UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Juan Manuel Espinoza Bullón

Analysis of an adversarial approach to blind source separation
Análise de uma abordagem adversária para separação cega de fontes

Campinas

2021

Juan Manuel Espinoza Bullón

Analysis of an adversarial approach to blind source separation

Análise de uma abordagem adversária para separação cega de fontes

Dissertation presented to the School of Electrical and Computer Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Electrical Engineering, in the area of Computer Engineering.

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica, na área de Engenharia de Computação.

Supervisor: Prof. Dr. Levy Boccato

Co-supervisor Prof. Dr. Romis Attux

Este trabalho corresponde à versão final da dissertação defendida pelo aluno Juan Manuel Espinoza Bullón, e orientada pelo Prof. Dr. Levy Boccato.

Campinas

2021

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

Espinoza Bullón, Juan Manuel, 1988-
Es65a Analysis of an adversarial approach to blind source separation / Juan Manuel Espinoza Bullón. – Campinas, SP : [s.n.], 2021.

Orientador: Levy Boccato.
Coorientador: Romis Ribeiro de Faissol Attux.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Separação cega de fontes. 2. Análise de componentes independentes. 3. Aprendizado de máquina. I. Boccato, Levy, 1986-. II. Attux, Romis Ribeiro de Faissol, 1978-. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Análise de uma abordagem adversária para separação cega de fontes
Palavras-chave em inglês:
Blind source separation
Independent component analysis
Machine Learning
Área de concentração: Engenharia de Computação
Titulação: Mestre em Engenharia Elétrica
Banca examinadora:
Levy Boccato [Orientador]
Leonardo Tomazeli Duarte
Ricardo Suyama
Data de defesa: 29-07-2021
Programa de Pós-Graduação: Engenharia Elétrica

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-3249-918X
- Currículo Lattes do autor: http://lattes.cnpq.br/7960356957881165

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Juan Manuel Espinoza Bullón RA: 228562
Data de defesa: 29 de Julho de 2021
Título da Tese: “Analysis of an adversarial approach to blind source separation”

Prof. Dr. Levy Boccato (Presidente)
Prof. Dr. Leonardo Tomazeli Duarte
Prof. Dr. Ricardo Suyama

A Ata de Defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

I dedicate this dissertation to the memory of Carmela Arana Bermudez and Augusto Víctor Bullón Suarez, my grandparents.

ACKNOWLEDGEMENTS

First of all, I would like to thank my mother, Carmela Verónica, whose love, determination and perseverance during our most difficult times opened up a horizon of opportunities previously unimaginable for my brother and I. My gratitude also goes to my aunts, uncles, cousins, my brother Pedro, step dad James and every member of my family for their love and solidarity in prosperity and adversity. I would also like to thank my late grandmother, Carmela, who always taught us, with tenderness and good humour, the utmost importance of having a tight family to rely on, to work with honesty, and inspired us to pursue education as an act of defiance to face the world. And my late grandfather, Augusto, for his silent and constant companionship in my early childhood, even though he never experienced one for himself. They both will fondly live in the memory of their children and grandchildren for many decades to come.

This journey would not have been possible without the kind and patient guidance of professor Levy and professor Romis. Their generous words of wisdom and encouragement brought back the motivation and assurance I needed to not give up and persist during the on-going pandemic. I take example from them not only to be a better student, but to find joy in whatever it is I am doing, and to be a more empathetic person to others. I would also like to thank professor Tomazeli and professor Suyama for their important suggestions and the time they generously spent in reviewing this dissertation. They are also a good example of what a good researcher should be.

An important part of this wonderful experience is due to the good friends I found day by day. Although my Portuguese is far from perfect yet, every friend I found in the DSPCOM always tried to make me feel comfortable and at home, be it during classes or drinking just a couple of sips of coffee while chatting in the lab. I also want to express my gratitude to my friends from “La casa del ritmo” for their honest friendship during the years we lived together and afterwards, for the parties, the nights playing videogames, the long conversations and jokes, and, especially, for taking care of me when my health did not even let me to stand up by myself and making me a part of their lives. I did not drive -completely- crazy during these years thanks to them.

Finally, I want to thank every member of the faculty, and the university as a whole, for their constant assistance whenever a problem appeared or an opportunity to help us, the students, was available. Also, my gratitude goes to the CNPq for providing me with the opportunity to fully dedicate myself to this research.

This study was financed in part by CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil. Number 133777/2019-7.

"The sciences, each straining in its own direction, have hitherto harmed us little; but some day the piecing together of dissociated knowledge will open up such terrifying vistas of reality, and of our frightful position therein, that we shall either go mad from the revelation or flee from the light into the peace and safety of a new dark age."

(H. P. Lovecraft - The Call of Cthulhu)

ABSTRACT

The problem of blind source separation (BSS) involves the challenge of retrieving a set of unknown signals, called sources, given observations of mixtures of such signals, being the mixing system also unknown. A key hypothesis explored in the literature is the statistical independence of the sources, which forms the basis of the solid approach known as independent component analysis (ICA).

In this work, we analyze a recent technique proposed in (BRAKEL; BENGIO, 2017) to solve the BSS problem under the same assumptions of ICA, which is based on the notion of adversarial learning. The method, named ANICA, employs an autoencoder, whose main task is to codify the mixtures into meaningful latent variables at the code layer, accompanied with a discriminator model, which is responsible for recognizing whether the distribution of the latent vector resembles the product of the individual distributions. By establishing an adversarial learning scheme between the autoencoder and the discriminator, the former is encouraged to generate independent variables at the internal code, which represent the estimates of the sources.

Motivated by the promising initial results reported in (BRAKEL; BENGIO, 2017), here we establish the conditions for convergence of ANICA and we propose a blind criterion to select the best training epoch. Additionally, considering the well-established JADE and FastICA algorithms as benchmarks, we analyze the performance of ANICA in different scenarios: by varying the number of samples and sources, the noise variance, by separating overdetermined mixtures, or considering different types of sources. The obtained results indicate that, albeit requiring a significantly higher number of samples, ANICA can successfully recover the sources and, in some cases, achieve an error smaller than its competitors, and suggest that ANICA has a significant potential of extension to more general scenarios.

Keywords: Blind Source Separation, Independent Component Analysis, Machine Learning, Autoencoders, Generative Adversarial Networks, ANICA.

RESUMO

O problema de Separação Cega de Fontes (BSS, do inglês *blind source separation*) envolve o desafio de recuperar um conjunto de sinais desconhecidos, denominados fontes, dadas as observações de misturas desses sinais, sendo o sistema de mistura também desconhecido. Uma hipótese chave explorada na literatura é a independência estatística das fontes, que define o fundamento da abordagem sólida conhecida como análise de componentes independentes (ICA, do inglês *independent component analysis*).

Neste trabalho, analisamos uma técnica recente proposta em (BRAKEL; BENGIO, 2017) para resolver o problema do BSS sob os mesmos pressupostos de ICA, que se baseia na noção de aprendizado adversário. O método, denominado ANICA, emprega uma rede autocodificadora (*autoencoder*), cuja principal tarefa é codificar as misturas em variáveis latentes significativas na camada de código, acompanhadas de um modelo discriminador, que é responsável por reconhecer se a distribuição do vetor latente se assemelha ao produto das distribuições individuais. Ao estabelecer um esquema de aprendizagem adversário entre o *autoencoder* e o discriminador, o primeiro é estimulado a gerar variáveis independentes no código interno, que representam as estimativas das fontes.

Motivados pelos resultados iniciais promissores relatados em (BRAKEL; BENGIO, 2017), aqui estabelecemos as condições para convergência do ANICA e propomos um critério cego para selecionar a melhor época de treinamento. Adicionalmente, considerando os renomados algoritmos JADE e FastICA bem estabelecidos como *benchmarks*, analisamos o desempenho do ANICA em diferentes cenários: variando o número de amostras e fontes, a variância do ruído, separando misturas sobredeterminadas, ou considerando diferentes tipos de fontes. Os resultados obtidos indicam que, embora necessite de um número significativamente maior de amostras, o método ANICA pode recuperar as fontes com sucesso e, em alguns casos, obter um erro menor que seus concorrentes, e sugerem que o ANICA tem um potencial significativo de extensão para cenários mais gerais.

Palavras-chaves: Separação Cega de Fontes, Análise de Componentes Independentes, Aprendizado de Máquina, *Autoencoders*, Redes Adversárias Generativas, ANICA.

LIST OF FIGURES

Figure 2.1 – Blind Source Separation as a generative model and a problem. 21
Figure 2.2 – Rank of the mixing system and the underdetermined case. 23
Figure 2.3 – Preprocessing the data: Whitening is a rotation away from independence. 30
Figure 2.4 – The problem of gaussianity. 32
Figure 2.5 – Geometry of sparsity in the time domain. 34
Figure 3.1 – Non-gaussianity maximization leads to independence. 41
Figure 4.1 – Structure of a MLP with 3 inputs, 2 hidden layers and a two-component output layer. 50
Figure 4.2 – General structure of an Autoencoder and its most important components. 52
Figure 4.3 – Connection between undercomplete LAEs and PCA. 54
Figure 4.4 – Structure and components of a Variational Autoencoder. 56
Figure 4.5 – Structure and components of a Generative Adversarial Network. 58
Figure 4.6 – DCGAN on MNIST. 59
Figure 4.7 – Structure and components of an Adversarial Autoencoder. 61
Figure 4.8 – AAE on the MNIST dataset. 62
Figure 5.1 – Structure and components of ANICA. 64
Figure 5.2 – Resampling procedure and its effect on statistical dependence. 66
Figure 6.1 – Training curves and convergence of ANICA. 73
Figure 6.2 – Proposed blind metrics to evaluate independence. 74
Figure 6.3 – Retrieval of independent estimates from whitened mixtures by ANICA. 74
Figure 6.4 – Matching of the PDFs of the data performed by ANICA. 75
Figure 6.5 – Experiment 1: Impact of the number of samples on the attained independence. 76
Figure 6.6 – Experiment 1: Impact of the number of samples on convergence. 77
Figure 6.7 – Experiment 2: Impact of the number of components on the attained independence. 78
Figure 6.8 – Experiment 3: Impact of the lack of decoder on the attained independence. 79
Figure 6.9 – Experiment 4: Impact of the SNR_{dB} on the attained independence. . . 80
Figure 6.10–Experiment 4: Impact of the SNR_{dB} on convergence 80
Figure 6.11–Experiment 5.1: All convergence curves. 81
Figure 6.12–Experiment 5.1: Impact of the number of mixtures on the attained independence considering the overdetermined scenario. 82
Figure 6.13–Experiment 5.2: Impact of the undercomplete autoencoder on the attained independence. 83
Figure 6.14–Experiment 6: Impact of the Gaussianity of the sources on ANICA. . . 84
Figure 6.15–Experiment 6: Best correspondence among signals. 85

Figure 6.16–Experiment 7: Audio signals separation with ANICA. 86
Figure 6.17–Experiment 7: Best correspondence among audio signals. 87

LIST OF ABBREVIATIONS AND ACRONYMS

AAE Adversarial Autoencoder.

AE Autoencoder.

AMUSE Algorithm for Multiple Unknown Signals Extraction.

ANICA Adversarial Non-linear Independent Component Analysis.

ANN Artificial Neural Network.

BSS Blind Source Separation.

CLT Central Limit Theorem.

CNN Convolutional Neural Network.

DAE Denoising Autoencoder.

DCGAN Deep Convolutional GAN.

GAN Generative Adversarial Network.

HOS Higher-Order Statistics.

ICA Independent Component Analysis.

JADE Joint Approximated Diagonalization of Eigenmatrices.

LAE Linear Autoencoder.

MAP Maximum A Posteriori.

MLP Multi-Layer Perceptron.

MN Module of the negentropy.

MNK Module of the normalized kurtosis.

MSE Mean Squared Error.

NAE Normalized Amari error.

PCA Principal Component Analysis.

PDF Probability Density Function.

PMF Probability Mass Function.

PNL Post-Nonlinear.

RNN Recurrent Neural Network.

SAE Sparse Autoencoder.

SCA Sparse Component Analysis.

SMI Shifted mutual information.

SNR Signal-To-Noise Ratio.

SOBI Second Order Blind Indentification.

SOS Second Order Statistic.

SSS Strict Sense Stationarity.

SVD Singular Value Decomposition.

TDSEP Temporal Decorrelation Source Separation.

VAE Variational Autoencoder.

WGN White Gaussian Noise.

WSS Wide Sense Stationarity.

CONTENTS

1 INTRODUCTION . . . 16

2 BLIND SOURCE SEPARATION . . . 18
2.1 On Signals and the property of stationarity . . . 18
2.2 BSS Latent model and problem . . . 20
2.3 Modelling the problem . . . 21
2.3.1 (Noiseless) Linear and Instantaneous Model . . . 22
2.4 Principal Component Analysis and Whitening . . . 24
2.4.1 On uncorrelatedness . . . 24
2.4.2 Principal Component Analysis . . . 25
2.4.3 Whitening . . . 26
2.5 Independent Component Analysis . . . 27
2.5.1 PCA and Whitening as preprocessing for ICA . . . 28
2.5.2 On non-gaussianity . . . 29
2.6 Other approaches . . . 31
2.6.1 Second Order Statistics . . . 31
2.6.2 Sparse Component Analysis . . . 33
2.6.3 Bayesian Approach . . . 35
2.7 Practical Applications . . . 36
2.8 Summary . . . 38

3 ICA: CRITERIA AND ALGORITHMS . . . 39
3.1 Mutual Information . . . 39
3.1.1 The Information maximization principle . . . 40
3.2 Non-Gaussianity . . . 40
3.2.1 Kurtosis . . . 41
3.2.2 Negentropy . . . 42
3.2.3 FastICA . . . 43
3.3 Higher-Order Statistics: Cumulants . . . 44
3.3.1 Joint Approximated Diagonalization of Eigenmatrices . . . 45
3.4 Summary . . . 47

4 AUTOENCODERS AND ADVERSARIAL NETWORKS . . . 48
4.1 Artificial Neural Networks . . . 48
4.1.1 Multi-Layer Perceptrons . . . 49
4.2 Autoencoders . . . 51

4.2.1 Linear Autoencoders and PCA . . . 53
4.2.2 Denoising Autoencoders . . . 55
4.2.3 Sparse Autoencoders . . . 55
4.2.4 Variational Autoencoders . . . 56
4.3 Generative Adversarial Networks . . . 57
4.4 Adversarial Autoencoders . . . 60
4.5 Summary . . . 63

5 AN ADVERSARIAL APPROACH TOWARDS ICA . . . 64
5.1 ANICA: Adversarial Non-linear ICA . . . 64
5.1.1 Resampling Procedure . . . 65
5.2 General algorithm and convergence analysis . . . 66
5.2.1 Convergence Analysis . . . 68
5.3 ANICA and its limits for ICA . . . 69
5.3.1 Underdeterminedness and ANICA . . . 69
5.3.2 Post-Nonlinear ICA . . . 70
5.4 Summary . . . 70

6 EXPERIMENTS . . . 71
6.1 Metrics and methodology . . . 71
6.2 Initial Experiment: Metrics and Convergence . . . 72
6.3 Experiment 1: Number of samples . . . 75
6.4 Experiment 2: Number of components . . . 76
6.5 Experiment 3: ANICA without the decoder . . . 77
6.6 Experiment 4: Signal-to-Noise Ratio . . . 78
6.7 Experiment 5: Overdetermined Case . . . 81
6.7.1 Experiment 5.1: Dimensionality reduction by PCA . . . 81
6.7.2 Experiment 5.2: Undercomplete Autoencoder . . . 82
6.8 Experiment 6: Gaussianity . . . 82
6.9 Experiment 7: Real Audio Signals . . . 83
6.10 Summary . . . 84

7 CONCLUSIONS . . . 88

BIBLIOGRAPHY . . . 90

1 INTRODUCTION

Suppose you are listening to a piece of music when some sequence of sounds, produced by an unknown instrument, catches your attention. At first, you try to mentally mute everything else and focus on that particular sound source among the mixtures of sounds presented to you by the set of speakers. Unfortunately, everything is so intertwined that you are initially unable to follow this sequence any longer. However, by paying attention to the timbre of the sounds, while listening to the same piece for a number of times, you not only recognize the sequence of interest, but all the others being played at the moment. In this case, you may consider you have learnt to mentally separate the mixture of sounds into an approximation of each of the sequences played by the instruments, even though you did not know them beforehand nor how they were actually combined.

Following the example, we may assume that every sequence of sounds produced by a different instrument shares a distinctive timbre and execution technique. These sequences, or sources, are somehow mixed and modified within a recording studio, or mixing system, in order to output a different mixture signal through each of the speakers, where each mixture is a different composition of the sources.

In the field of signal processing, this scenario is formally defined and addressed by the research area known as Blind Source Separation (BSS) which is more formally presented in the Chapter 2. In a nutshell, BSS interprets a set of mixture signals as a manifestation of another set of unknown source signals through an also unknown mixing system. Then, BSS poses the problem of retrieving an approximation of such sources, which are called estimates, only from the mixtures themselves and some a priori information related to the sources and mixing system (as, for example, all sources must belong to different instruments and the mixing system is noiseless). The problem is said to be blind due to the lack of direct access to the sources and to the mixing system.

A proper solution is achieved by adopting a separation criterion, such as considering that all sounds from the same instrument should share a distinctive timbre and execution, from which a separation system is designed to receive the mixtures and produce the estimates that best satisfy the chosen criterion.

