independence, this leads to defining a new separation criteria in order to separate instantaneous mixtures of possibly dependent source components by finding an appropriate copula that models the dependency structure of the sources using model selection approaches. This approach can separate dependent sources with unknown copula model and/or unknown parameter which is the case in many real applications almost perfectly for instantaneous mixtures. It gave very satisfying results, however to have even greater results with a faster divergence, we proposed to use the alpha-divergence. Note that the kullback-leibler divergence is just a special case of the alpha-divergence. We show that the proposed approach can magnificently separate instantaneous mixtures of dependent sources with unknown copula model and/or unknown parameter which is the case in many real applications. The most interesting results of the thesis are the followings:

- We introduced the concept of copula, as the statistical object, in order to model the dependency structure between components sources.

- We introduced a new robust BSS approach for noisy mixtures of independent and dependent sources, using the bilateral total variation regularization (BTV) and copulas.

- We introduced the alpha-divergence, as our new cost function to minimize considering its superiority to handle noisy data as well as its ability to converge faster.

- We designed new methods for separating linear instantaneous mixtures of dependent sources when for three case, where the parameter is known and when only the copula model is known and when the copula model and the parameter are unknown. These methods are presented in the noisy-free case and also noisy cases, using the two concepts of copula and alpha-divergence.

- We tested our approach in multiple real-life applications, especially in the the image processing field such as image denoising and restoration.

In the future, we would like to test and use our new approach on more advanced applications and even propose new approach in BSS more complicated cases such as the convolutive case on even the non-linear case where most of the approaches proposed in literature wont even give an average solution. BSS is such an interesting research fiels and its applications are far more interesting and it is helping in multiple scientific fields and not exclusive to one problem, it can be used in any problem as long as there are sources that should be separated such as audio, images, radar-signals, vibrations, and so forth, hence we want to use real data and real case scenarios especially in the bio-medical field. Another extension that we can make is to optimize our second approach by minimizing over the parameter alpha to get the perfect separation.

Bibliography

[1] Abayomi, K., U. Lall, and V. de la Pena. 2008. Copula based independent component analysis, *Avaiable at SSRN: http://ssrn. com/abstract= 1028822*.

[2] Abrard, Frederic and Yannick Deville. 2005. A time–frequency blind signal separation method applicable to underdetermined mixtures of dependent sources, *Signal processing*, 85(7), 1389–1403.

[3] Akaike, Hirotugu. 1974. A new look at the statistical model identification, *IEEE transactions on automatic control*, 19(6), 716–723.

[4] Albright, Russell, James Cox, David Duling, Amy N Langville, and C Meyer. 2006. Algorithms, initializations, and convergence for the nonnegative matrix factorization, Tech. rep., Tech. rep. 919. NCSU Technical Report Math 81706. http://meyer. math. ncsu ....

[5] Ali, M., N. Mikhail, and M. Haq. 1978. A class of bivariate distributions including the bivariate logistic, *J. Multivariate Anal.*, 8(3), 405–412.

[6] Ali, S. and S. Silvey. 1966. A general class of coefficients of divergence of one distribution from another, *Journal of the Royal Statistical Society. Series B*, 28(1), 131–142.

[7] Almeida, Luís B and Aapo Hyvärinen. 2005. Separating a real-life nonlinear image mixture., *Journal of Machine Learning Research*, 6(7).

[8] Ans, Bernard. 1985. Adaptive neural architectures, *Detection of primitives. Proceedings of COGNITIVA'85*, 593–597.

[9] Arberet, Simon. 2008. *Estimation robuste et apprentissage aveugle de modèles pour la séparation de sources sonores*, Ph.D. thesis, Université Rennes 1.

[10] Aubert, Gilles and Pierre Kornprobst. 2006. *Mathematical problems in image processing: partial differential equations and the calculus of variations*, vol. 147, Springer Science & Business Media.

[11] Babaie-Zadeh, M. and C. Jutten. 2005. A general approach for mutual information minimization and its application to blind source separation, *Signal Processing*, 85(5), 975–995.

[12] Bach, F. and M. Jordan. 2002. Kernel independent component analysis, *Journal of machine learning research*, 3(Jul), 1–48.