Different approaches to find a proper solution for the BSS problem are based on different restrictions and assumptions regarding the sources and how the mixtures are generated. Due to its simplicity, a popular framework since the early days of BSS has been Independent Component Analysis (ICA), whose main assumption is the statistical independence of the sources. ICA is presented in Section 2.5, and exposed with more detail in Chapter 3.

Additionally, and also from the beginning of the discipline, the field of Machine Learning has offered a useful repertoire of algorithms, especially those based on neural networks, such as the autoencoder, to model adequate separation systems and their respective cost function, which should be optimized to retrieve independent estimates according to the selected separation criterion. More recently, adversarial training seems to provide a good alternative approach to the BSS problem by implicitly learning the distribution of the sources, and, in the case of ICA, imposing independence. For this reason, we dedicate Chapter 4 to the topics of Machine Learning of greater interest to BSS.

The purpose of this work is to present and deepen the understanding of a novel algorithm called Adversarial Non-linear Independent Component Analysis (ANICA), which is designed for the solution of the BSS problem under the framework of ICA. ANICA was developed by the researchers Philemon Brakel and Yoshua Bengio in (2017) and, roughly speaking, is a generative algorithm that trains an autoencoder and a discriminator in an adversarial scheme to generate a code of independent components. A key component of ANICA is its resampling procedure, which simulates samples from the independent distribution of the generated code at every training epoch and, thus, represents the target distribution of the code. Chapter 5 is dedicated to the exposition of this adversarial approach, along with an analysis of its convergence.

In Chapter 6, we evaluate the advantages and shortcomings of ANICA in a number of different experiments, evaluating the performance of ANICA in comparison to well-established algorithms such as FastICA and Joint Approximated Diagonalization of Eigenmatrices (JADE). For this purpose, we have employed blind metrics, based on Mutual Information, Kurtosis and Negentropy, as well as the non-blind Normalized Amari error (NAE) to confirm the results.

Finally, Chapter 7 brings the main conclusions and points some perspectives for future investigations.

2 BLIND SOURCE SEPARATION

This chapter is dedicated to presenting the problem of blind source separation, its main characteristics and requirements, as well as some of the best known approaches to solve it, among which independent component analysis stands out as one of the most relevant.

2.1 ON SIGNALS AND THE PROPERTY OF STATIONARITY

Signals are modelled as random processes due to the inherent uncertainty about their temporal evolution as well as the values they may have at each time instant. Generally, some conditions regarding their random behaviour are assumed in order to establish a tractable scenario.

Strict Sense Stationarity (SSS) imposes a fixed structure for the joint Probability Density Function (PDF) across different instants, or time-states, of a random process. Based on this property, it can be argued that it is not important when a signal is observed, due to its statistical properties being invariant to any time shift (PAPOULIS; PILLAI, 2002), but rather the frequency of these observations. Thus, stationarity can be defined as:

Definition 2.1.1 (Strict Sense Stationarity)**.** For any finite sequence of k time-states of a random process $v(t)$, stationarity implies that the joint PDF of these states must be invariant to any time shift t_s. Thus, for k different time-states:

$$\mathrm{p}_{v(t_1),\ v(t_2),\ \cdots,\ v(t_k)}(\cdot) = \mathrm{p}_{v(t_1+t_s),\ v(t_2+t_s),\ \cdots,\ v(t_k+t_s)}(\cdot), \tag{2.1}$$

where the left-hand side of (2.1) refers to the joint PDF of the sequence $v(t_1)$ to $v(t_k)$, while the right-hand side is the joint PDF after a time-shift t_s is applied to all of its states. ■

If we consider $k = 1$ in Definition 2.1.1, it follows that stationarity implies that all the marginal densities of the time-states are equal to the same PDF:

$$\mathrm{p}_{v(t_1)}(\cdot) = \mathrm{p}_{v(t_1+t_s)}(\cdot) = \mathrm{p}(\cdot), \tag{2.2}$$

where $\mathrm{p}(\cdot)$ is the common marginal PDF for any time-state.

Additionally, for any two time-states, denoted as t_1 and t_2, their joint PDF depends only on the time difference $\tau = t_2 - t_1$:

$$\mathrm{p}_{v(t_1),\ v(t_2)}(\cdot) = \mathrm{p}_{v(t_1+t_s),\ v(t_1+t_s+\tau)}(\cdot) = \mathrm{p}_{\tau}(\cdot), \tag{2.3}$$

where $\mathrm{p}_\tau(\cdot)$ is the common joint PDF for any two time-states separated by τ.

As a consequence, SSS guarantees that the mean $m_v(t)$ will remain constant for all states, while the autocorrelation $\mathrm{R}_v(\cdot)$ and autocovariance $\mathrm{C}_v(\cdot)$ functions, defined in (2.5) and (2.7) for real-valued processes, will depend only on the time difference τ:

$$m_v(t) = m_v \tag{2.4}$$
$$\mathrm{R}_v(t,\ \tau) = \mathbb{E}[v(t)v(t+\tau)] \tag{2.5}$$
$$= \mathrm{R}_v(\tau) \tag{2.6}$$
$$\mathrm{C}_v(t,\ \tau) = \mathrm{R}_v(t,\ \tau) - m_v(t)m_v(t+\tau) \tag{2.7}$$
$$= \mathrm{R}_v(\tau) - m_v^2 = \mathrm{C}_v(\tau), \tag{2.8}$$

where m_v is constant, while $\mathrm{R}_v(\tau)$, $\mathrm{C}_v(\tau)$ are only functions of τ considering SSS.

SSS is a stringent condition for random processes, whereas Wide Sense Stationarity (WSS) requires only the compliance of (2.4), (2.6) and (2.8), thus encompassing a broader set of random phenomena. Additionally, all independent and identically distributed random processes satisfy Definition 2.1.1, so they are a subset within SSS processes (KAY, 2006).

Similarly, when dealing with two stationary processes $v(t)$ and $u(t)$, they are said to be jointly wide-sense stationary if their cross-correlation and cross-covariance functions, defined in (2.9) and (2.11) for real-valued processes, are only affected by the time difference τ:

$$\mathrm{R}_{vu}(t,\ \tau) = \mathbb{E}[v(t)u(t+\tau)] \tag{2.9}$$
$$= \mathrm{R}_{vu}(\tau) \tag{2.10}$$
$$\mathrm{C}_{vu}(t,\ \tau) = \mathrm{R}_{vu}(t,\ \tau) - m_v(t)m_u(t+\tau) \tag{2.11}$$
$$= \mathrm{R}_{vu}(\tau) - m_v m_u = \mathrm{C}_{vu}(\tau), \tag{2.12}$$

where m_v, m_u are constants, while $\mathrm{R}_{vu}(\tau)$, $\mathrm{C}_{vu}(\tau)$ are only functions of τ considering jointly stationarity.

In order to estimate these statistical entities, a useful property of some SSS (or WSS) random processes is called ergodicity. An ergodic process allows obtaining its statistical measures based only on the knowledge of a single realization, having access to a sufficiently large number of time samples, instead of requiring its complete ensemble (PAPOULIS; PILLAI, 2002). However, stationarity does not guarantee ergodicity.

In the context of BSS, considering stationary signals simplifies the statistical analysis of the algorithms developed for its solution. Furthermore, when there is no time lag among signals, i.e., $\tau = 0$, the resulting cross-correlation and cross-covariance are no different to a simple correlation and covariance defined for two random variables, thus allowing the use of well-known decorrelation techniques.

Therefore, unless stated otherwise, all signals will be considered to be ergodic and stationary real-valued random processes, while the retrieved or observed values would be the realizations of such signals in a number of samples.

2.2 BSS LATENT MODEL AND PROBLEM

As previously mentioned in Chapter 1, BSS proposes a latent model to explain the generation of the mixture signals. To present this model, consider M different sources $s_i(t)$ collected in $\mathbf{s}(t) = (s_1(t),\ s_2(t),\ \cdots,\ s_M(t))^T$, interacting with each other and with the environment where they exist, thus producing another set of N mixtures $x_j(t)$, collected in $\mathbf{x}(t) = (x_1(t),\ x_2(t),\ \cdots,\ x_N(t))^T$.

The environment, or the mixing system $\mathcal{A}$, is the responsible for generating mixtures of the hidden sources. Typically, the exact composition of each mixture is unknown, so that the mixture vector $\mathbf{x}(t)$ is modelled as:

$$\mathbf{x}(t) = \mathcal{A}(\mathbf{s}(t)), \tag{2.13}$$

where $\mathbf{x}(t) \in \Re^N$, $\mathbf{s}(t) \in \Re^M$ are random vectors, while $\mathcal{A} : \Re^M \mapsto \Re^N$ is the unknown mapping describing the mixing system.

BSS is not only interested in explaining the generation of the mixtures, but, more importantly, the main challenge consists in retrieving an approximation of all the underlying sources. So, to find these approximations or estimates $y_k(t)$, collected in $\mathbf{y}(t) = (y_1(t),\ y_2(t),\ \cdots,\ y_M(t))^T$, a second transformation is needed in order to invert the effects of the mixing system. This second transformation constitutes the separation system $\mathcal{W}$, whose action is expressed as follows:

$$\mathbf{y}(t) = \mathcal{W}(\mathbf{x}(t)) = \mathcal{W}(\mathcal{A}(\mathbf{s}(t))), \tag{2.14}$$

where $\mathbf{y}(t) \in \Re^M$ is a random vector, while $\mathcal{W} : \Re^N \mapsto \Re^M$ denotes the mapping produced by the separation system.

Figure 2.1 shows the main elements involved in the BSS problem, where everything behind the mixtures, including the number M of sources, is unknown.

For the solution of the BSS problem, it is not relevant to capture the specific order and magnitude of the original sources in $\mathbf{s}(t)$. In other words, it is acceptable that the estimate vector $\mathbf{y}(t)$ be a permuted an scaled version of the source vector. Mathematically, this means that:

$$\mathbf{y}(t) = \Lambda \mathrm{P}\mathbf{s}(t), \tag{2.15}$$

where $\Lambda,\ \mathrm{P} \in \Re^{M \times M}$ correspond to a permutation matrix and a diagonal matrix that specifies the scales applied to the sources.

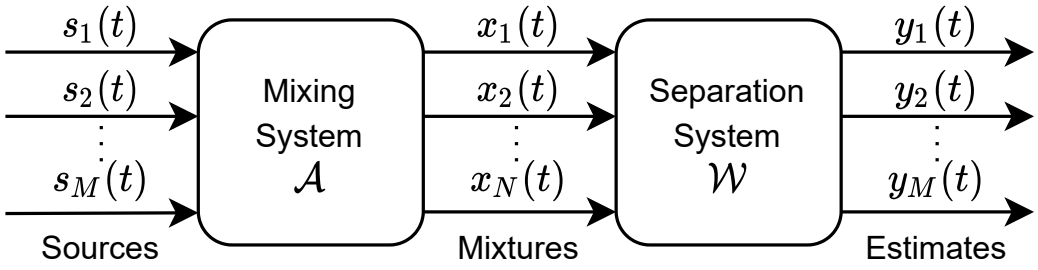


Figure 2.1 – Blind Source Separation as a generative model and a problem.

Unfortunately, there is not enough information to design a separation system capable of retrieving proper estimates of the sources from the mixtures given by the general model of (2.13), nor a universal criterion to evaluate the quality of these estimates. In this sense, the BSS problem is ill-posed in its most general formulation.

Hence, it becomes necessary to restrict the general mixing system according some properties. Additionally, some conditions regarding the character of the sources are also necessary to check whether the estimates capture a useful representation of them.

2.3 MODELLING THE PROBLEM

Although it is not possible to "open" the mixing system and see its internal mechanisms, it is possible to presume that it presents a few characteristics, such as memory and linearity, in order to attain a simpler, yet general, model with a practical appeal and a feasible solution.

MEMORY

A mixing system is said to present memory when source samples from past and present time-states are mixed together in order to produce the current mixtures, as occurs, for example, when audio signals are recorded in an environment with reverberation. A particularly relevant scenario involving memory is related to convolutive mixtures. In this case, each mixture can be seen as the result of a linear combination involving present and past samples of the available sources. In (CASTELLA; CHEVREUIL; PESQUET, 2010) a more detailed account on convolutive mixing models is offered.

On the other hand, a mixing system is instantaneous, or memoryless, when the observation of each mixture at instant t depends solely on the source samples at the same instant.

LINEARITY

A mixing system is said to be linear only if its mapping satisfies the superposition principle. Otherwise, it is a non-linear system. For example, considering the signals $\mathbf{v}(t)$ and $\mathbf{w}(t)$, the mapping $\mathcal{F}(\cdot)$ is linear if and only if:

$$\mathcal{F}(\alpha\mathbf{v}(t) + \beta\mathbf{w}(t)) = \alpha\mathcal{F}(\mathbf{v}(t)) + \beta\mathcal{F}(\mathbf{w}(t)), \tag{2.16}$$

where α and β are scalar constant values.

NUMBER OF SOURCES AND MIXTURES

Another important aspect that leads to specific scenarios of the BSS problem refers to the relation between the number of mixtures N and the number of sources M. This relation defines whether the problem is well-determined ($N = M$), overdetermined ($N > M$), or underdetermined ($N < M$). The last case is especially complicated since it generally involves two separate steps: first, the identification of the mixing system and, second, the approximation of the sources, which are both harder compared to the other scenarios.

2.3.1 (NOISELESS) LINEAR AND INSTANTANEOUS MODEL

The classical model explored in BSS arises when considering linear and instantaneous mixing and separation systems. From these characteristics, and assuming there is no significant noise, both systems are modelled as matrices and the time indices are dropped. Then, the model is presented as follows:

$$\begin{aligned} \mathbf{x} &= \mathrm{A}\mathbf{s} \\ \mathbf{y} &= \mathrm{W}\mathbf{x} = (\mathrm{WA})\mathbf{s}, \end{aligned} \tag{2.17}$$

where $\mathrm{A} \in \Re^{N\times M}$, $\mathrm{W} \in \Re^{M\times N}$ are the mixing and separation matrix, respectively.

THE PROBLEM WITH THE UNDERDETERMINED CASE

From this model, an initial argument against the underdetermined case can be offered due to the model being fully solvable when W takes either one of the forms:

$$\mathrm{W} = \Lambda\mathrm{P}\mathrm{A}^{-1} \tag{2.18}$$

$$\mathrm{W} = \Lambda\mathrm{P}\mathrm{A}^{+}, \tag{2.19}$$

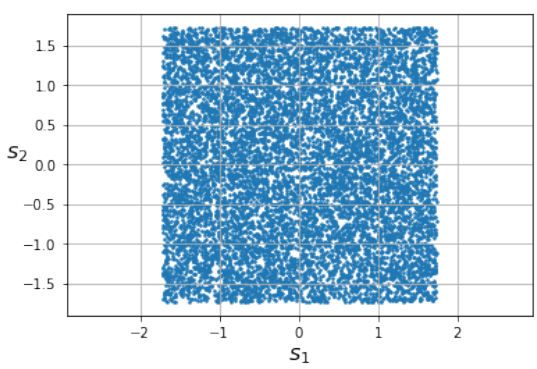
where $\mathrm{A}^{+} \in \Re^{M\times N}$ is the left pseudoinverse of A, i.e., $\mathrm{A}^{+}\mathrm{A} = \mathrm{I}$.

For (2.18) or (2.19) to be possible, A must necessary be full rank, i.e., $\mathrm{rank}(\mathrm{A}) = \min(N,\ M)$ and either be square and invertible, i.e., $\mathrm{rank}(\mathrm{A}) = N = M$, or rectangular such that $\mathrm{rank}(\mathrm{A}) = M < N$, respectively. None of these direct solutions can be formulated for the underdetermined case ($M > N$).

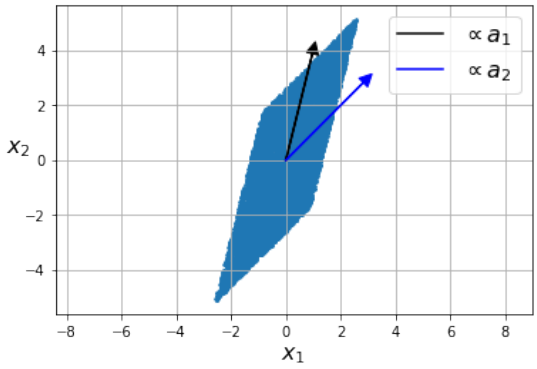
Example 2.3.1. This point is exemplified in Figure 2.2, where two different mixing matrices, A and $\tilde{\text{A}}$, are applied over two statistically independent sources (more on statistical independence on Section 2.5). The corresponding mixing matrices are:

$$\text{A} = \begin{bmatrix} 0.5 & 1 \\ 2 & 1 \end{bmatrix} \qquad \tilde{\text{A}} = \begin{bmatrix} 0.5 & 0.25 \\ 2 & 1 \end{bmatrix}, \tag{2.20}$$

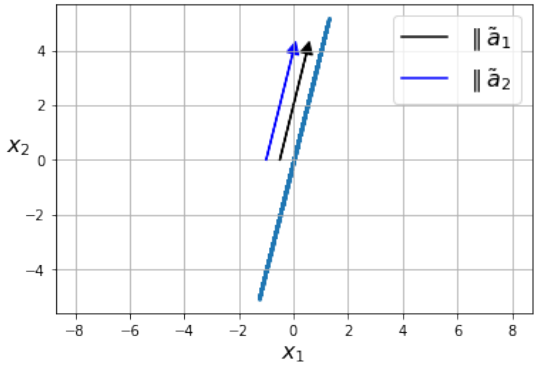
where only A is full rank.