[13] Banerjee, Arindam, Inderjit Dhillon, Joydeep Ghosh, Srujana Merugu, and Dharmendra S Modha. 2007. A generalized maximum entropy approach to bregman co-clustering and matrix approximation, *The Journal of Machine Learning Research*, 8, 1919–1986.

[14] Banerjee, Arindam, Srujana Merugu, Inderjit S Dhillon, Joydeep Ghosh, and John Lafferty. 2005. Clustering with bregman divergences., *Journal of machine learning research*, 6(10).

[15] Bar-Ness, Y, JW Carlin, and ML Steinberger. 1982. Bootstrapping adaptive interference cancelers-some practical limitations, in *Globecom'82-Global Telecommunications Conference*, vol. 3, 1251–1255.

[16] Bauschke, Heinz H. 2003. Duality for bregman projections onto translated cones and affine subspaces, *Journal of Approximation Theory*, 121(1), 1–12.

[17] Bell, A. and T. Sejnowski. 1995. An information-maximization approach to blind separation and blind deconvolution, *Neural computation*, 7(6), 1129–1159.

[18] Bercher, J-F. 2008. On some entropy functionals derived from rényi information divergence, *Information Sciences*, 178(12), 2489–2506.

[19] Bro, Rasmus and Sijmen De Jong. 1997. A fast non-negativity-constrained least squares algorithm, *Journal of Chemometrics: A Journal of the Chemometrics Society*, 11(5), 393–401.

[20] Buades, Antoni, Bartomeu Coll, and J-M Morel. 2005. A non-local algorithm for image denoising, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE, vol. 2, 60–65.

[21] Burel, Gilles. 1992. Blind separation of sources: A nonlinear neural algorithm, *Neural networks*, 5(6), 937–947.

[22] Capdevielle, Véronique, Christine Serviere, and Jean-Louis Lacoume. 1995. Blind separation of wide-band sources in the frequency domain, in *1995 International Conference on Acoustics, Speech, and Signal Processing*, IEEE, vol. 3, 2080–2083.

[23] Cardoso, J. 1993. A. souloumiac, blind signal beamforming for non gaussian signals, *Proceedings of the IEE*, 140(6), 362–370.

[24] Cardoso, J. 1998. Multidimensional independent component analysis, in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, IEEE, vol. 4, 1941–1944.

[25] Cardosot, J. 1998. Blind signal separation: statistical principles, *Proceedings of the IEEE*, 86(10), 2009–2025.

[26] Chambolle, Antonin. 2004. An algorithm for total variation minimization and applications, *Journal of Mathematical imaging and vision*, 20(1), 89–97.

[27] Chen, X. and Y. Fan. 2006. Estimation and model selection of semiparametric copula-based multivariate dynamic models under copula misspecification, *J. Econometrics*, 135(1-2), 125–154.

[28] Cichocki, A. and S. Amari. 2002. *Adaptive blind signal and image processing: learning algorithms and applications*, John Wiley & Sons.

[29] Cichocki, Andrzej, Shun-ichi Amari, Rafal Zdunek, Raul Kompass, Gen Hori, and Zhaohui He. 2006. Extended smart algorithms for non-negative matrix factorization, in *International Conference on Artificial Intelligence and Soft Computing*, Springer, 548–562.

[30] Cichocki, Andrzej, Rafal Zdunek, and Shun-ichi Amari. 2006. Csiszar's divergences for non-negative matrix factorization: Family of new algorithms, in *International Conference on Independent Component Analysis and Signal Separation*, Springer, 32–39.

[31] Cichocki, Andrzej, Rafal Zdunek, and Shun-ichi Amari. 2007. Nonnegative matrix and tensor factorization [lecture notes], *IEEE signal processing magazine*, 25(1), 142–145.

[32] Cichocki, Andrzej, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. 2009. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*, John Wiley & Sons.

[33] Cichocki, Andrzej and Jacek M Zurada. 2004. Blind signal separation and extraction: Recent trends, future perspectives, and applications, in *International Conference on Artificial Intelligence and Soft Computing*, Springer, 30–37.

[34] Clayton, D. 1978. A model for association in bivariate life tables and its application in epidemiological studies of familial tendency in chronic disease incidence, *Biometrika*, 65(1), 141–151.