(a) Uniform Sources **s**.



(b) Mixtures **x** produced by A.



(c) Mixtures **x** produced by $\tilde{\text{A}}$.

Figure 2.2 – Rank of the mixing system and the underdetermined case.

Figure 2.2a shows two independent zero-mean, unit-variance sources, both with uniform distributions. Then, in Figure 2.2b, these sources are mixed by A, such that the mixtures are in a space defined by its two linearly independent columns ($\mathbf{a}_1$ and $\mathbf{a}_2$). However, in Figure 2.2c, a rank-deficient matrix $\tilde{\text{A}}$ is not capable of providing enough information, as one dimension is lost, in order to build an adequate separation matrix from the mixtures. Proportional and parallel columns are displayed for convenience in the last two figures. ■

Furthermore, as pointed out in (KIM; YOO, 2004; KIM; YOO, 2009) for the underdetermined case, after identifying (or even knowing) the mixing matrix, there would be an infinity of possible solutions for the estimates of the sources unless some assumptions, such as the sources being sparse or discrete, are considered (JUTTEN; COMON, 2010).

Additionally, this result also emphasizes the need for having distinct mixtures spanned in different directions in order to deal with a full rank matrix A and, thus, avoiding the underdetermined case.

2.4 PRINCIPAL COMPONENT ANALYSIS AND WHITENING

Although Principal Component Analysis (PCA) is not considered to be a technique within the approaches used to solve the BSS problem, it still discovers something meaningful about the multivariate data at hand in the form of its principal components, which are mutually uncorrelated, but not necessarily independent. The notion of whitening is closely related to PCA and also involves finding a linear transformation that yields uncorrelated components, but with normalized variance and without reducing the dimensionality of the data. Both topics are important in the context of BSS, as will be seen in Section 2.5.1, and, thus, are described in the sequence.

2.4.1 ON UNCORRELATEDNESS

Two random variables, v_1 and v_2, are linearly dependent, or correlated, when there is some kind of joint variation between them, either both increasing (or decreasing) their values together, or, while one increases, the other one decreases. Conversely, they are said to be uncorrelated when there is no direct way to predict the growth of one from the other, which can be expressed in terms of a null joint covariance, or $\text{cov}(v_1,\ v_2) = 0$. This idea can be extended to a random vector $\mathbf{v}$ and expressed in terms of the covariance matrix $\mathrm{C}_{\mathbf{v}}$:

$$\mathrm{C}_{\mathbf{v}} = \mathbb{E}[(\mathbf{v} - \mu_{\mathbf{v}})(\mathbf{v} - \mu_{\mathbf{v}})^T], \tag{2.21}$$

where $\mu_{\mathbf{v}}$ is the mean vector.

Then, the components in $\mathbf{v}$ are said to be uncorrelated when the covariance matrix $\mathrm{C}_{\mathbf{v}}$ is diagonal.

Similarly, two random processes are said to be uncorrelated when their cross-covariance function $\mathrm{C}_{vu}(t,\ \tau)$, defined in (2.11), is always null for any time-state t and time lag τ (PAPOULIS; PILLAI, 2002). Then, for a vector $\mathbf{v}(t)$ of jointly stationary components, a cross-covariance function matrix $\mathrm{K}_{\mathbf{v}}(\tau)$ may be defined analogous to (2.21):

$$\mathrm{K}_{\mathbf{v}}(\tau) = \mathbb{E}[(\mathbf{v}(t) - \mu_{\mathbf{v}})(\mathbf{v}(t+\tau) - \mu_{\mathbf{v}})^T], \tag{2.22}$$

where $\mathrm{K}_{\mathbf{v}}(\tau)$ depends only on τ.

If $\mathbf{v}(t)$ is uncorrelated, $\mathrm{K}_{\mathbf{v}}(\tau)$ must be diagonal for all τ. Additionally, when there is no lag among components, i.e., $\tau = 0$, (2.21) and (2.22) amount to the same.

2.4.2 PRINCIPAL COMPONENT ANALYSIS

As explained in (SHLENS, 2014), the idea behind PCA is that a random vector, expressing a multivariate data set, may contain some redundant components, i.e. correlated, and others offering little or nothing of interest by remaining almost constant sample by sample. Some noise with little variance may be present, too. Then, PCA attempts to simplify this description by finding a new orthonormal basis onto which to project the data, so that it may be expressed with fewer and uncorrelated new components, such that each subsequent new component corresponds to the next direction of the basis maximizing the variance of the projection, thus filtering the noise. The components found in this orderly manner, prioritizing the variance in order to lose as little information as possible, are called the principal components of the data set.

By considering a previously centered, i.e., with zero mean, L-component random vector $\mathbf{v}$, and calculating its covariance matrix $\mathrm{C}_{\mathbf{v}}$, a solution for such orthonormal basis is found in the eigenvector matrix U of the covariance matrix $\mathrm{C}_{\mathbf{v}}$, which is usually obtained through Singular Value Decomposition (SVD), when the corresponding eigenvalue matrix $\mathbf{\Sigma}$, holding eigenvalues σ_i, is decreasingly ordered, or:

$$\begin{aligned} \mathrm{C}_{\mathbf{v}} &= \mathbb{E}[\mathbf{v}\mathbf{v}^T] && (2.23)\\ &= \mathrm{U}\Sigma\mathrm{U}^T, && (2.24) \end{aligned}$$

where $\Sigma \in \Re^{L\times L}$ is diagonal with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_R \geq \cdots \geq \sigma_L$, while $\mathrm{U} \in \Re^{L\times L}$ is the orthogonal matrix of corresponding eigenvectors $\mathbf{u}_1,\ \mathbf{u}_2,\ \cdots,\ \mathbf{u}_R,\ \cdots,\ \mathbf{u}_N$.

Thus, the R first principal components, such that $R \leq L$ and collected in the random vector $\mathbf{r}$, are computed by the linear transformation:

$$\mathbf{r} = \mathrm{U}_R^T\mathbf{v}, \tag{2.25}$$

where $\mathrm{U}_R \in \Re^{L\times R}$ holds the R first eigenvectors, from $\mathbf{u}_1$ to $\mathbf{u}_R$.

These components are mutually uncorrelated and with maximum variance, as can be inferred from the covariance matrix $\mathrm{C}_{\mathbf{r}}$:

$$\begin{aligned} \mathrm{C}_{\mathbf{r}} &= \mathrm{U}_R^T\mathrm{C}_{\mathbf{v}}\mathrm{U}_R \\ &= \mathrm{U}_R^T(\mathrm{U}\Sigma\mathrm{U}^T)\mathrm{U}_R \\ &= \Sigma_R, && (2.26) \end{aligned}$$

where $\Sigma_R \in \Re^{R\times R}$ holds the R largest eigenvalues, or $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_R$.

Furthermore, it can be proven that, after preserving most of the variance in the new orthonormal basis, PCA minimizes the Mean Squared Error (MSE) of reconstructing $\mathbf{v}$ from $\mathbf{r}$ (HYVäRINEN; KARHUNEN; OJA, 2001). Such reconstruction is accomplished

by "going back" from the new basis, or:

$$\tilde{\mathbf{v}} = \mathrm{U}_R\mathbf{r} = \mathrm{U}_R\mathrm{U}_R^T\mathbf{v}, \tag{2.27}$$

where $\tilde{\mathbf{v}} \in \Re^L$ is the reconstruction of $\mathbf{v}$.

Thus, the MSE between $\mathbf{v}$ and $\tilde{\mathbf{v}}$ is dependent on the basis and can be measured by a function $\mathrm{J}_{\mathrm{PCA}}(\cdot)$ expressed as:

$$\mathrm{J}_{\mathrm{PCA}}(\mathrm{U}_R) = \mathbb{E}[\|\mathbf{v} - \tilde{\mathbf{v}}\|^2], \tag{2.28}$$

where $\|\cdot\|$ is the euclidean norm.

Furthermore, the MSE function defined in (2.28), even for different orthonormal basis, can be equivalently expressed as:

$$\begin{aligned}\mathrm{J}_{\mathrm{PCA}}(\mathrm{U}_R) &= \mathbb{E}[\mathbf{v}^T\mathbf{v}] - \mathbb{E}[(\mathrm{U}_R^T\mathbf{v})^T(\mathrm{U}_R^T\mathbf{v})] = \mathbb{E}[\mathbf{v}^T\mathbf{v}] - \mathbb{E}[\mathbf{r}^T\mathbf{r}] \\ &= \mathrm{tr}(\mathrm{C}_{\mathbf{v}}) - \mathrm{tr}(\mathrm{C}_{\mathbf{r}}) \\ &= \sum_{i=1}^{L}\sigma_i - \mathrm{tr}(\mathrm{C}_{\mathbf{r}}),\end{aligned} \tag{2.29}$$

where $\mathrm{tr}(\cdot)$ is the trace of a matrix.

From (2.29), the MSE reconstruction function $\mathrm{J}_{\mathrm{PCA}}(\cdot)$ is minimum when $\mathrm{C}_{\mathbf{r}}$ holds the R largest eigenvalues of $\mathrm{C}_{\mathbf{r}}$, what has already been established in (2.26). Hence:

$$\min_{\mathrm{V}}\{\mathrm{J}_{\mathrm{PCA}}(\mathrm{V})\} = \mathrm{J}_{\mathrm{PCA}}(\mathrm{U}_R) = \sum_{i=R+1}^{L}\sigma_i, \tag{2.30}$$

where V is any orthonormal basis.

2.4.3 WHITENING

Whitening (or sphering) goes a step further than PCA by transforming $\mathbf{v}$ into a new random vector $\mathbf{z}$, such that the energy (or variance) is equally shared among its components. Furthermore, unlike PCA, whitening is not interested in dimensionality reduction nor retrieving latent variables from the data. Therefore, a whitening transformation B, producing an identity covariance matrix, is:

$$\mathrm{B} = \Sigma^{-1/2}\mathrm{U}^T \tag{2.31}$$

$$\mathbf{z} = \mathrm{B}\mathbf{v} \tag{2.32}$$

$$\mathrm{C}_{\mathbf{z}} = \mathrm{B}\mathrm{C}_{\mathbf{v}}\mathrm{B}^T = \mathrm{I}, \tag{2.33}$$

where $\mathrm{B} \in \Re^{L\times L}$, while $\Sigma \in \Re^{L\times L}$ must be invertible.

Interestingly, whitening is not affected by orthogonal transformations, i.e., $\mathbf{z}$ could be rotated by any orthogonal matrix Q, and its resulting covariance matrix would persist being an identity ($\mathrm{Q}\mathrm{C}_{\mathbf{z}}\mathrm{Q}^T = \mathrm{I}$), so any matrix of the form QB would also be a valid whitening transformation.

2.5 INDEPENDENT COMPONENT ANALYSIS

With respect to the nature of the sources, an assumption considered since the beginnings of BSS in the early 80's, as is presented in (JUTTEN; TALEB, 2000), is the statistical independence of the sources, which gives rise to the general concept of Independent Component Analysis (ICA), initially presented by Christian Jutten in (1987).

According to this hypothesis, the information contained in any subset of the sources does not bring forward anything significant to predict the behaviour of any other remaining subset in the vector. In mathematical terms, the independence of the M sources contained in $\mathbf{s}(t)$ implies that:

$$\mathrm{p}_{\mathbf{s}(t)}(\cdot) = \prod_{i}^{M} \mathrm{p}_{s_i(t)}(\cdot), \tag{2.34}$$

where the left-hand term refers to the joint PDF of the source vector $\mathbf{s}(t)$, while the right-hand term indicates the product of the marginal PDFs of the sources.

In a more strict sense, for two random processes to be independent, the joint PDF of any two sets of L time-states, one set for each process, should equal the product of the joint PDF of each process (PAPOULIS; PILLAI, 2002). However, this more general definition can be overlooked in the instantaneous ICA case, where the independent sources do not interact with each other at different time-states, hence $L = 1$ and (2.34) holds.

Independence is a much stronger condition than just uncorrelatedness, where other non-linear dependencies still subsist. Therefore, ICA may be understood as a generalization of PCA (JUTTEN; HERAUT, 1988), seeking for latent components that not only are uncorrelated, but also independent. However, decorrelating the mixtures is still useful as an initial step before attempting to solve the problem. This connection to PCA and whitening is explored in subsection 2.5.1.

Pierre Comon (1992; 1994) formalized the concept of ICA for the linear and instantaneous case, and proved that, in order to blindly capture a single source in each estimate, the independent source vector must have, at most, only one source with a Gaussian density function. This necessary second assumption about the Gaussianity of the sources is a direct result of the Darmois-Skitovich theorem, which will be presented and discussed in Section 2.5.2. Then, a simplified formulation of the definition presented by Comon, for linear instantaneous ICA, may be expressed as:

Definition 2.5.1 (Linear instantaneous ICA)**.** For an N-component random vector $\mathbf{x}$, ICA consists in finding a linear transformation W that produces a vector $\mathbf{y}$, whose components are as statistically independent as possible according to the optimization of a contrast

function $f(\cdot)$, or:

$$\begin{aligned} \mathbf{y} &= \mathrm{W}\mathbf{x} \\ &\text{subject to optimize } f(\mathbf{y}), \end{aligned} \tag{2.35}$$

where $\mathbf{y} \in \Re^M$, $\mathrm{W} \in \Re^{M\times N}$, while $f(\cdot)$ establishes the criterion that direct or indirectly expresses the notion of independence. ■

To sum up, ICA is based on a first and main assumption of the statistical independence of the sources, and a second necessary assumption limiting the number of Gaussian sources to a maximum of one. Additionally, for linear instantaneous ICA, the mixing matrix must be full rank. These conditions encapsulate the separability theorem presented in (SUYAMA, 2007), based on the works of Comon (1992; 1994).

2.5.1 PCA AND WHITENING AS PREPROCESSING FOR ICA

Independent random variables (and processes) are also uncorrelated, although the reverse is not generally true. It is only for Gaussian distributions that uncorrelatedness implies independence (PAPOULIS; PILLAI, 2002). Nonetheless, applying some decorrelation method on the mixtures can be seen as a step closer towards independence and ends up simplifying the task of the ICA technique.

Due to stationarity, this may be accomplished by applying PCA or whitening, as exposed in Section 2.4, before using any ICA algorithm. So, considering the linear instantaneous ICA model and the N-component mixture vector $\mathbf{x} = \mathrm{A}\mathbf{s}$, the initial covariance matrix would be:

$$\mathrm{C}_\mathbf{x} = \mathrm{A}\mathrm{C}_\mathbf{s}\mathrm{A}^T \tag{2.36}$$

$$= \mathrm{A}\mathrm{A}^T \tag{2.37}$$

$$= \mathrm{U}\Sigma\mathrm{U}^T, \tag{2.38}$$

where $\mathrm{A} \in \Re^{N\times M}$, $\mathrm{C}_\mathbf{s} \in \Re^{M\times M}$, while $\mathrm{U},\ \Sigma \in \Re^{N\times N}$ correspond to the SVD of $\mathrm{C}_\mathbf{x}$.

In (2.36), the source covariance matrix $\mathrm{C}_\mathbf{s}$ is diagonal as a consequence of independence, and, without loss of generality, it can be considered as the identity matrix, since the individual source variances are not relevant for ICA and BSS in general.

Thus, the resulting decorrelated mixtures by PCA and whitening, and their corresponding covariance matrices, are:

$$\mathbf{r} = \mathrm{U}^T\mathbf{x} \qquad \mathrm{C}_\mathbf{r} = \Sigma \tag{2.39}$$

$$\mathbf{z} = \Sigma^{-1/2}\mathrm{U}^T\mathbf{x} \qquad \mathrm{C}_\mathbf{z} = \mathrm{I}, \tag{2.40}$$

where $\mathbf{x},\ \mathbf{r},\ \mathbf{z} \in \Re^N$, and $\mathrm{C}_\mathbf{r},\ \mathrm{C}_\mathbf{z} \in \Re^{N\times N}$.

From (2.40), we recognize that whitening is only feasible when the eigenvalue matrix Σ is invertible, which translates to $\mathrm{C}_{\mathbf{x}}$ being full rank, i.e., $\mathrm{rank}(\mathrm{C}_{\mathbf{x}}) = N$, either in the well-determined ($N = M$) or underdetermined ($N < M$) scenarios.

Furthermore, by exploring the rank of matrix $\mathrm{C}_{\mathbf{x}}$ in (2.36), it is possible to shed some light on the structure of the full rank mixing matrix A (separability) as they must be equal, i.e., $\mathrm{rank}(\mathrm{C}_{\mathbf{x}}) = \mathrm{rank}(\mathrm{A})$. Thus, the following can be said about the unknown number of sources M and the "determinedness" of the problem:

$$\mathrm{rank}(\mathrm{C}_{\mathbf{x}}) = N \quad \leftrightarrow \quad \text{Well-determined, or underdetermined.} \tag{2.41}$$

$$\mathrm{rank}(\mathrm{C}_{\mathbf{x}}) < N \quad \leftrightarrow \quad \text{Overdetermined } (N > M), \text{ and } M = \mathrm{rank}(\mathrm{C}_{\mathbf{x}}). \tag{2.42}$$

Following (2.42), a valid strategy for the overdetermined problem could be reducing the number of components to M, by PCA, in order to deal with a well-determined setting. Additionally, considering M is already known, this strategy is also valid with noisy mixtures (JOHO; MATHIS; LAMBERT, 2000).

An interesting consequence of whitening is the orthogonalization of the mixing matrix A, when such matrix is square. According to the results presented in (2.40), a "new" whitened mixing matrix could be $\mathrm{A}_{\mathbf{z}} = \Sigma^{-1/2}\mathrm{U}^T\mathrm{A}$, for which orthogonality is satisfied ($\mathrm{A}_{\mathbf{z}}\mathrm{A}_{\mathbf{z}}^T = \mathrm{I}$). Then, in the well-determined scenario, the whitened mixtures are just a rotated version of the sources.

Therefore, after whitening and considering $N = M$, we can restrict the search space related to the separation matrix W to the set of orthogonal matrices, according to $\mathbf{y} = \mathrm{W}\mathbf{z}$. In this sense, whitening would be only a rotation away from attaining independence.

Example 2.5.1. Following Example 2.3.1, and considering only the mixtures produced by the full rank mixing matrix A, the resulting new basis $\mathrm{U} = [\mathbf{u}_1,\ \mathbf{u}_2]$ and whitening transformation B are:

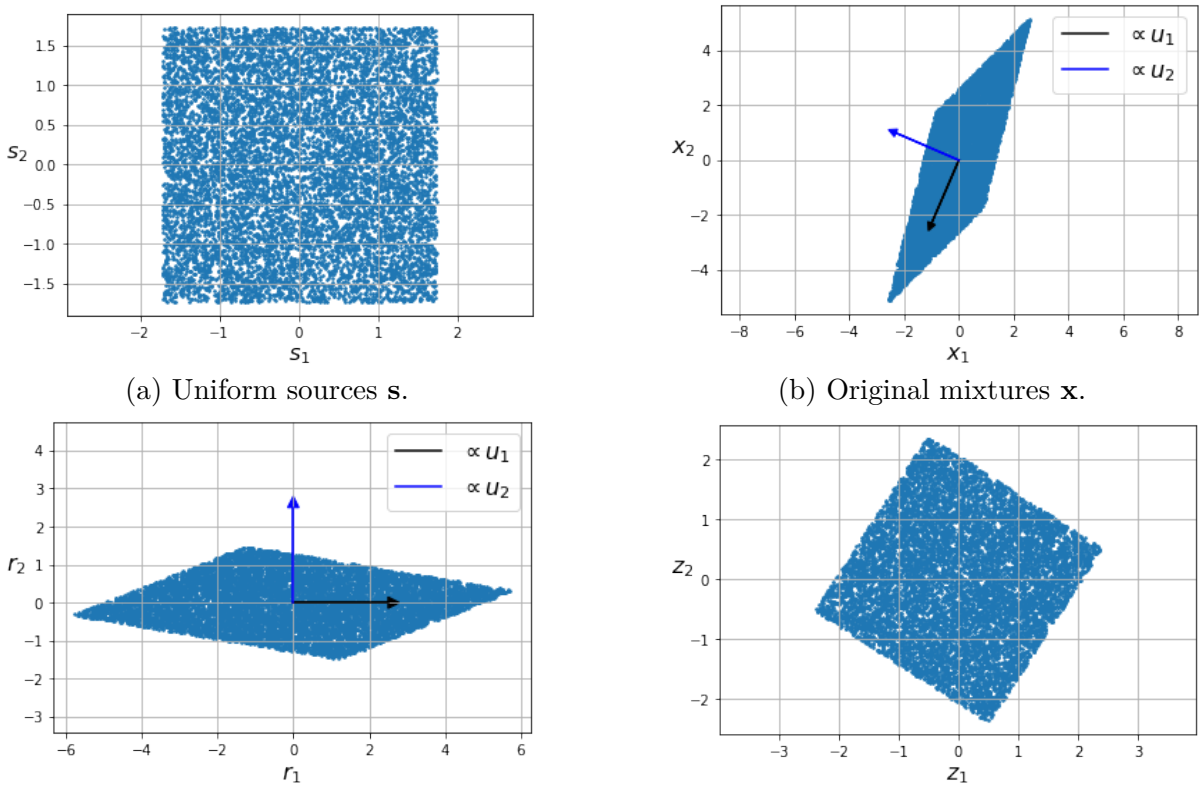
$$\mathrm{U} = \begin{bmatrix} -0.4 & -0.92 \\ -0.92 & 0.4 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 5.87 & 0 \\ 0 & 0.38 \end{bmatrix} \qquad \mathrm{B} = \begin{bmatrix} -0.16 & -0.38 \\ -1.48 & 0.64 \end{bmatrix}, \tag{2.43}$$

where all matrices are rounded up to two decimals.

Figure 2.3 shows the subsequent uncorrelated mixtures. In Figure 2.3c, the mixtures have been projected onto the eigenvectors u_1 and u_2, shown unmodified in Figure 2.3b, whereas in Figure 2.3d the distribution of the whitened mixtures resemble a rotated version of the distribution of the sources, depicted in Figure 2.3a. ■

2.5.2 ON NON-GAUSSIANITY

As already mentioned, uncorrelatedness and independence are equivalent for random variables with Gaussian distributions. Then, in order to avoid retrieving inde-



(a) Uniform sources $\mathbf{s}$.

(b) Original mixtures $\mathbf{x}$.

(c) Decorrelated mixtures $\mathbf{r}$ by PCA.

(d) Whitened mixtures $\mathbf{z}$.

Figure 2.3 – Preprocessing the data: Whitening is a rotation away from independence.

pendent estimates capturing uncorrelated mixtures of Gaussian sources, a restriction on their gaussianity must be considered. This restriction is based on the Darmois-Skitovich theorem (KAGAN; LINNIK; RAO, 1973):

Theorem 2.5.2 (Darmois-Skitovich)**.** Consider L zero-mean and independent random variables $v_1,\ v_2,\ \cdots,\ v_L$, and their linear combinations u_1 and u_2, expressed as follows:

$$\begin{aligned} u_1 &= \sum_{i=1}^{L} a_i v_i \\ u_2 &= \sum_{i=1}^{L} b_i v_i, \end{aligned} \tag{2.44}$$

where all a_i and b_i are constant scalars.

If u_1, u_2 are independent, then, for any $a_i b_i \neq 0$, v_i must be Gaussian. ■

Following the premise of the theorem, there is no limitation to how many independent random variables v are simultaneously present in the also independent pair u_1 and u_2, the only requirement being that the shared variables v must be Gaussian. Then, by applying this to the linear instantaneous ICA problem, while considering no limit to the Gaussianity of the sources, it could not be possible to always capture a single source s_i in each estimate y_i with just the assumption of independence, so the number of Gaussian sources should be limited to one to avoid retrieving such independent mixtures.

Example 2.5.3. Consider the Gaussian and independent source vector $\mathbf{s} = (s_1,\ s_2)^T$, mixed by some matrix A, and a candidate 2-component estimate vector $\mathbf{y} = (\mathrm{WA})\mathbf{s}$:

$$\begin{aligned} y_1 &= wa_{1,1}s_1 + wa_{1,2}s_2 \\ y_2 &= wa_{2,1}s_1 + wa_{2,2}s_2, \end{aligned} \tag{2.45}$$

where each $wa_{i,j}$ is a coefficient of matrix WA at a row i and column j, while y_1, y_2, and s_1, s_2 are Gaussian.

In (2.45), y_1 and y_2 are Gaussian due to being linear combinations of s_1 and s_2. So, their uncorrelatedness implies independence.

One of the infinite solutions for a "separation" matrix W, such that it ensures independence, and considering the mixing matrix A used in Example 2.5.1, corresponds to the whitening transformation B from the same example, both repeated here for convenience:

$$\mathrm{B} = \begin{bmatrix} -0.16 & -0.38 \\ -1.48 & +0.64 \end{bmatrix} \qquad \mathrm{A} = \begin{bmatrix} 0.5 & 1 \\ 2 & 1 \end{bmatrix} \qquad \mathrm{WA} = \begin{bmatrix} -0.84 & -0.54 \\ +0.54 & -0.84 \end{bmatrix}, \tag{2.46}$$

where B becomes a false separation matrix, B = W.

Thus, a perfectly possible solution according to Theorem 2.5.2 would be:

$$\begin{aligned} y_1 &= -0.84s_1 - 0.54s_2 \\ y_2 &= +0.54s_1 - 0.84s_2, \end{aligned} \tag{2.47}$$

where y_1 and y_2 are independent whitened mixtures.

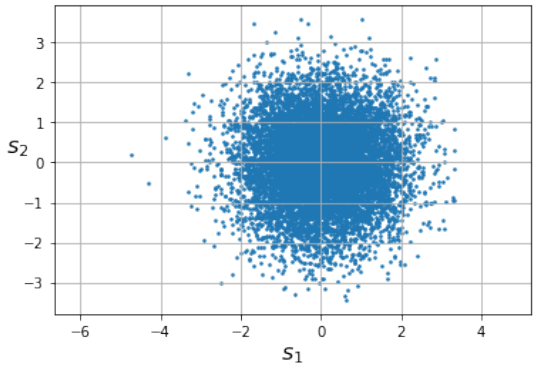
Figure 2.4 depicts this false solution, where Figure 2.4d corresponds to the independent estimates, y_1 and y_2, retrieved by whitening the initially correlated available mixtures in Figure 2.4b. These estimates, although being independent, are still mixtures of the original standard Gaussian sources, s_1 and s_2, shown in Figure 2.4a. The mixtures after PCA are shown in Figure 2.4c only for completion. Unlike Figure 2.3 in Example 2.5.1, it is not possible to blindly determine an adequate additional rotation needed to actually separate the sources based only on their independence.

On the other hand, if only one source was Gaussian, for example s_1, then s_2 could not be shared among independent estimates, and s_1 could only be in the remaining estimate to guarantee the independence of the pair. Thus, as a consequence of limiting gaussianity, each estimate would capture only one source with no repetition. ■

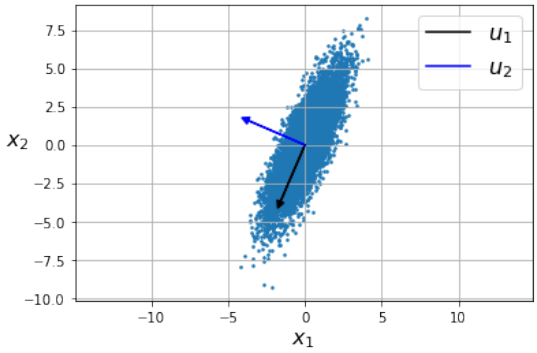
2.6 OTHER APPROACHES

2.6.1 SECOND ORDER STATISTICS

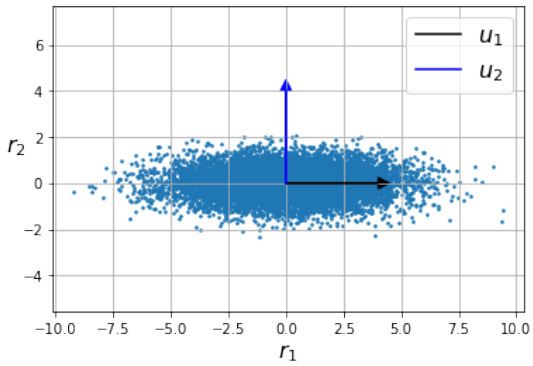
Second Order Statistic (SOS) algorithms are based on the weaker assumption of uncorrelatedness of the sources, and do not require their non-gaussianity. Furthermore,



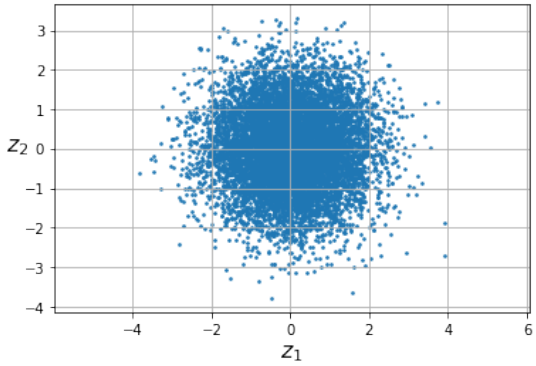
(a) Gaussian sources $\mathbf{s}$.



(b) Original mixtures $\mathbf{x}$.



(c) Decorrelated mixtures $\mathbf{r}$ by PCA.



(d) Whitened mixtures $\mathbf{z}$.

Figure 2.4 – The problem of gaussianity.

regarding the stationarity of the signals, SOS algorithms may be divided between those based on WSS, as explained in Section 2.1, and those based on nonstationarity. An example of the first group is the Algorithm for Multiple Unknown Signals Extraction (AMUSE) (TONG et al., 1990).

AMUSE relies on finding some time delay τ such that the corresponding cross-covariance function matrix $\mathrm{K}_{\mathbf{s}}(\tau)$ of the source vector is not only diagonal, due to uncorrelatedness, but its components must be different after some orthogonalization procedure of the mixing matrix, due to the sources having different correlation structures.

$$\mathbb{E}[s_i(t)s_i(t+\tau)] \neq \mathbb{E}[s_j(t)s_j(t+\tau)], \tag{2.48}$$

where each s_i has been centered and is unit-variance, $i \neq j$, and τ is a delay to be found.

After whitening, as indicated in Section 2.5.1 for $\tau = 0$, the mixing matrix $\mathrm{A}_{\mathbf{z}}$ can be considered to be orthogonal and the sources normalized, as their variances have been "absorbed". Then, considering $\mathbf{z} = \mathrm{A}_{\mathbf{z}}\mathbf{s}$, it follows that the cross-covariance matrix of the whitened mixtures $\mathrm{K}_{\mathbf{z}}(\tau)$ is diagonalized by the orthogonal $\mathrm{A}_{\mathbf{z}}$:

$$\mathrm{K}_{\mathbf{z}}(\tau) = \mathrm{A}_{\mathbf{z}}\mathrm{K}_{\mathbf{s}}(\tau)\mathrm{A}_{\mathbf{z}}^T = \mathrm{A}_{\mathbf{z}}\mathrm{K}_{\mathbf{s}}(\tau)\mathrm{A}_{\mathbf{z}}^{-1}, \tag{2.49}$$

where $\mathrm{A}_{\mathbf{z}} \in \Re^{N\times N}$ is orthogonal, thus $\mathrm{A}_{\mathbf{z}}^T = \mathrm{A}_{\mathbf{z}}^{-1}$.

After computing the eigendecomposition of $\mathrm{K}_{\mathbf{z}}(\tau)$, expressed in (2.50), the mixing matrix $\mathrm{A}_{\mathbf{z}}$ may be identified as the resulting orthogonal eigenvector matrix U_{τ}, but only when all the corresponding eigenvalues are different, which is equivalent to condition (2.48) for all sources, as $\mathrm{K}_{\mathbf{s}}(\tau) = \Sigma_{\tau}$. Otherwise, the repeated eigenvalues would not be able to define a unique direction for their eigenvectors. Therefore, a separation matrix W is set to be equal to U_{τ}^{T} and, thus, the problem is solved:

$$\mathrm{K}_{\mathbf{z}}(\tau) = \mathrm{U}_{\tau}\Sigma_{\tau}\mathrm{U}_{\tau}^{-1} \tag{2.50}$$

$$\mathbf{y} = \mathrm{U}_{\tau}^{T}\mathbf{z}, \tag{2.51}$$

where $\mathrm{U}_{\tau},\ \Sigma_{\tau} \in \Re^{N\times N}$ and Σ_{τ} has mutually distinct diagonal components.

Although AMUSE is advantageous in its conceptual simplicity, unfortunately, due to estimation errors and not always counting with distinct enough eigenvalues (YEREDOR, 2010; ROMANO et al., 2011), the results delivered by a single time-delay are not always an accurate representation of the actual sources. Moreover, in the original paper there is no criterion for a proper selection of τ.

Other algorithms, such as Second Order Blind Indentification (SOBI) (BELOUCHRANI et al., 1997) and Temporal Decorrelation Source Separation (TDSEP) (ZIEHE; MULLER, 1998), consider a number of different delays in order to compute a joint diagonalizer U_{jd} of the corresponding set of cross-covariance matrices $\mathrm{K}_{\mathbf{z}}(\tau)$, such that it minimizes a diagonalization error, therefore producing a more robust separation matrix, $\mathrm{W} = \mathrm{U}_{jd}^{T}$, able to deliver more accurate results.

2.6.2 SPARSE COMPONENT ANALYSIS

Sparse Component Analysis (SCA) is based on the assumption that jointly sparse mixtures are the manifestation of sparse sources with disjoint supports in some domain, i.e., when one sparse source presents values different from zero, the rest should be approximately inactive, even if the support is only instantaneous. As a consequence, considering a linear instantaneous setting, the direction of every column of the mixing matrix is revealed through the geometry of the observable mixtures. Thus, the mixing matrix can be estimated from scatter plots, which is particularly useful in the underdetermined case.