[35] Collins, Michael, Robert E Schapire, and Yoram Singer. 2002. Logistic regression, adaboost and bregman distances, *Machine Learning*, 48(1), 253–285.

[36] Combettes, Patrick L and Jean-Christophe Pesquet. 2004. Wavelet-constrained image restoration, *International Journal of Wavelets, Multiresolution and Information Processing*, 2(04), 371–389.

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

[38] Comon, P. and C. Jutten. 2010. *Handbook of blind source separation : independent component analysis and applications*, Communications engineering, Elsevier.

[39] Comon, Pierre. 1990. Analyse en composantes indépendantes et identification aveugle, *Traitement du signal*, 7(5), 435–450.

[40] Comon, Pierre. 1994. Independent component analysis, a new concept?, *Signal processing*, 36(3), 287–314.

[41] Comon, Pierre and Christian Jutten. 2010. *Handbook of Blind Source Separation: Independent component analysis and applications*, Academic press.

[42] Coupé, Pierrick, Pierre Yger, Sylvain Prima, Pierre Hellier, Charles Kervrann, and Christian Barillot. 2008. An optimized blockwise nonlocal means denoising filter for 3-d magnetic resonance images, *IEEE transactions on medical imaging*, 27(4), 425–441.

[43] Cressie, N. and T. Read. 1984. Multinomial goodness-of-fit tests, *Journal of the Royal Statistical Society: Serie B*, 46(3), 440–464.

[44] Csiszár, Imre. 1967. Information-type measures of difference of probability distributions and indirect observation, *studia scientiarum Mathematicarum Hungarica*, 2, 229–318.

[45] Da Silva, Ricardo Dutra, Rodrigo Minetto, William Robson Schwartz, and Helio Pedrini. 2013. Adaptive edge-preserving image denoising using wavelet transforms, *Pattern analysis and applications*, 16(4), 567–580.

[46] Deheuvels, Paul. 1979. La fonction de dépendance empirique et ses propriétés. un test non paramétrique d'indépendance, *Bulletins de l'Académie Royale de Belgique*, 65(1), 274–292.

[47] Della Pietra, Stephen, Vincent Della Pietra, and John Lafferty. 2001. Duality and auxiliary functions for bregman distances, Tech. rep., CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE.

[48] Deville, Yannick. 2015. Matrix factorization for bilinear blind source separation: Methods, separability and conditioning, in *2015 23rd European Signal Processing Conference (EUSIPCO)*, IEEE, 1900–1904.

[49] Deville, Yannick and Matthieu Puigt. 2007. Temporal and time-frequency correlation-based blind source separation methods. part i: Determined and underdetermined linear instantaneous mixtures, *Signal Processing*, 87(3), 374–407.

[50] Dhillon, Inderjit S and Suvrit Sra. 2005. Generalized nonnegative matrix approximations with bregman divergences, in *NIPS*, Citeseer, vol. 18.

[51] Dong, Weisheng, Lei Zhang, and Guangming Shi. 2011. Centralized sparse representation for image restoration, in *2011 International Conference on Computer Vision*, IEEE, 1259–1266.

[52] Donoho, David L and Victoria C Stodden. 2004. When does non-negative matrix factorization give a correct decomposition into parts?

[53] Dragomir, S. 2000. *Inequalities for Csiszar f-Divergence in Information Theory,Victoria University: Melbourne, Australia.*

[54] El Rhabi, M., H. Fenniri, A. Keziou, and E. Moreau. 2013. A robust algorithm for convolutive blind source separation in presence of noise, *Signal Processing*, 93(4), 818–827.

[55] El Rhabi, M., G. Gelle, H. Fenniri, and G. Delauna. 2004. A penalized mutual information criterion for blind separation of convolutive mixtures, *Signal Processing*, 84(10), 1979–1984.

[56] Elad, Michael and Michal Aharon. 2006. Image denoising via sparse and redundant representations over learned dictionaries, *IEEE Transactions on Image processing*, 15(12), 3736–3745.

[57] Fan, Linwei, Xuemei Li, Qiang Guo, and Caiming Zhang. 2018. Nonlocal image denoising using edge-based similarity metric and adaptive parameter selection., *Sci. China Inf. Sci.*, 61(4), 049101–1.