Example 2.6.1. This interesting geometrical result can be exemplified considering three audio signals s_1, s_2 and s_3, shown in Figure 2.5a, which are not only sparse, but mostly active one at a time (approximately disjoint support), and an underdetermined mixing matrix A:

$$\mathrm{A} = [\mathbf{a_1},\ \mathbf{a_2},\ \mathbf{a_3}] = \begin{bmatrix} 0.75 & 2 & 1 \\ -0.75 & 0.25 & 0.5 \end{bmatrix}, \tag{2.52}$$

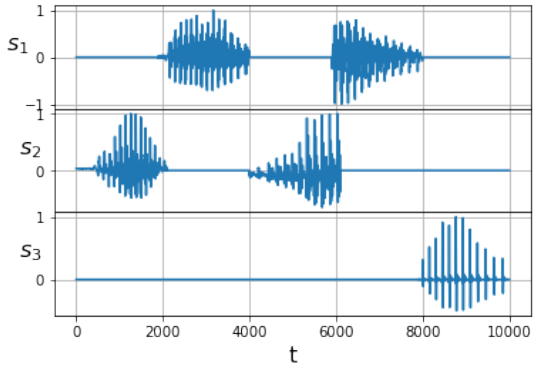
Figure 2.5b suggests that the available mixtures, x_1 and x_2, are the product of sparse sources with disjoint support. Additionally, the scatter plot in Figure 2.5c shows their joint sparsity and also reveals the structure of the mixing matrix by making visible the direction of each column, and, thus, the number of sources when such columns are not co-linear. This result, considering $\mathbf{x} = \sum_i^3 a_i s_i$, may be expressed in terms of the disjoint supports of the sources:

$$\mathbf{x}(t_1) \approx \mathbf{a_1} s_1(t_1) \tag{2.53}$$

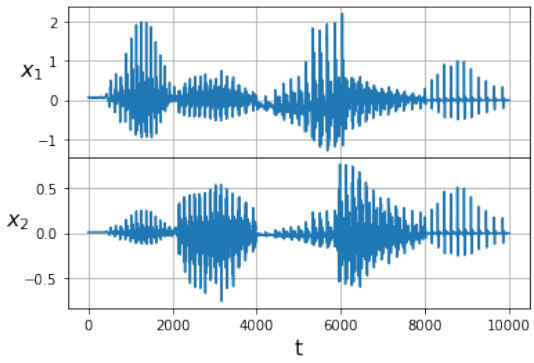
$$\mathbf{x}(t_2) \approx \mathbf{a_2} s_2(t_2) \tag{2.54}$$

$$\mathbf{x}(t_3) \approx \mathbf{a_3} s_3(t_3), \tag{2.55}$$

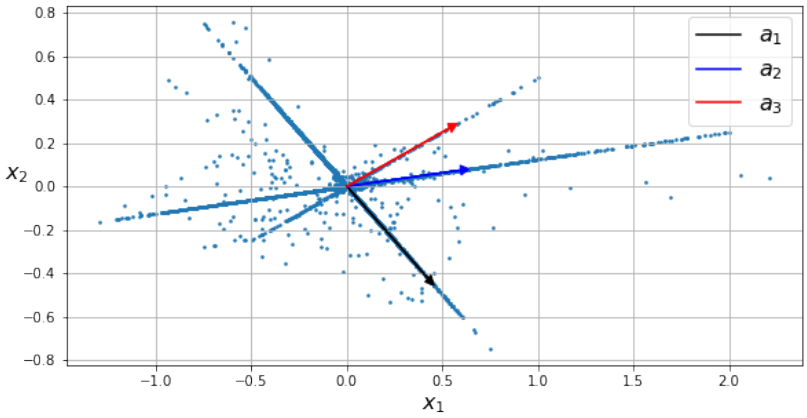
where $t_1 \in [1900,\ 4000] \cup [5900,\ 8000]$, $t_2 \in [0,\ 2100] \cup [4000,\ 6100]$, and $t_3 \in [7900,\ 10000]$ are time-samples from the (approximately) disjoint supports of sources. ■



(a) Sources $\mathbf{s}$ with disjoint time-supports.



(b) Sparse mixtures $\mathbf{x}$ w.r.t. time.



(c) Scatter plot of the mixtures $\mathbf{x}$.

Figure 2.5 – Geometry of sparsity in the time domain.

Although it may seem plausible to determine the directions of the mixing matrix by simple inspection, as in the example, it is actually a task better suited for clustering algorithms, such as those studied in (ARBERET; GRIBONVAL; BIMBOT, 2006; MOVAHEDI et al., 2008; HE et al., 2009; YI et al., 2019), which estimate the mixing

matrix in more complicated scenarios. Then, once the mixing matrix is estimated, the source estimates can be computed.

Furthermore, the example also emphasizes the importance of having completely disjoint sources, as a little overlap in their supports will difficult the correct identification of the clusters from the scatter plot. Therefore, it is usually convenient to transform the mixtures into a different domain, e.g. frequency or time-frequency (PRINCEN; BRADLEY, 1986; SANEI et al., 2005; O'GRADY; PEARLMUTTER; RICKARD, 2005), in order to ensure truly disjoint sources.

To summarize, the SCA problem may be divided into four steps, as explained in (GRIBONVAL; ZIBULEVSKY, 2010):

1. Transformation into a domain where the sources have supports as disjoint as possible.
2. Estimation of the mixing matrix from the clusters obtained in the scatter plots.
3. Retrieval of the estimates corresponding to the new domain.
4. Inversion of the transformation to reconstruct the sources in the original domain.

2.6.3 BAYESIAN APPROACH

Unlike previous methodologies, the Bayesian approach, as explained in (KNUTH, 1999; MOHAMMAD-DJAFARI, 2001), does not depend on an specific set of characteristics about the sources, such as their statistical independence, in order to define a procedure to find adequate estimates. Instead, by applying Bayes' theorem (PAPOULIS; PILLAI, 2002) to the BSS problem, it predicts the parameters of the model, i.e., the values it may take, based on the observed signals, and on any available prior knowledge of the mixing matrix and sources, thus enriching the predicted model.

In order to predict these parameters, Bayes' theorem establishes a probabilistic relation between the posterior PDF of the model, or $p_{A,\mathbf{s}|\mathbf{x}}(\cdot)$, the likelihood $p_{\mathbf{x}|A,\mathbf{s}}(\cdot)$, and the priors $p_A(\cdot)$ and $p_{\mathbf{s}}(\cdot)$. This relation is expressed as:

$$p_{A,\mathbf{s}|\mathbf{x}}(\cdot) = \frac{p_{\mathbf{x}|A,\mathbf{s}}(\cdot)p_A(\cdot)p_{\mathbf{s}}(\cdot)}{p_{\mathbf{x}}(\cdot)}, \tag{2.56}$$

where $p_A(\cdot)$ only expresses a belief on the values the constant matrix could take, while $p_{\mathbf{x}}(\cdot)$ does not depend on the model (unlike the likelihood), and $\mathbf{x}$ is already known.

According to the Maximum A Posteriori (MAP) principle, the optimal estimates $\widehat{A}$ and $\widehat{\mathbf{s}}$ are those which maximize the posterior PDF in (2.56) (MOHAMMAD-DJAFARI; KNUTH, 2010). Then, by using the logarithmic function, it is possible to find the afore-

mentioned parameters following different estimation approaches:

$$\widehat{\mathrm{A}},\ \widehat{\mathbf{s}} = \arg\max_{\mathrm{A},\ \mathbf{s}} \{\log(\mathrm{p}_{\mathbf{x}|\mathrm{A},\mathbf{s}}(\mathbf{x} \mid \mathrm{A},\ \mathbf{s})) + \log(\mathrm{p}_{\mathrm{A}}(\mathrm{A})) + \log(\mathrm{p}_{\mathbf{s}}(\mathbf{s}))\} \tag{2.57}$$

$$\widehat{\mathrm{A}} = \arg\max_{\mathrm{A}} \{\log(\mathrm{p}_{\mathbf{x}|\mathrm{A}}(\mathbf{x} \mid \mathrm{A})) + \log(\mathrm{p}_{\mathrm{A}}(\mathrm{A}))\} \tag{2.58}$$

$$\widehat{\mathbf{s}} = \arg\max_{\mathbf{s}} \{\log(\mathrm{p}_{\mathbf{x}|\mathbf{s}}(\mathbf{x} \mid \mathbf{s})) + \log(\mathrm{p}_{\mathbf{s}}(\mathbf{s}))\}, \tag{2.59}$$

where (2.57) is a joint estimation of the parameters, while (2.58) is the marginal estimation of only the mixing matrix, and (2.59) is the marginal estimation of the sources.

Example 2.6.2. Prior knowledge about the model is easily incorporated in any approach. For example, assuming that the coefficients of A come from the standard Gaussian distribution, while the M sources are not only statistically independent, as in ICA, but all share the same uniform distribution, then their priors can be expressed as:

$$\ln(\mathrm{p}_{\mathrm{A}}(\mathrm{A})) = NM \ln\left(\frac{1}{\sqrt{2\pi}}\right) - \sum_{j=1}^{M}\sum_{i=1}^{N} \frac{a_{i,j}^2}{2} \tag{2.60}$$

$$\ln(\mathrm{p}_{\mathbf{s}}(\mathbf{s})) = -M \ln(s_{\max} - s_{\min}), \tag{2.61}$$

where each $a_{i,j}$ corresponds to a coefficient of A at a row i and column j.

Additionally, considering only the marginal estimation of the mixing matrix, and $N = M$, the likelihood and log-likelihood can be expressed by means of the PDF transformation (HYVäRINEN; KARHUNEN; OJA, 2001) as:

$$\mathrm{p}_{\mathbf{x}|\mathrm{A}}(\mathbf{x} \mid \mathrm{A}) = \frac{1}{|\det(\mathrm{A})|} \mathrm{p}_{\mathbf{s}}(\mathrm{A}^{-1}\mathbf{x}) \tag{2.62}$$

$$\log(\mathrm{p}_{\mathbf{x}|\mathrm{A}}(\mathbf{x} \mid \mathrm{A})) = -\ln(|\det(\mathrm{A})|) - N \ln(s_{\max} - s_{\min}), \tag{2.63}$$

where A^{-1} exists, $\mathbf{s} = \mathrm{A}^{-1}\mathbf{x}$, and $\det(\cdot)$ is the determinant function.

Thus, the assumptions of the model are made explicit, leaving the non-trivial task of maximizing the priors together with the likelihood in order to estimate the parameters of the model. ■

2.7 PRACTICAL APPLICATIONS

Due to is broad formulation, BSS may be applied to a wide variety of problems in many different fields, whenever there is only indirect access to the hidden signals of interest. A few examples in popular fields of application are:

BIOMEDICINE

A classical application of BSS in the field of biomedicine is fetal electrocardiography. In such application, the objective is to capture, in a non-invasive procedure, the

electrical signals produced by the heart of the fetus from mixtures of many electrical signals, all received by well-located electrodes on the skin of the mother. To solve this challenging problem, different methods based on linear instantaneous ICA (LATHAUWER; MOOR; VANDEWALLE, 2000), higher-order statistics BSS (ZARZOSO; NANDI, 2001), and BSS in the wavelet domain (JAFARI; CHAMBERS, 2005) have been proposed. Similarly, other tasks involving electroencephalography and magnetoencephalography signals can also be interpreted as BSS problems.

AUDIO PROCESSING

The cocktail party problem (CHERRY, 1953; HAYKIN; CHEN, 2005) is a well-known example of how BSS occurs naturally when, in a noisy environment with multiple conversations happening simultaneously, it is necessary to focus on an audio source from an specific person, thus separating this particular source from the mixtures.

Other examples of BSS in audio processing involve automatic music transcription (PLUMBLEY et al., 2002), detection of underwater elements by sonar systems (FAN; ZHANG; JIANG, 2010), and speech enhancement for voice command of vehicles (LEI; CHEN; WANG, 2019).

TELECOMMUNICATIONS

As presented in (LUO; LI; ZHU, 2018), the appeal of BSS in telecommunications is due to its capacity to recover incoming signals, mixed in a wireless receiver, based only on the knowledge of some of their features. This is especially advantageous when considering the increasing number of signals crowding the frequency spectrum, thus lowering its efficiency, hindering the proper detection of the signals and increasing any interference.

Applications of BSS in this field are found in Radio Frequency Identification, or RFID, systems to avoid overlapping signals (MINDIKOGLU; VEEN, 2008), encryption methods for secure military communications (DOUKAS; KARADIMAS, 2008), among others.

FINANCE

Although financial time-series are usually non-stationary and noisy, BSS may be useful in finding some underlying structure, in the form of sources and a mixing system, able to explain, and maybe predict, their evolution.

In (BACK; WEIGEND, 1997), it is shown that the daily stock price variation in the Japanese stock market may depend on two groups of independent components: a first group of large, but infrequent, shocks in price, and a second group explaining slight frequent variations. In (KIVILUOTO; OJA, 1998), the matter of interest is explaining

weekly cashflow variations. A different approach is offered in (LU; LEE; CHIU, 2009), where ICA is used in a first step, and, in a second step, support vector regression forecast the time-series from the resulting independent components.

2.8 SUMMARY

In this chapter, we have attempted to describe the fundamentals necessary to formally define the BSS problem: from the requirements the signals must comply, the characteristics to be considered for an appropriate modelling, up to the estimates we should obtain according to the assumptions made about the nature of sources. Some practical applications have also been considered to illustrate the versatility of BSS as a general framework to find meaningful and hidden signals in various scenarios.

Regarding the solution to the problem, we have presented some of the most important approaches, such as ICA, SCA and the Bayesian approach, to try to understand how different sets of assumptions about the sources define different methodologies to retrieving corresponding estimates. Additionally, we included PCA as a sort of precursor to BSS, especially ICA, which simplified the final solution.

As already mentioned, we consider ICA to be the most important of the approaches to BSS due to its simple formulation and realistic, though stringent, assumptions about the sources. In the following chapter, we aim at explaining some of the existing quantities to measure independence, as well as some relevant algorithms for the development of our research.

3 ICA: CRITERIA AND ALGORITHMS

As discussed in Section 2.5, after the preprocessing step is complete and the mixtures are whitened, there remains the task of searching for the right rotation that will separate the signals into independent components. In order to find a separation matrix that yields this rotation, up to a subsequent permutation and scaling, it is necessary to first define a functional, or criterion, such that its optimization leads to independence.

In this chapter, we expose some of the main criteria adopted in ICA, as well as briefly present the ideas of two classical algorithms: FastICA (HYVäRINEN; OJA, 1997; HYVäRINEN, 1999) and JADE (CARDOSO; SOULOUMIAC, 1993).

3.1 MUTUAL INFORMATION

For an L-component random vector $\mathbf{v}$, Mutual Information $\mathrm{MI}(\cdot)$ can be defined as the non-negative quantity measuring how different the joint PDF is from the product of the corresponding marginal densities (i.e. the independent PDF), being null if and only if both PDFs are equal at almost every point in their domain (COVER; THOMAS, 2006), i.e., when the random vector is composed of statistically independent components. Mutual information can be equivalently expressed as either of the following expressions:

$$\mathrm{MI}(\mathbf{v}) = \mathrm{D}_{\mathrm{KL}}\left(\mathrm{p}_{\mathbf{v}} \parallel \prod_{i=1}^{L} \mathrm{p}_{v_i}\right) \tag{3.1}$$

$$\mathrm{MI}(\mathbf{v}) = \left(\sum_{i=1}^{L} \mathrm{H}(\mathrm{p}_{v_i})\right) - \mathrm{H}(\mathrm{p}_{\mathbf{v}}), \tag{3.2}$$

where $\mathbf{v} \in \Re^L$, $\mathrm{D}_{\mathrm{KL}}(\cdot)$ is the Kullback-Leibler divergence between two PDFs, and $\mathrm{H}(\cdot)$ is the differential entropy of a PDF, both defined in (COVER; THOMAS, 2006).

In a well-determined scenario, by minimizing the mutual information of the estimate vector $\mathbf{y} = \mathrm{W}\mathbf{z}$ with respect to the separation matrix W, the recovered estimates y_i tend to become as independent as possible, which meets the objective of ICA. In such case, by exploring properties of the differential entropy, $\mathrm{MI}(\cdot)$ can be written as (HYVäRINEN; KARHUNEN; OJA, 2001):

$$\min_{\mathrm{W}} \{\mathrm{MI}(\mathbf{y} = \mathrm{W}\mathbf{z})\} = \min_{\mathrm{W}} \left\{ \left(\sum_{i=1}^{N} \mathrm{H}(\mathrm{p}_{y_i})\right) - \mathrm{H}(\mathrm{p}_{\mathbf{z}}) - \log(|\det(\mathrm{W})|) \right\}, \tag{3.3}$$

where $\mathbf{z} \in \Re^N$ corresponds to the whitened mixtures and $\mathrm{H}(\mathrm{p}_{\mathbf{z}})$ does not depend on the separation matrix W.

As implied by (3.1) and (3.2), in order to compute $\mathrm{MI}(\mathbf{v})$, it is necessary to estimate the involved densities, which corresponds to a non-parametric task. Therefore,

other quantities, such as those based on non-gaussianity, are also used to indirectly measure independence.

3.1.1 THE INFORMATION MAXIMIZATION PRINCIPLE

Introduced in (BELL; SEJNOWSKI, 1995), the Information maximization principle, or simply Infomax, exploits the architecture and flexibility of artificial neural networks to model a separation system analogous to a single layer perceptron (Section 4.1), such that the set of samples of the mixture vector $\mathbf{x}$ should be mapped to an output $\mathbf{y}$ with a maximum transference of information between input and output.

The assumed architecture comprises a matrix W of adaptable parameters and a subsequent array $\mathbf{g}(\cdot)$ of non-linear and invertible transformations $g_i(\cdot)$, such that the final output is $\mathbf{y} = \mathbf{g}(W\mathbf{x})$, which is more general than linear ICA. Thus, the gist of the Infomax principle is to find a separation matrix W that maximizes the mutual information $MI(\cdot)$ between $\mathbf{x}$ and $\mathbf{y}$, which, from (3.2), can be expressed as:

$$\begin{aligned} MI(\mathbf{x},\ \mathbf{y}) &= H(p_{\mathbf{x}}) + H(p_{\mathbf{y}}) - H(p_{\mathbf{x},\ \mathbf{y}}) \\ &= H(p_{\mathbf{y}}) - H(p_{\mathbf{y}|\mathbf{x}}), \end{aligned} \tag{3.4}$$

where $H(p_{\mathbf{y}|\mathbf{x}})$ indicates the conditional differential entropy of $\mathbf{y}$ given $\mathbf{x}$.

Due to the deterministic nature of the mapping between $\mathbf{x}$ and $\mathbf{y}$ (assuming there is no noise and no source of stochasticity other than $\mathbf{x}$), the conditional differential entropy $H(p_{\mathbf{y}|\mathbf{x}})$ is invariant to W and diverges to minus infinity, as there is no uncertainty about $\mathbf{y}$ once $\mathbf{x}$ is known.

Thus, the maximization of $MI(\mathbf{x},\ \mathbf{y})$ with respect to W is equivalent to the maximization of the differential entropy $H(p_{\mathbf{y}})$, which is maximum only when $MI(\mathbf{y}) \geq 0$ is minimum and, therefore, $\mathbf{y}$ is composed of statistically independent components:

$$\arg\max_{W}\{MI(\mathbf{x},\ \mathbf{y})\} = \arg\max_{W}\{H(p_{\mathbf{y}}) = \sum H(p_{y_i}) - MI(\mathbf{y})\} \tag{3.5}$$

$$= \arg\min_{W}\{MI(\mathbf{y})\} \tag{3.6}$$

Furthermore, in (CARDOSO, 1998), it is demonstrated that the maximization objective of the Infomax principle is equivalent to the maximization of the likelihood of $\mathbf{x}$ given $\mathbf{A}$ and $\mathbf{s}$ (Section 2.6.3). Hence, both principles yield similar results.

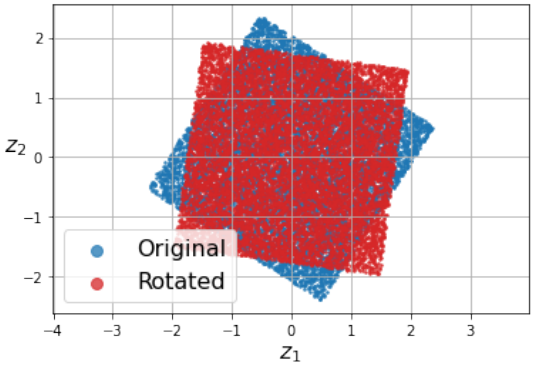
3.2 NON-GAUSSIANITY

Based on the Central Limit Theorem (CLT) (PAPOULIS; PILLAI, 2002), it is possible to state that each nontrivial mixture tends to be "more Gaussian" as more sources

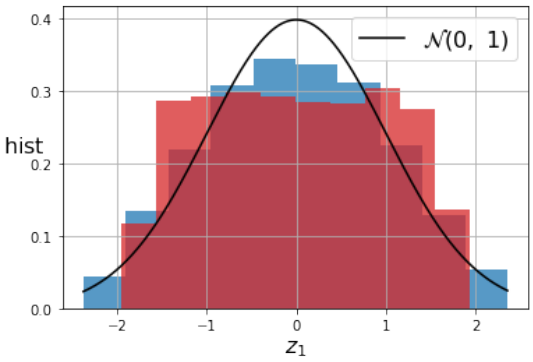
equally contribute to its generation. Therefore, to seek a transformation that maximizes non-Gaussianity for all mixtures may lead to source separation, as only one dominant source may be present in each one. Two classic measures to evaluate non-Gaussianity are kurtosis and negentropy (ROMANO et al., 2011).

Example 3.2.1. Considering the same whitened mixture vector $\mathbf{z}$ as in Example 2.5.1, as well as the same mixing matrix A and source vector $\mathbf{s}$, Figure 3.1 illustrates the connection between non-Gaussianity and independence.

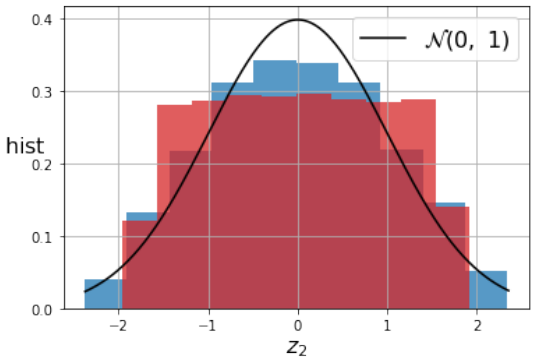
In Figure 3.1a, a 25° counterclockwise rotation (in red) closer to independence leads to less Gaussian components, as can be seen in Figure 3.1b and Figure 3.1c, where the red histograms resemble more those of the uniform sources, while those of the original whitened mixtures (in blue) are more similar to a standard Gaussian PDF $\mathcal{N}(0,\ 1)$.



(a) Whitened mixtures $\mathbf{z}$.



(b) Histogram of z_1.



(c) Histogram of z_2.

Figure 3.1 – Non-gaussianity maximization leads to independence.

■

3.2.1 KURTOSIS

A possible way to quantify the Gaussianity of a signal is by estimating a quantity known as (excess) Kurtosis.

Considering a zero-mean signal v for convenience, its kurtosis may be defined as the fourth-order cumulant of v (HYVäRINEN; KARHUNEN; OJA, 2001) (more on

cumulants in Section 3.3), which is expressed in terms of the fourth and second-order moments as:

$$\text{Kurt}(v) = \mathbb{E}[v^4] - 3\mathbb{E}[v^2]^2, \tag{3.7}$$

where Kurt$(\cdot)$ refers to the kurtosis.

Interestingly, considering unit-variance signals, a simple classification of the PDFs can be attained according to the value of the kurtosis, which ranges from -2 to infinity in this case. A PDF with negative kurtosis is called subgaussian, whereas those with positive kurtosis are considered as supergaussian. Common examples belonging to these two groups of PDFs are the uniform and the Laplacian distributions, respectively.

Therefore, the idea of maximizing the non-Gaussianity of the estimated sources, which was suggested in Section 2.5.2, can be translated into the task of maximizing the absolute value of the normalized kurtosis, which simply is the kurtosis in (3.7) but divided by the squared variance, and it is expressed as follows:

$$\max_{\text{W}} \{|\text{kurt}(y_i)|\} = \max_{\text{W}} \left\{ \left| \frac{\mathbb{E}[y_i^4]}{\text{var}(y_i)^2} - 3 \right| \right\}, \tag{3.8}$$

where kurt$(\cdot)$ refers to the normalized kurtosis.

It is pertinent to remark that the Gaussian distribution is not the only PDF with kurtosis equal to zero. Fortunately, the other PDFs with this characteristic are quite atypical and most likely do not model any signal of practical interest. Additionally, the kurtosis is sensitive to the presence of outliers, which can be an obstacle to the precise estimation of its value.

3.2.2 NEGENTROPY

Considering all random vectors with complete support on the real space $\Re^L$, maximum entropy is achieved by the Gaussian distribution, among all possible distributions with the same mean vector and a constant covariance matrix (COVER; THOMAS, 2006).

Then, it is possible to establish a non-negative measure of non-Gaussianity in terms of the difference between the maximum entropy and that of a vector $\mathbf{v}$, both with the same covariance matrix $\text{C}_\mathbf{v}$. This quantity is called Negentropy, denoted by J$(\cdot)$, and is defined as follows:

$$\begin{aligned} \text{J}(\mathbf{v}) &= \text{H}(\mathcal{N}(0,\ \text{C}_\mathbf{v})) - \text{H}(\text{p}_\mathbf{v}) \\ &= \frac{1}{2}\ln(|\det(\text{C}_\mathbf{v})|) + \frac{L}{2}(1 + \ln(2\pi)) - \text{H}(\text{p}_\mathbf{v}), \end{aligned} \tag{3.9}$$

where $\mathbf{v} \in \Re^L$, $\text{C}_\mathbf{v} \in \Re^{L \times L}$, while $\mathcal{N}(\cdot)$ is the Gaussian distribution.

Negentropy is zero exclusively for the Gaussian distribution, with no exceptions, and its value always increases the less Gaussian the PDF of the random vector $\mathbf{v}$ is.

Due to negentropy being invariant to invertible linear transformations, expression (3.9) is more useful for the individual components after any transformation. Thus, a matrix W should yield candidate components y_i such that their marginal negentropy is maximum, which is expressed as:

$$\max_{\mathrm{W}} \{\mathrm{J}(y_i)\} = \max_{\mathrm{W}} \left\{ \frac{1}{2} \ln(\mathrm{var}(y_i)) + \frac{1}{2}(1 + \ln(2\pi)) - \mathrm{H}(\mathrm{p}_{y_i}) \right\}, \tag{3.10}$$

Similarly to the mutual information, negentropy also requires the knowledge or an approximation of the involved PDFs for its computation.

3.2.3 FASTICA

FastICA is a fixed-point algorithm, proposed in (HYVäRINEN; OJA, 1997; HYVäRINEN, 1999), which aims at maximizing the non-Gaussianity of the individual mixtures after whitening, either measured by negentropy or by the absolute value of the kurtosis, thus finding adequate approximations of the independent sources. In this work, we shall consider the negentropy-based version of FastICA.

To accomplish this, a fixed-point iteration is employed to converge from a random initialization to a single unit norm separation vector $\mathbf{w}$, of an orthogonal separation matrix $\mathrm{W} = [\mathbf{w}_1,\ \mathbf{w}_2,\ \cdots,\ \mathbf{w}_M]$, such that it adopts the direction of growth of the negentropy in each iteration. Conveniently for this purpose, it is not necessary to precisely approximate the negentropy, being a proportional value enough (HYVäRINEN; KARHUNEN; OJA, 2001):

$$\mathrm{J}(y) \propto (\mathbb{E}[\mathrm{G}(y)] - \mathbb{E}[\mathrm{G}(n)])^2, \tag{3.11}$$

where $\mathrm{G}(\cdot)$ is a non-quadratic function, while $y = \mathbf{w}^T\mathbf{z}$ is a single unit-variance candidate estimate, and n is a unit-variance random variable with Gaussian distribution.

Then, after some simplifications and modifications to improve convergence, the fixed-point iteration of the gradients of $\mathrm{J}(y)$ with respect to $\mathbf{w}$ to a subsequent $\mathbf{w}$ is computed as:

$$\mathbf{w} \leftarrow \mathbb{E}[\mathbf{z}\mathrm{g}(\mathbf{w}^T\mathbf{z})] - \mathbb{E}[\mathrm{g}'(\mathbf{w}^T\mathbf{z})\mathbf{w}], \tag{3.12}$$

where $\mathrm{g}(\cdot)$ and $\mathrm{g}'(\cdot)$ are the first and second derivatives of $\mathrm{G}(\cdot)$, respectively. The separation vector $\mathbf{w}$ must be later normalized so it is always unit-norm.

In (HYVäRINEN; KARHUNEN; OJA, 2001), three possible choices for functions $\mathrm{g}(\cdot)$ and $\mathrm{g}'(\cdot)$ are presented and reproduced below:

$$\mathrm{g}(y) = \tanh(ay) \qquad \mathrm{g}'(y) = a(1 - \tanh(ay)^2) \tag{3.13}$$

$$\mathrm{g}(y) = y\exp(-y^2/2) \qquad \mathrm{g}'(y) = (1 - y^2)\exp(-y^2/2) \tag{3.14}$$

$$\mathrm{g}(y) = y^3 \qquad \mathrm{g}'(y) = 3y^2, \tag{3.15}$$

where a is a constant scalar to be selected between 1 and 2. Curiously, the last choice in (3.15) leads to gradients close to those related to the kurtosis criterion.

In order to estimate the complete separation matrix W, and thus retrieve all remaining sources, FastICA relies on the invariance of whitening for orthogonal transformations to ensure, at least, the uncorrelatedness of the estimates at each iteration. Then, for this purpose, FastICA employs two different methods: deflation-based and symmetric orthogonalization.

Deflation-based FastICA uses the Gram-Schmidt method to sequentially orthogonalize each separation vector $\mathbf{w}$ with respect to all the previously estimated vectors at each iteration step, thus constructing an orthogonal separation matrix W. The problem with this approach is the error accumulation and the greater relevance it confers to the first estimated vectors.

On the other hand, symmetric FastICA simultaneously initializes the set of separation vectors and, then, orthogonalizes the set with symmetric orthogonalization at each iteration step, such that no vector $\mathbf{w}$ is more important than the rest for the construction of the orthogonal separation matrix. Symmetric orthogonalization is accomplished through the following operation:

$$\mathrm{W} \leftarrow (\mathrm{W}\mathrm{W}^T)^{-1/2}\mathrm{W} \tag{3.16}$$

Finally, the symmetric FastICA algorithm may be expressed as:

Algorithm 1: Symmetric FastIca algorithm.

Result: Orthogonal separation matrix W.
Center and whiten the mixtures.
Indicate a number of M sources to estimate.
Initialize an M dimensional orthogonal separation matrix W.
while *Not converged* **do**
    Update the separation vectors in W according to (3.12).
    Orthogonalize W according to (3.16).
end

3.3 HIGHER-ORDER STATISTICS: CUMULANTS

Due to SOS alone not providing information beyond uncorrelatedness (Section 2.4.1), it becomes necessary to delve into Higher-Order Statistics (HOS) to find something more substantial about the independence of the involved signals. Particularly, cumulants offer helpful properties to evaluate non-Gaussianity and independence.

Cumulants are defined as the derivatives of the natural logarithmic of the moment generating function (also known as characteristic function) (PAPOULIS; PILLAI,

2002). Perhaps, most important of all for the context of BSS are the fourth-order cumulants, which are expressed as follows for a zero-mean L-component random vector $\mathbf{v}$:

$$\text{cum}(v_i, v_j, v_k, v_l) = \mathbb{E}[v_i v_j v_k v_l] - \mathbb{E}[v_i v_j]\mathbb{E}[v_k v_l] - \mathbb{E}[v_i v_k]\mathbb{E}[v_j v_l] - \mathbb{E}[v_i v_l]\mathbb{E}[v_j v_k], \quad (3.17)$$

where the four indices $i,\ j,\ k,\ l$ may assume any value from 1 to L.

A cumulant of a single repeated component, i.e., when all indices are equal, is referred to as an auto-cumulant. For example, the kurtosis expressed in (3.7) is a fourth-order auto-cumulant. Otherwise, it is a cross-cumulant, which is invariant to the order of its components and is always null when the involved components are independent. Therefore, considering a linear instantaneous ICA model, a separation matrix W should maximize the auto-cumulants and minimize the cross-cumulants.

CUMULANT TENSOR

Furthermore, the set of all L^4 fourth-order cumulants can be grouped into a 4-dimensional tensor $\mathrm{T}_{\mathbf{v}} \in \Re^{L \times L \times L \times L}$, such that each position in $\mathrm{T}_{\mathbf{v}}$ is assigned according to (3.17), i.e., $\mathrm{T}_{\mathbf{v}}(i,\ j,\ k,\ l) = \text{cum}(v_i,\ v_j,\ v_k,\ v_l)$. Then, the kurtosis of each component is placed in the diagonal, while the cross-cumulants are symmetrically placed outside of it.

It is possible to define a linear operation between a tensor and a matrix (similar to the multiplication between a matrix and a vector). As a consequence, an eigenmatrix can be defined as the matrix that cannot be linearly transformed by a tensor, only scaled by an eigenvalue (HYVäRINEN; KARHUNEN; OJA, 2001):

$$\mathrm{T}_{\mathbf{v}}(\mathrm{V}) = \lambda \mathrm{V}, \quad (3.18)$$

where $\mathrm{V} \in \Re^{L \times L}$ is a matrix linearly transformed by tensor $\mathrm{T}_{\mathbf{v}}$ into a scaled version of itself, or an eigenmatrix, while λ is the corresponding eigenvalue.

3.3.1 JOINT APPROXIMATED DIAGONALIZATION OF EIGENMATRICES

The Joint Approximated Diagonalization of Eigenmatrices (JADE) algorithm was proposed in (CARDOSO; SOULOUMIAC, 1993) with the purpose of finding a separation matrix W able to, simultaneously and indirectly, minimize the fourth-order cross-cumulants of the N-component estimate vector $\mathbf{y} = \mathrm{W}\mathbf{z}$, and maximize the kurtosis of each component, therefore steering towards independent estimates.

To accomplish this for the well-determined linear instantaneous ICA problem, JADE exploits the fact that the whitened mixing matrix $\mathrm{A}_{\mathbf{z}}$ (Section 2.5.1) diagonalizes every transformation of tensor $\mathrm{T}_{\mathbf{z}}$ on any matrix (CARDOSO; SOULOUMIAC, 1993).

JADE initially computes the N significant eigenmatrices V of tensor $\mathrm{T}_{\mathbf{z}}$, i.e., those with non-zero eigenvalues λ, which, interestingly, are also equal to the kurtosis. Then,

it aims at finding the orthogonal matrix that jointly diagonalizes all $\mathrm{T}_{\mathbf{z}}(\mathrm{V})$ through the maximization of the cost function $\mathrm{J}_{\mathrm{JADE}}(\cdot)$:

$$\mathrm{J}_{\mathrm{JADE}}(\mathrm{W}) = \sum_{i=1}^{N} \left\| \mathrm{diag}(\mathrm{W}(\mathrm{T}_{\mathbf{z}}(\mathrm{V}_i))\mathrm{W}^T) \right\|^2, \tag{3.19}$$

where $\|\mathrm{diag}(\cdot)\|^2$ refers to the squared euclidean norm of the diagonal.