[58] Farsiu, Sina, M Dirk Robinson, Michael Elad, and Peyman Milanfar. 2004. Fast and robust multiframe super resolution, *IEEE transactions on image processing*, 13(10), 1327–1344.

[59] Fermanian, Jean-David. 2005. Goodness-of-fit tests for copulas, *Journal of multivariate analysis*, 95(1), 119–152.

[60] Ferrari, Davide, Yuhong Yang, et al. 2010. Maximum lq-likelihood estimation, *The Annals of Statistics*, 38(2), 753–783.

[61] Févotte, Cédric, Nancy Bertin, and Jean-Louis Durrieu. 2009. Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis, *Neural computation*, 21(3), 793–830.

[62] Févotte, Cédric and Jérôme Idier. 2011. Algorithms for nonnegative matrix factorization with the β-divergence, *Neural computation*, 23(9), 2421–2456.

[63] Frank, M. 1979. On the simultaneous associativity of $F(x, y)$ and $x + y - F(x, y)$, *Aequationes Math.*, 19(2-3), 194–226.

[64] Fréchet, Maurice. 1951. Sur les tableaux de corrélation dont les marges sont données, *Ann. Univ. Lyon, 3ˆ e serie, Sciences, Sect. A*, 14, 53–77.

[65] Fujimoto, Yu and Noboru Murata. 2007. A modified em algorithm for mixture models based on bregman divergence, *Annals of the Institute of Statistical Mathematics*, 59(1), 3–25.

[66] Gajbhar, Shrishail S and Manjunath V Joshi. 2013. Image denoising using redundant finer directional wavelet transform, in *2013 Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, IEEE, 1–4.

[67] Gao, B., W. Woo, and S. Dlay. 2010. Single-channel source separation using emd-subband variable regularized sparse features, *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4), 961–976.

[68] Genest, Christian, Kilani Ghoudi, and L-P Rivest. 1995. A semiparametric estimation procedure of dependence parameters in multivariate families of distributions, *Biometrika*, 82(3), 543–552.

[69] Ghazdali, A., M. El Rhabi, H. Fenniri, A. Hakim, and A. Keziou. 2017. Blind noisy mixture separation for independent/dependent sources through a regularized criterion on copulas, *Signal Processing*, 131, 502–513.

[70] Ghazdali, A., A. Hakim, A. Laghrib, N. Mamouni, and S. Raghay. 2015. A new method for the extraction of fetal ecg from the dependent abdominal signals using blind source separation and adaptive noise cancellation techniques, *Theoretical Biology and Medical Modelling*, 12(1), 25.

[71] Gijbels, Iène and Jan Mielniczuk. 1990. Estimating the density of a copula function, *Communications in Statistics-Theory and Methods*, 19(2), 445–464.

[72] Grewenig, Sven, Sebastian Zimmer, and Joachim Weickert. 2011. Rotationally invariant similarity measures for nonlocal image denoising, *Journal of Visual Communication and Image Representation*, 22(2), 117–130.

[73] Gribonval, Rémi and Sylvain Lesage. 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*, d-side publi., 323–330.

[74] Gumbel, E. 1960. Bivariate exponential distributions, *J. Amer. Statist. Assoc.*, 55, 698–707.

[75] Guntuboyina, Adityanand. 2011. Lower bounds for the minimax risk using f-divergences, and applications, *IEEE Transactions on Information Theory*, 57(4), 2386–2399.

[76] Gupta, Vikas, Rajesh Mahle, and Raviprakash S Shriwas. 2013. Image denoising using wavelet transform method, in *2013 Tenth International Conference on Wireless and Optical Communications Networks (WOCN)*, IEEE, 1–4.

[77] Han, Xian-Hua, Yen-Wei Chen, and Zensho Nakao. 2003. An ica-based method for poisson noise reduction, in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, Springer, 1449–1454.

[78] Herault, Jeanny and Christian Jutten. 1986. Space or time adaptive signal processing by neural network models, in *AIP conference proceedings*, American Institute of Physics, vol. 151, 206–211.

[79] Hyvarinen, A. 1999. Fast and robust fixed-point algorithms for independent component analysis, *IEEE transactions on Neural Networks*, 10(3), 626–634.

[80] Hyvärinen, A., P. Hoyer, and M. Inki. 2001. Topographic independent component analysis, *Neural computation*, 13(7), 1527–1558.