Such a cost function is maximum when $\mathrm{W} = \mathrm{A}_{\mathbf{z}}^T$, up to a permutation and sign variation of its rows, thus leading to source separation.

Furthermore, for any orthogonal W, the value of (3.19) is equal to the sum of the squared fourth-order cumulants of $\mathbf{y}$, with at least repeated first and second components:

$$\mathrm{J}_{\mathrm{JADE}}(\mathrm{W}) = \sum_{\forall i,k,l} \mathrm{cum}(y_i,\ y_i,\ y_k,\ y_l)^2 \tag{3.20}$$

Therefore, considering the invariance of the sum of all the squared fourth-order cumulants for any orthogonal transformation on $\mathbf{z}$ (COMON; CARDOSO, 1990) (which equals the sum of all the squared kurtosis of the sources), the maximization of (3.20) indirectly maximizes the kurtosis of each component of $\mathbf{y}$ and minimizes the fourth-order cross-cumulants with no repeated components.

Considering an over-determined case, the JADE algorithm requires the knowledge of the number of sources M in order to previously reduce the dimensionality into a well-determined problem. JADE offers no solution to the underdetermined case.

Due to the complexity of the cost function, JADE requires a large number of samples to precisely estimate the cumulants and deliver satisfactory results. Additionally, as pointed out in (HYVäRINEN; KARHUNEN; OJA, 2001), it does not perform well when dealing with high-dimensional vectors.

For the maximization of the joint diagonalizer (3.19), a generalization of the Jacobi technique to matrix diagonalization together with Givens rotations is used in the original implementation made publicly available by the authors.

To summarize, the JADE algorithm may be very succinctly described as shown in Algorithm 2:

Algorithm 2: Joint Approximated Diagonalization of Eigenmatrices algorithm.

Result: Orthogonal separation matrix W.
Center and whiten the mixtures.
Compute the cumulants of $\mathbf{z}$ and form the cumulant tensor.
Compute the N most significant eigenmatrices.
maximize the cost function $\mathrm{J}_{\mathrm{JADE}}(\cdot)$ in (3.19) to find the separation matrix W.

3.4 SUMMARY

As already mentioned in Chapter 2, in order to solve the BSS problem it is necessary the definition of a criterion capturing the hypothesis made about the sources, such that its optimization yields representative estimates.

In this chapter, we have focused on some of the main criteria for the retrieval of independent estimates under ICA. First, the minimization of the mutual information to guarantee statistical independence among the estimates, which is followed by the Infomax principle. Then, the maximization of the non-Gaussianity as indicative of source separation and independence, either in the form of the kurtosis or negentropy, and finally, regarding fourth-order cumulants, the maximization of the auto-cumulants (or kurtosis), and the minimization of cross-cumulants, which indicate non-linear correlation when only two different components are present, though repeated, and independence once they are all null. Due to the well-known performance of the FastICA and JADE algorithms, we have selected them to illustrate the two later approaches to independence.

In the following chapter, we cover the key concepts of artificial neural networks and adversarial learning, which form the basis for ANICA.

4 AUTOENCODERS AND ADVERSARIAL NETWORKS

In the previous chapters, we have discussed the fundamentals of the BSS problem and of ICA, one of the most relevant approaches to solve it. Moreover, some of the quantities and strategies used to measure and attain independence, which is the key hypothesis behind ICA, have been outlined, as well as two well-known algorithms based on non-gaussianity and higher-order cumulants, viz., FastICA and JADE.

Now, in this chapter, we shift the focus of the exposition to the field of machine learning, especially to the topics of Autoencoders (AEs) and Generative Adversarial Networks (GANs) (GOODFELLOW et al., 2014), which together form the basis of the Adversarial Autoencoder (AAE) (MAKHZANI et al., 2016). These subjects are of great importance, as they establish the main ideas that support the proposal of ANICA (BRAKEL; BENGIO, 2017), which will be detailed in the next chapter. In order to better explain these concepts, a brief discussion on artificial neural networks is provided in the sequence.

4.1 ARTIFICIAL NEURAL NETWORKS

According to its initial conception, an Artificial Neural Network (ANN) is a computational structure loosely based on an idealization of how the neurons in the brain communicate and adapt in order to improve their performance on some specific task (ALPAYDIN, 2010; GéRON, 2019).

In a more practical sense, an ANN may be considered as an adaptable model that creates a mapping between input data and an output. It is composed of a fixed number of processing units, named artificial neurons, each defined by a parametric function, which are sequentially connected in layers according to some architecture (BISHOP, 2006; GOODFELLOW; BENGIO; COURVILLE, 2016). These parameters of the network must be adjusted by some training mechanism, typically a supervised framework in which the energy of the error is minimized. This adaptation process is carried out with the aid of iterative optimization algorithms.

Among the existing ANN architectures, perhaps the most classic and widely used option is the Multi-Layer Perceptron (MLP), due to its standardized architecture and its capacity for addressing different tasks, such as pattern classification, prediction and retrieval of latent variables, to cite a few. Nonetheless, other successful architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs)

are employed for different purposes in more challenging contexts, like natural language processing and computer vision (GOODFELLOW; BENGIO; COURVILLE, 2016; GéRON, 2019).

With respect to training, in brief, the parameters of an MLP are usually updated by methods based on the notion of gradient descent, such that the parameters are updated by following the opposite direction of that defined by the gradients of the cost function. The computation of such gradients is carried out with the aid of the backpropagation algorithm (RUMELHART; HINTON; WILLIAMS, 1986; WERBOS, 1982), which allows determining the derivatives of the cost function (whose definition directly involves the network output) with respect to the parameters of the inner layers. This brief explanation is also valid for the training process of CNNs and RNNs, though with some non-trivial adaptations, as explained in (GOODFELLOW; BENGIO; COURVILLE, 2016).

4.1.1 MULTI-LAYER PERCEPTRONS

An MLP is an architecture that organizes its computational units, also known as neurons or perceptrons, in a number of hidden layers and an output layer. In an MLP, all n^l neurons in a layer l typically share a common activation function $\mathrm{f}_a^l(\cdot)$, and the set of signals they generate $\mathbf{a}^l$ is transmitted as input to the next layer $l+1$. This results in a mapping $\mathrm{mlp}(\cdot)$ between the input set of signals $\mathbf{v}$ and the output projected in the last layer.

Typically, MLPs are fully-connected, which means that each layer l can be modelled by a $n^l \times n^{l-1}$ weight matrix Θ^l and a n^l-component bias vector b^l, where n^{l-1} is the number of neurons the previous layer holds. Therefore, in such MLPs, the output of a single layer can be computed as:

$$\mathbf{a}^l = \mathrm{f}_a^l(\Theta^l \mathbf{a}^{l-1} + b^l) \tag{4.1}$$

Example 4.1.1. Considering two hidden layers and an output layer with arbitrary activation functions, the final mapping between an input $\mathbf{v}$ and output $\mathbf{a}^3$, defined by a fully-connected MLP, may be computed by:

$$\left.\begin{array}{ll} \mathbf{a}^1 & = \mathrm{f}_a^1(\Theta^1 \mathbf{v} + b^1) \\ \mathbf{a}^2 & = \mathrm{f}_a^2(\Theta^2 \mathbf{a}^1 + b^2) \\ \mathbf{a}^3 & = \mathrm{f}_a^3(\Theta^3 \mathbf{a}^2 + b^3) \end{array}\right\} \mathbf{a}^3 = \mathrm{mlp}(\mathbf{v},\ \Theta), \tag{4.2}$$

where all Θ^l and b^l are trainable parameters, while $\mathrm{mlp}(\cdot)$ is the overall MLP mapping.

Figure 4.1 depicts the MLP defined in (4.2) and how the signals are transmitted from a three-component input to a two-component output. ■

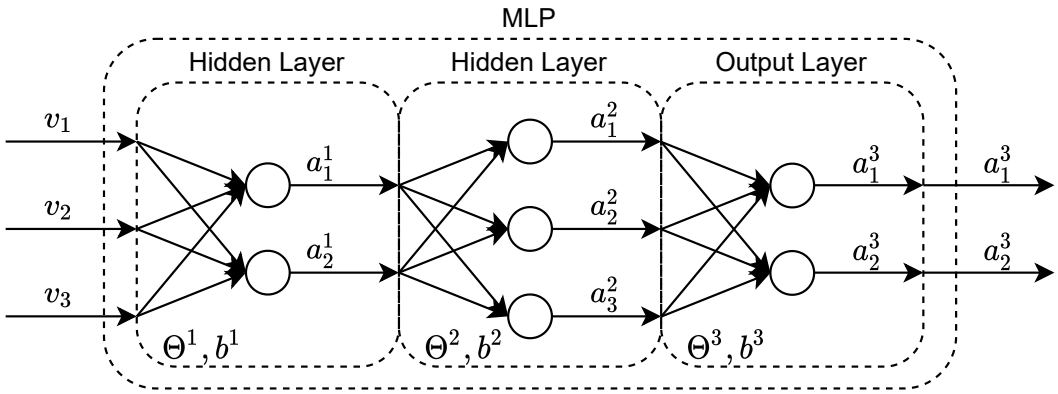


Figure 4.1 – Structure of a MLP with 3 inputs, 2 hidden layers and a two-component output layer.

CLASSIFICATION BY DISCRIMINATORS

As already mentioned, a common application of MLPs is in pattern classification. The samples of an input set of signals $\mathbf{v}$ may be divided into mutually exclusive groups, or classes c, according to some characteristics of interest they present, so each sample belongs to one class only. Then, classification is the task of learning to recognize these characteristics and, therefore, generate a class-prediction for unseen data (DUDA R. O. HART; STORK, 2000). For brevity of exposition, only two classes may be considered: a True and a False class, denoted by labels $c = 1$ and $c = 0$, respectively. In this case, a single output neuron is needed for the task, though additional output neurons could be added for a multi-class problem.

An MLP designed for classification is called a Discriminator and it is designated by the mapping $\mathrm{d}(\cdot)$. A discriminator must learn to map each, or most, of the input data samples into a conditional posterior probability of belonging to the True class, denoted by $\mathrm{P}_{\mathrm{True}|\mathbf{v}}(\cdot)$. In other words, for every pair of samples $(\mathbf{v}^i,\ c^i)$, a discriminator should output $\mathrm{d}(\mathbf{v}^i) = \mathrm{P}_{\mathrm{True}|\mathbf{v}}(c^i = 1 \mid \mathbf{v}^i)$ indicating the probability of True class membership, i.e., for a single sample $\mathbf{v}^i$ with label $c^i = 1$, the corresponding output sample $\mathrm{d}(\mathbf{v}^i)$ should be close to one, whereas, when the class of the input sample is $c^i = 0$, the output should be almost zero. In such a case, considering the output approximates the actual class labels, the discriminator has learnt to correctly classify the input samples.

In order to properly train a discriminator, a cost function is designed in terms of the mean (across all samples) of the cross-entropy between the Probability Mass Function (PMF) defined by the network mapping $\mathrm{P}_{\mathrm{map}}(\cdot)$ and the PMF defined by the labels $\mathrm{P}_{\mathrm{class}}(\cdot)$, which is denoted by $\mathrm{H}(\mathrm{P}_{\mathrm{class}},\ \mathrm{P}_{\mathrm{map}})$ and thus measures the dissimilarity between the class prediction and the actual class membership. Therefore, such a cost function $\mathrm{J}_{\mathrm{D}}(\cdot)$, to be

minimized during training, is expressed as:

$$\begin{aligned} \mathrm{J_D(D)} &= \mathbb{E}[\mathrm{H}(\mathrm{P_{class}},\ \mathrm{P_{map}})] \\ &= \frac{1}{N_s}\sum_{\forall i} -c^i \log(\mathrm{d}(\mathbf{v}^i)) - (1-c^i)\log(1-\mathrm{d}(\mathbf{v}^i)), \end{aligned} \tag{4.3}$$

where D refers to all weight matrices and bias vectors, and N_s is the number of samples.

Under this interpretation, every input sample $\mathbf{v}^i$ is associated to the two aforementioned PMFs, $\mathrm{P_{map}}(\cdot)$ and $\mathrm{P_{class}}(\cdot)$, which indicate the probability of any sample to belong to the True or False class and are expressed as follows:

$$\mathrm{P_{map}}(\mathrm{True}) = \mathrm{d}(\mathbf{v}^i) \qquad \mathrm{P_{map}}(\mathrm{False}) = 1 - \mathrm{d}(\mathbf{v}^i) \tag{4.4}$$

$$\mathrm{P_{class}}(\mathrm{True}) = c^i \qquad \mathrm{P_{class}}(\mathrm{False}) = 1 - c^i, \tag{4.5}$$

where $\mathrm{P_{class}}(\cdot)$ is fixed and deterministic as c^i can only be equal to 1 or 0.

Thus, if the discriminator learns a mapping PMF consistently similar to the class PMF, i.e., $\mathrm{P_{map}}(\cdot) \to \mathrm{P_{class}}(\cdot)$, the aforementioned cost function (4.3) is consequently minimized to zero as most of the individual cross-entropies are reduced to their corresponding class entropy, i.e., $\mathrm{H}(\mathrm{P_{class}},\ \mathrm{P_{map}}) = \mathrm{H}(\mathrm{P_{class}}) + \mathrm{D_{KL}}\left(\mathrm{P_{class}} \parallel \mathrm{P_{map}}\right) \to \mathrm{H}(\mathrm{P_{class}})$, whose value is equal to zero due to the deterministic nature of the class PMF.

4.2 AUTOENCODERS

The model known as an Autoencoder (AE) corresponds to an ANN trained to reconstruct the input data $\mathbf{v}$ at the output layer $\hat{\mathbf{v}}$ (GOODFELLOW; BENGIO; COURVILLE, 2016; GéRON, 2019). Although this may seem to be a trivial task, by doing so, an AE is able to capture a meaningful and latent representation of the data in the form of the output of one of its hidden layers. The retrieval of this latent representation, which is called a code $\mathbf{h}$, is the real aim of AEs, as this code should contain enough information to allow the following layers to reconstruct the input data into the final output layer.

The structure of an AE can be summarized as two functions in a sequence: (1) an encoder function $\mathrm{f_{en}}(\cdot)$, and (2) a decoder function $\mathrm{f_{de}}(\cdot)$, where the encoder creates the code $\mathbf{h} = \mathrm{f_{en}}(\mathbf{v})$ and the decoder outputs the reconstructed data $\hat{\mathbf{v}} = \mathrm{f_{de}}(\mathbf{h})$. Figure 4.2 depicts the general architecture of an AE, where the encoder projects an M-component code $\mathbf{h}$, while the decoder attempts to output an N-component reconstruction of the input $\mathbf{v}$. The encoder and decoder functions can be implemented by stacking an arbitrary number of hidden layers.

An AE may be interpreted as a latent variable model: the code $\mathbf{h}$ is a representation of the latent variables underlying the input data, whose usefulness can be confirmed by the data reconstruction it allows. In this context, we can recognize a conceptual connection

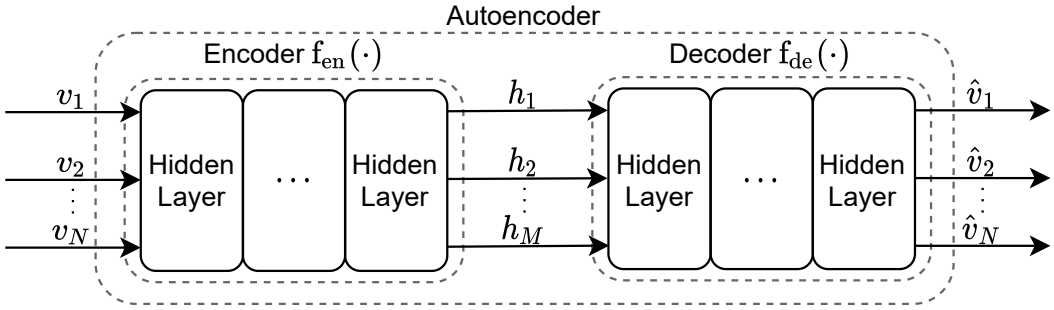


Figure 4.2 – General structure of an Autoencoder and its most important components.

between AEs and the BSS problem. The encoder must learn to act as a separation system $\mathcal{W}$, retrieving the latent variables that indeed generated the observed data. On the other hand, the decoder plays the role of a mixing system $\mathcal{A}$, since it must combine the latent variables in order to recover the original data.

This connection suggests that, given a set of observations of the mixtures, it may be possible for an AE to retrieve estimates of the sources at its internal code, since the decoder should be able to reconstruct the mixtures. Evidently, the assumptions of ICA (Section 2.5) should be satisfied by AEs as well.

AEs require the minimization of a cost function $\mathrm{J_{AE}}(\cdot)$ that expresses the quality of the reconstruction. A common, but far from unique, choice for such a function is the Mean Squared Error (MSE), which is expressed as:

$$\begin{aligned}\mathrm{J_{AE}}(\mathrm{F_{en}},\ \mathrm{F_{de}}) &= \mathbb{E}[\|\mathbf{v}-\hat{\mathbf{v}}\|^2] \\ &= \frac{1}{N_s}\sum_{\forall i}\left\|\mathbf{v}^i - \mathrm{f_{de}}(\mathrm{f_{en}}(\mathbf{v}^i))\right\|^2, \end{aligned} \tag{4.6}$$

where $\mathrm{F_{en}}$ and $\mathrm{F_{de}}$ encompass all weight matrices and bias vectors of the encoder and decoder, respectively, N_s is the number of samples, and $\|\cdot\|$ is the Euclidean norm.