[81] Hyvärinen, A. and J. Hurri. 2004. Blind separation of sources that have spatiotemporal variance dependencies, *Signal processing*, 84(2), 247–254.

[82] Hyvärinen, A., J. Karhunen, and E. Oja. 2004. Independent component analysis, vol. 46. hoboken.

[83] Hyvärinen, A. and E. Oja. 1997. A fast fixed-point algorithm for independent component analysis, *Neural computation*, 9(7), 1483–1492.

[84] Hyvärinen, A. and E. Oja. 2000. Independent component analysis: algorithms and applications, *Neural networks*, 13(4-5), 411–430.

[85] Hyvärinen, Aapo, Juha Karhunen, and Erkki Oja. 2001. Independent component analysis, adaptive and learning systems for signal processing, communications, and control, *John Wiley & Sons, Inc*, 1, 11–14.

[86] Hyvärinen, Aapo and Petteri Pajunen. 1999. Nonlinear independent component analysis: Existence and uniqueness results, *Neural networks*, 12(3), 429–439.

[87] Jiang, Hui-Yan, Zhen-Yu Cheng, Yan Huo, Xiao-Jie Zhou, and Tian-You Chai. 2007. Research on image denoising methods based on wavelet transform and rolling-ball algorithm, in *2007 International Conference on Wavelet Analysis and Pattern Recognition*, IEEE, vol. 4, 1604–1607.

[88] Joe, H. 1997. *Multivariate models and dependence concepts*, vol. 73 of *Monographs on Statistics and Applied Probability*, Chapman & Hall, London.

[89] Joe, Harry. 1996. Families of m-variate distributions with given margins and m (m-1)/2 bivariate dependence parameters, *Lecture Notes-Monograph Series*, 120–141.

[90] Joe, Harry. 2005. Asymptotic efficiency of the two-stage estimation method for copula-based models, *Journal of multivariate Analysis*, 94(2), 401–419.

[91] Jourjine, Alexander, Scott Rickard, and Ozgur Yilmaz. 2000. Blind separation of disjoint orthogonal signals: Demixing n sources from 2 mixtures, in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, IEEE, vol. 5, 2985–2988.

[92] Jutten, Christian. 1987. *Calcul neuromimétique et traitement du signal: analyse en composantes indépendantes*, Ph.D. thesis, Grenoble INPG.

[93] Jutten, Christian and Pierre Comon. 2007. Séparation de sources 2. au-delà de l'aveugle et applications.

[94] Jutten, Christian and Juha Karhunen. 2003. Advances in nonlinear blind source separation, in *Proc. of the 4th Int. Symp. on Independent Component Analysis and Blind Signal Separation (ICA2003)*, 245–256.

[95] Karoui, M, Y. Deville, S. Hosseini, and A. Ouamri. 2012. Blind spatial unmixing of multispectral images: New methods combining sparse component analysis, clustering and non-negativity constraints, *Pattern Recognition*, 45(12), 4263–4278.

[96] Keziou, A., H. Fenniri, A. Ghazdali, and E. Moreau. 2014. New blind source separation method of independent/dependent sources, *Signal Processing*, 104, 319–324.

[97] Kim, Gunky, Mervyn J Silvapulle, and Paramsothy Silvapulle. 2007. Comparison of semiparametric and parametric methods for estimating copulas, *Computational Statistics & Data Analysis*, 51(6), 2836–2850.

[98] Knockaert, Luc. 1993. A class of statistical and spectral distance measures based on bose-einstein statistics, *IEEE transactions on signal processing*, 41(11), 3171–3174.

[99] Knockaert, Luc. 1994. Statistical thermodynamics and natural f-divergences, *Submitted to IEEE Trans. Information Theory*.

[100] Kompass, Raul. 2007. A generalized divergence measure for nonnegative matrix factorization, *Neural computation*, 19(3), 780–791.

[101] Laurberg, Hans, Mads Græsbøll Christensen, Mark D Plumbley, Lars Kai Hansen, and Søren Holdt Jensen. 2008. Theorems on positive data: On the uniqueness of nmf, *Computational intelligence and neuroscience*, 2008.

[102] Lavanya, P Venkata, C Venkata Narasimhulu, and K Satya Prasad. 2017. Transformations analysis for image denoising using complex wavelet transform, in *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, IEEE, 1–7.

[103] Le Besnerais, Guy, J-F Bercher, and Guy Demoment. 1999. A new look at entropy for solving linear inverse problems, *IEEE Transactions on Information Theory*, 45(5), 1565–1578.

[104] Lee, Jae Sung, Daniel D Lee, Seungjin Choi, Kwang Suk Park, and Dong Soo Lee. 2001. Non-negative matrix factorization of dynamic images in nuclear medicine, in *2001 IEEE Nuclear Science Symposium Conference Record (Cat. No. 01CH37310)*, IEEE, vol. 4, 2027–2030.

[105] Li, R., H. Li, and F. Wang. 2010. Dependent component analysis: Concepts and main algorithms., *JCP*, 5(4), 589–597.

[106] Li, Yuanqing, Shun-Ichi Amari, Andrzej Cichocki, Daniel WC Ho, and Shengli Xie. 2006. Underdetermined blind source separation based on sparse representation, *IEEE Transactions on signal processing*, 54(2), 423–437.

[107] Liese, F. and I. Vajda. 1987. Convex statistical distances, *Teubner-Texte zur Mathematik Teubner Texts in Mathematics*, 95, 1–85.

[108] Lin, Chih-Jen. 2007. Projected gradient methods for nonnegative matrix factorization, *Neural computation*, 19(10), 2756–2779.

[109] Lin, Jianhua. 1991. Divergence measures based on the shannon entropy, *IEEE Transactions on Information theory*, 37(1), 145–151.

[110] Lysetskiy, Mykola, Andrzej Lozowski, and Jacek M Zurada. 2002. Temporal-to-spatial dynamic mapping, flexible recognition, and temporal correlations in an olfactory cortex model, *Biological cybernetics*, 87(1), 58–67.

[111] Ma, J. and Z. Sun. 2007. Copula component analysis, in *International Conference on Independent Component Analysis and Signal Separation*, Springer, 73–80.

[112] Mairal, Julien, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. 2009. Non-local sparse models for image restoration, in *2009 IEEE 12th international conference on computer vision*, IEEE, 2272–2279.

[113] Makino, Shoji, Te-Won Lee, and Hiroshi Sawada. 2007. *Blind speech separation*, vol. 615, Springer.

[114] Mallat, Stephane G. 1989. A theory for multiresolution signal decomposition: the wavelet representation, *IEEE transactions on pattern analysis and machine intelligence*, 11(7), 674–693.

[115] Matsuyama, Yasuo. 2003. The alpha-em algorithm: surrogate likelihood maximization using alpha-logarithmic information measures, *IEEE Transactions on Information Theory*, 49(3), 692–706.

[116] Matsuyama, Yasuo, Naoto Katsumata, and Shuichiro Imahara. 2003. Convex divergence as a surrogate function for independence: The f-divergence ica, in *Proceedings of 4th International Workshop on ICA and BSS, Nara*, Citeseer, 173–178.

[117] Miller, E. and J. Fisher III. 2003. Independent components analysis by direct entropy minimization, *Computer Science*.

[118] Morgenstern, D. 1956. Einfache Beispiele zweidimensionaler Verteilungen, *Mitteilungsbl. Math. Statist.*, 8, 234–235.

[119] Moussaoui, Saïd, David Brie, and Jérôme Idier. 2005. Non-negative source separation: range of admissible solutions and conditions for the uniqueness of the solution, in *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, IEEE, vol. 5, v–289.

[120] Nelsen, R. 2006. *An introduction to copulas*, Springer Series in Statistics, Springer, New York, second edn.

[121] Nguyen, XuanLong, Martin J Wainwright, Michael I Jordan, et al. 2009. On surrogate loss functions and f-divergences, *The Annals of Statistics*, 37(2), 876–904.

[122] Nock, Richard and Frank Nielsen. 2008. Bregman divergences and surrogates for learning, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11), 2048–2059.

[123] Nordhausen, K. and H. Oja. 2018. Independent component analysis: A statistical perspective, *Wiley Interdisciplinary Reviews: Computational Statistics*, 10(5), e1440.

[124] Oakes, David. 1986. Semiparametric inference in a model for association in bivanate survival data, *Biometrika*, 73(2), 353–361.

[125] Omelka, M., I. Gijbels, and N. Veraverbeke. 2009. Improved kernel estimation of copulas: weak convergence and goodness-of-fit testing, *Ann. Statist.*, 37(5B), 3023–3058.

[126] Paatero, Pentti and Unto Tapper. 1994. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values, *Environmetrics*, 5(2), 111–126.

[127] Patil, Priyanka D and Anil D Kumbhar. 2015. Bilateral filter for image denoising, in *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, IEEE, 299–302.

[128] Pham, D. 2002. Mutual information approach to blind separation of stationary sources, *IEEE Trans. on Information Theory*, 48(7), 1935–1946.

[129] Puigt, M. and Y. Deville. 2005. Time–frequency ratio-based blind separation methods for attenuated and time-delayed sources, *Mechanical Systems and Signal Processing*, 19(6), 1348–1379.

[130] Puigt, Mathieu. 2007. *Méthodes de séparation aveugle de sources fondées sur des transformées temps-fréquence. Application à des signaux de parole.*, Ph.D. thesis, Université Paul Sabatier-Toulouse III.

[131] Ravikumar, Pradeep, Alekh Agarwal, and Martin J Wainwright. 2010. Message-passing for graph-structured linear programs: Proximal methods and rounding schemes., *Journal of Machine Learning Research*, 11(3).

[132] Renuka, Simi Venuji and Damodar Reddy Edla. 2019. Adaptive shrinkage on dual-tree complex wavelet transform for denoising real-time mr images, *Biocybernetics and Biomedical Engineering*, 39(1), 133–147.

[133] Roll, Jean-Pierre. 1981. *Contribution à la Proprioception Musculaire, à la Perception et au Contrôle du Mouvement Chez l'Homme*, Ph.D. thesis, Éditeur inconnu.

[134] Rudin, Leonid I, Stanley Osher, and Emad Fatemi. 1992. Nonlinear total variation based noise removal algorithms, *Physica D: nonlinear phenomena*, 60(1-4), 259–268.

[135] Sancetta, Alessio and Stephen Satchell. 2004. The bernstein copula and its applications to modeling and approximations of multivariate distributions, *Econometric theory*, 535–562.

[136] Scaillet, Olivier and Jean-David Fermanian. 2002. Nonparametric estimation of copulas for time series, *FAME Research paper*, (57).

[137] Schwarz, G. 1978. Estimating the dimension of a model, *The annals of statistics*, 6(2), 461–464.

[138] Silverman, B. 1986. *Density estimation for statistics and data analysis*, Monographs on Statistics and Applied Probability, Chapman & Hall, London.

[139] Sklar, M. 1959. Fonctions de répartition à n dimensions et leurs marges, *Publ. Inst. Statist. Univ. Paris*, 8, 229–231.

[140] Taleb, Anisse and Christian Jutten. 1999. Source separation in post-nonlinear mixtures, *IEEE Transactions on signal Processing*, 47(10), 2807–2820.

[141] Tanaka, T. and A. Cichocki. 2004. Subband decomposition independent component analysis and new performance criteria, in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, IEEE, vol. 5, V–541.

[142] Taskar, Ben, Simon Lacoste-Julien, Michael I Jordan, Kristin P Bennett, and Emilio Parrado-Hernández. 2006. Structured prediction, dual extragradient and bregman projections., *Journal of Machine Learning Research*, 7(7).

[143] Teboulle, Marc. 2007. A unified continuous optimization framework for center-based clustering methods., *Journal of Machine Learning Research*, 8(1).

[144] Teboulle, Marc, Pavel Berkhin, I Dhillon, Yuqiang Guan, and Jacob Kogan. 2006. Clustering with entropy-like k-means algorithms, in *Grouping multidimensional data*, Springer, 127–160.

[145] Tengtrairat, N., B. Gao, W. Woo, and S. Dlay. 2013. Single-channel blind separation using pseudo-stereo mixture and complex 2-d histogram, *IEEE Transactions on Neural Networks and Learning Systems*, 24(11), 1722–1735.

[146] Tengtrairat, N., W. Woo, S. Dlay, and B. Gao. 2015. Online noisy single-channel source separation using adaptive spectrum amplitude estimator and masking, *IEEE Transactions on Signal Processing*, 64(7), 1881–1895.

[147] Thaipanich, Tanaphol, Byung Tae Oh, Ping-Hao Wu, Daru Xu, and C-C Jay Kuo. 2010. Improved image denoising with adaptive nonlocal means (anl-means) algorithm, *IEEE Transactions on Consumer Electronics*, 56(4), 2623–2630.

[148] Tikhonov, Andrey N and Vasiliy Y Arsenin. 1977. Solutions of ill-posed problems. vh winston & sons.

[149] Tomasi, Carlo and Roberto Manduchi. 1998. Bilateral filtering for gray and color images, in *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, IEEE, 839–846.

[150] Tonazzini, Anna, Luigi Bedini, and Emanuele Salerno. 2004. Independent component analysis for document restoration, *Document Analysis and Recognition*, 7(1), 17–27.

[151] Tsukahara, H. 2005. Semiparametric estimation in copula models, *Canad. J. Statist.*, 33(3), 357–375.

[152] Vajda, Igor. 2009. On metric divergences of probability measures, *Kybernetika*, 45(6), 885–900.

[153] Villmann, T. and S. Haase. 2010. Divergence based vector quantization using frechet derivatives, *Neural Comput.*.

[154] Vyas, Aparna and Joonki Paik. 2018. Applications of multiscale transforms to image denoising: Survey, in *2018 International Conference on Electronics, Information, and Communication (ICEIC)*, IEEE, 1–3.

[155] Wand, Matt P and M Chris Jones. 1994. *Kernel smoothing*, CRC press.

[156] Wang, Xin. 2006. Moving window-based double haar wavelet transform for image processing, *IEEE Transactions on image processing*, 15(9), 2771–2779.

[157] Wu, Lei, Rong Jin, Steven CH Hoi, Jianke Zhu, and Nenghai Yu. 2009. Learning bregman distance functions and its application for semi-supervised clustering, Neural Information Processing Systems.

[158] Yang, Jianchao, John Wright, Thomas S Huang, and Yi Ma. 2010. Image super-resolution via sparse representation, *IEEE transactions on image processing*, 19(11), 2861–2873.

[159] Yu, K., W. Woo, and S. Dlay. 2015. Variational regularized two-dimensional nonnegative matrix factorization with the flexible ß-divergence for single channel source separation.

[160] Zayyani, H., M. Babaie-Zadeh, F. Haddadi, and C. Jutten. 2008. On the cramer-rao bound for estimating the mixing matrix in noisy sparse component analysis, *IEEE Signal Processing Letters*, 15, 609–612.

[161] Zayyani, H., M. Babaie-Zadeh, and C. Jutten. 2009. An iterative bayesian algorithm for sparse component analysis in presence of noise, *IEEE Transactions on Signal Processing*, 57(11), 4378–4390.

[162] Zdunek, Rafal and Andrzej Cichocki. 2006. Non-negative matrix factorization with quasi-newton optimization, in *International conference on artificial intelligence and soft computing*, Springer, 870–879.

[163] Zdunek, Rafal and Andrzej Cichocki. 2008. Fast nonnegative matrix factorization algorithms using projected gradient approaches for large-scale problems, *Computational intelligence and neuroscience*, 2008.

[164] Zhang, K. and L. Chan. 2006. An adaptive method for subband decomposition ica, *Neural computation*, 18(1), 191–223.

[165] Zhang, Kun and Aapo Hyvarinen. 2012. On the identifiability of the post-nonlinear causal model, *arXiv preprint arXiv:1205.2599*.

[166] Zhang, Lei, Weisheng Dong, David Zhang, and Guangming Shi. 2010. Two-stage image denoising by principal component analysis with local pixel grouping, *Pattern recognition*, 43(4), 1531–1549.

[167] Zhang, Yan and Xiaobin Huang. 2013. Denoising method based on independent component analysis and its application to optical imaging of functional brain, in *2013 IEEE Third International Conference on Information Science and Technology (ICIST)*, IEEE, 6–8.

[168] Ziehe, Andreas, Motoaki Kawanabe, Stefan Harmeling, and Klaus-Robert Müller. 2003. Blind separation of post-nonlinear mixtures using linearizing transformations and temporal decorrelation, *The Journal of Machine Learning Research*, 4, 1319–1338.