Unfortunately, given enough capacity, the AE could simply learn a decoder function that inverts the previous encoder, or $\mathrm{f_{de}}(\cdot) = \mathrm{f_{en}^{-1}}(\cdot)$, rendering the generated code $\mathbf{h}$ irrelevant and uninformative about the data. However, it is possible to avoid these trivial results by considering some structural constraints or by introducing some form of regularization in order to prevent a perfect but useless reconstruction, for example, by limiting the number components of the code or by adding noise to the data (GOODFELLOW; BENGIO; COURVILLE, 2016).

Interestingly, even though AEs are trained in a supervised fashion with the purpose of minimizing an error measure between the input data and the reconstruted output, they learn to create the internal representation in an unsupervised manner, since there is no target information for the code. Therefore, AEs do represent a blind scheme for obtaining latent variables, which is in accordance with the spirit of the BSS problem.

4.2.1 LINEAR AUTOENCODERS AND PCA

An AE is said to be linear when both the encoder and decoder consists of a single hidden layer with a linear activation function. Thus, a Linear Autoencoder (LAE) can be modelled as:

$$\left.\begin{aligned}\mathbf{h} &= \mathrm{f}_{\mathrm{en}}(\mathbf{v}) &&= \Theta_{\mathrm{en}}\mathbf{v} + b_{\mathrm{en}} \\ \hat{\mathbf{v}} &= \mathrm{f}_{\mathrm{de}}(\mathbf{h}) &&= \Theta_{\mathrm{de}}\mathbf{h} + b_{\mathrm{de}}\end{aligned}\right\} \hat{\mathbf{v}} = \Theta_{\mathrm{de}}\Theta_{\mathrm{en}}\mathbf{v} + \Theta_{\mathrm{de}}b_{\mathrm{en}} + b_{\mathrm{de}} \tag{4.7}$$

From (4.7), it is evident that adding more linear layers to LAEs has no impact on improving reconstruction nor code retrieval, as this would be equivalent to replacing a number of weight matrices (or bias vectors) for a single one.

A common structural limitation imposed on AEs is to produce a code with fewer components than the input, i.e., $M < N$ as described in Figure 4.2, thus forcing the encoder to learn a dense and compact representation of the data. Such an AE is said to be undercomplete, and it may also be used for dimensionality reduction and visualization.

A well studied connection between AEs and PCA (Section 2.4.2) indicates that an undercomplete LAE, trained to minimize the MSE cost function (4.6), projects a code within the subspace spanned by PCA (GOODFELLOW; BENGIO; COURVILLE, 2016). This connection is grounded on the fact that PCA is able to reconstruct, by means of (2.27), its original input from the principal components it retrieves, which are computed under the restrictions of uncorrelatedness and an orthonormal basis, while minimizing the same MSE defined in (2.28) and attaining the same result (2.30). In other words, such a LAE and PCA have the same MSE minimization objective, based on the reconstruction of an original input from a representation with possibly fewer components. However, both differ in their results due to the flexibility of undercomplete LAEs, which are not necessarily under the same restrictions as PCA.

Example 4.2.1. In an attempt to better understand the connection between AEs and PCA, the Iris data set (DUA; GRAFF, 2017), which is composed of 150 observations of four geometrical characteristics (length and width of the sepal and petal) from three different species of irises (setosa, virginica and versicolor), was used to separately compute the two first principal components and the subsequent reconstruction through PCA, and to train an undercomplete LAE in order to retrieve a 2-component code from the aforementioned 4-component input.

After PCA and training the LAE to minimize the MSE cost function (4.6), the

resulting orthonormal basis U_R and weight matrices Θ_{en} and Θ_{de} are:

$$\mathrm{U}_R = \begin{bmatrix} -0.36 & -0.66 \\ +0.08 & -0.73 \\ -0.86 & +0.17 \\ -0.36 & +0.08 \end{bmatrix}$$
$$\Theta_{\text{en}} = \begin{bmatrix} -0.73 & -0.47 & -0.46 & -0.19 \\ +0.19 & +0.56 & -0.74 & -0.31 \end{bmatrix} \qquad \Theta_{\text{de}} = \begin{bmatrix} -0.73 & +0.19 \\ -0.47 & +0.56 \\ -0.46 & -0.74 \\ -0.19 & -0.31 \end{bmatrix}, \tag{4.8}$$

where all matrices are rounded up to two decimals. No bias vectors were considered.

From (4.8), it is noticeable that the encoder and decoder share the same parameters. This is accomplished by a technique called Tying weights (GéRON, 2019), which basically forces the parameters of the decoder to be the transpose of the parameters of the encoder, i.e., $\Theta_{\text{de}} = \Theta_{\text{en}}^T$, which leads towards learning an encoder with parameters closer to an orthonormal basis, i.e., $\Theta_{\text{en}}\Theta_{\text{en}}^T \rightarrow \mathrm{I}$, due to the reconstruction objective. Furthermore, since every rotation Q of the orthonormal basis U_R (considering eigenvectors with possibly different signs) yields the same minimum reconstruction error as PCA (DIAMANTARAS; KUNG, 1996), then, the rows of the encoder lie almost perfectly within the subspace defined by PCA, i.e., $\Theta_{\text{en}} \approx \mathrm{QU}_R^T$, thus proving that the code is within the same subspace.

Figure 4.3 shows that the code produced by the LAE (in red) coincides with the two first principal components (in blue) after a vertical reflection, i.e., with respect to $\mathbf{u}_2$, and an approximately 46.57° counterclockwise rotation. Moreover, the components of the code are clearly not uncorrelated, as it was not explicitly considered during training nor the design of the LAE. ■

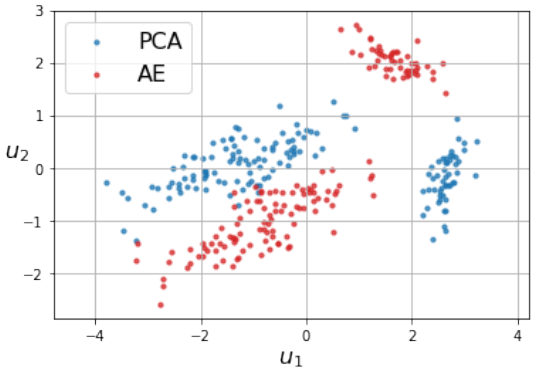


Figure 4.3 – Connection between undercomplete LAEs and PCA.

Aditionally, as also indicated in (GOODFELLOW; BENGIO; COURVILLE, 2016), undercomplete non-linear AEs, i.e., when the encoder and decoder do not correspond

to linear functions, also trained to minimize the MSE cost function, may learn a code within the subspace defined by a non-linear generalization of PCA. Perhaps most importantly of all, AEs in general have the capability to go beyond PCA given the right restrictions and training.

Other than limiting the number of components in the code layer, as in under-complete AEs, it is possible to avoid trivial solutions by means of regularization, i.e., by guiding the AE training towards a desired relevant code, as in the following cases.

4.2.2 DENOISING AUTOENCODERS

Originally proposed in (VINCENT et al., 2008; VINCENT et al., 2010), Denoising Autoencoders (DAEs) try to reconstruct an original data set after it has been corrupted (or partially destroyed) by some random process, thus capturing a code more robust to small perturbations in the data.

The original data $\mathbf{v}$ is corrupted by either adding some random noise, which is usually Gaussian, or by randomly selecting some of its components to be replaced with zeros, instead of eliminating them, to preserve the same dimensionality. Thus, the newly corrupted vector $\tilde{\mathbf{v}}$ becomes the input of the DAE, while the original vector $\mathbf{v}$ remains as the reconstruction target. Hence, the decoder cannot learn a trivial mapping, i.e., $\mathrm{f}_{\mathrm{de}}(\cdot) \neq \mathrm{f}_{\mathrm{en}}^{-1}(\cdot)$, as $\mathbf{v} \neq \tilde{\mathbf{v}}$, and, thus, the code $\mathbf{h} = \mathrm{f}_{\mathrm{en}}(\tilde{\mathbf{v}})$ is forced to learn the most relevant information contained in $\tilde{\mathbf{v}}$ to recover the partially unseen original data onto the output layer, i.e., $\hat{\mathbf{v}} = \mathrm{f}_{\mathrm{de}}(\mathrm{f}_{\mathrm{en}}(\tilde{\mathbf{v}}))$. Importantly, the random corruption process by which $\mathbf{v}$ turns into $\tilde{\mathbf{v}}$ is only useful during training.

4.2.3 SPARSE AUTOENCODERS

A different approach towards finding a meaningful code consists in imposing an sparse distribution on the code layer, by means of adding an explicit regularization cost to the original recunstruction cost function, thus allowing only a few active code neurons at each time (due to sparsity), forcing them to only pick up relevant information. These Sparse Autoencoders (SAEs) may also be useful for feature extraction, as the trained sparse code may be later used for classification tasks.

Typically, the sparsity regularization cost to be minimized is based on the l_1 norm of the samples of the code. Furthermore, according to (GOODFELLOW; BENGIO; COURVILLE, 2016), the minimization of such cost guides the code towards a Laplacian distribution (with known parameters), which, subsequently, turns the SAE into a generative model, as it would be possible to randomly generate new code samples from the known distribution.

Additionally, (GéRON, 2019) indicates that sparsity may also be attainable

by selecting adequate activation functions in the code layer, such as the sigmoid function that restricts the values of the code layer to a limited range.

4.2.4 VARIATIONAL AUTOENCODERS

In a nutshell, Variational Autoencoders (VAEs) (KINGMA; WELLING, 2014) reconstruct their inputs $\mathbf{v}$ from a random code $\mathbf{h}$. This code is typically sampled from a Gaussian distribution, although other distributions may be employed, whose parameters are the actual outputs the encoder must learn. Finally, a conventional decoder attempts to yield a reconstruction $\hat{\mathbf{v}}$ from such randomly sampled codes. Figure 4.4 depicts the main components of a VAE, while the sophisticated mechanism it follows is expressed as:

$$\begin{aligned} (\boldsymbol{\mu},\ \boldsymbol{\sigma}) &= \mathrm{f}_{\mathrm{en}}(\mathbf{v}) \\ \mathbf{h}^i &\sim \mathcal{N}(\boldsymbol{\mu}^i,\ \boldsymbol{\sigma}^i) \\ \hat{\mathbf{v}} &= \mathrm{f}_{\mathrm{de}}(\mathbf{h}), \end{aligned} \tag{4.9}$$

where $\boldsymbol{\mu}^i,\ \boldsymbol{\sigma}^i,\ \mathbf{h}^i \in \Re^M$ correspond to the sample mean, standard deviation and code vectors, respectively.

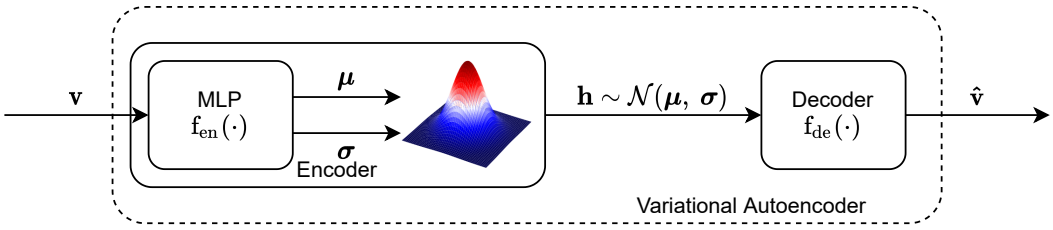


Figure 4.4 – Structure and components of a Variational Autoencoder.

From (4.9), it is clear that the vector samples $(\boldsymbol{\mu}^i,\ \boldsymbol{\sigma}^i)$ of $(\boldsymbol{\mu},\ \boldsymbol{\sigma})$ are initially different, as the N_s samples of the input $\mathbf{v}$ are different as well, and, thus, such initial parameters do not define a common PDF to the code space, although a distribution is chosen from the beginning.

Therefore, the VAE cost function to be minimized must include an additional latent cost, which is the responsible for pushing the codes towards a common and pre-specified PDF. Such latent cost is based on the Kullback-Leibler divergence between a Gaussian PDF with desired parameters and the PDF defined by the parameters computed by the encoder for an input sample. Fortunately, as indicated in (KINGMA; WELLING, 2014) for a pre-specified $\mathcal{N}(0,\ \mathrm{I})$ PDF, the latent loss $\mathcal{L}_{\mathrm{VAE}}^{\mathrm{L}}$ of a single sample may be easily expressed as:

$$\mathcal{L}_{\mathrm{VAE}}^{\mathrm{L}} = -\frac{1}{2}\sum_{\forall j} 1 + \log(\sigma_j^2) - \sigma_j^2 - \mu_j^2, \tag{4.10}$$

where j refers to any of the M components.

As indicated in (GéRON, 2019), VAEs are probabilistic models due to the inherit uncertainty in the code, so that a single input may have many slightly different reconstructions. Furthermore and most importantly, VAEs are generative models: once the encoder learns the necessary parameters, it is possible to sample a new code from the known Gaussian (or any other selected) distribution and, through the decoder, generate a new data sample.

4.3 GENERATIVE ADVERSARIAL NETWORKS

Similarly to VAEs, GANs are also generative models that produce new data from the inherit randomness within their mechanism. However, GANs originate from a substantially different adversarial training approach which provides a far superior "realness" to their results.

Based on the simple idea of a min-max two-player game played between a team of counterfeiters and the police, where the former are attempting to produce some undetectable fake currency while the latter is trying to improve their ability to recognize which currency is real and which is fake, Ian Goodfellow et al. proposed in (2014) a groundbreaking adversarial framework able to train two competing networks in such a fashion that the whole system produces some fake output data virtually indistinguishable from the real input data. This final system is called a Generative Adversarial Network (GAN) and it is composed of a Generator network, mimicking the counterfeiters, and a Discriminator, acting as the police.

The Generator is an ANN whose purpose is to produce output samples that could belong within the real data of interest. To achieve this, the generator transforms an input random vector $\mathbf{n} \sim p_{\mathbf{n}}$, drawn from a given distribution (usually Gaussian), into an output $g(\mathbf{n})$, through a mapping $g(\cdot)$ that implicitly imposes an unknown distribution $p_{\mathbf{g}}$, such that the output samples should resemble those of the data of interest. Hence, it is a generative model.

Furthermore, the samples of $g(\mathbf{n}) \sim p_{\mathbf{g}}$ are not meant to imitate the samples of the real data vector $\mathbf{r} \sim p_{\mathbf{r}}$, which is the input of the GAN, as this would not lead towards the generation of new "real" samples. Instead, the generated distribution $p_{\mathbf{g}}$ must converge towards the, presumably also unknown, real distribution $p_{\mathbf{r}}$. Thus, both generated and real samples become indistinguishable and, yet, different from each other.

On the other hand, the discriminator is meant to classify among the generated and real samples that compose its input vector $\mathbf{u}$, thus deciding whether the generator outputs plausible samples or not. In order to clarify how these components are connected, Figure 4.5 shows the general structure of a GAN, where the generated samples of $g(\mathbf{n})$ correspond to the desired output once the GAN is successfully trained.

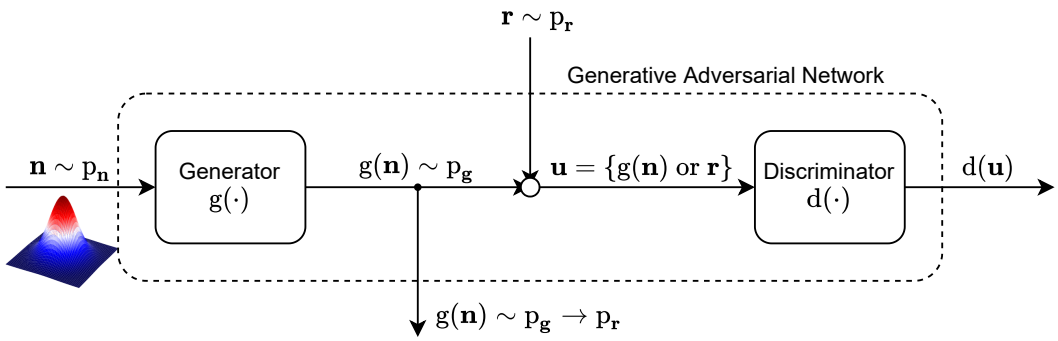


Figure 4.5 – Structure and components of a Generative Adversarial Network.

As originally devised, the GAN cost function $\mathrm{J_{GAN}}(\cdot)$ should be optimized with respect to the mappings of both components (GOODFELLOW et al., 2014). Nonetheless, such optimization can be performed on the parameters G and D of the generator and discriminator, respectively. Thus, the training process can be formulated as follows:

$$\begin{aligned}\min_{\mathrm{G}} \max_{\mathrm{D}} \mathrm{J_{GAN}}(\mathrm{D},\ \mathrm{G}) &= \mathbb{E}[\log(\mathrm{d}(\mathbf{r}))] + \mathbb{E}[\log(1 - \mathrm{d}(\mathrm{g}(\mathbf{n})))] \\ &= \frac{1}{N_s} \sum_{\forall i} \log\left(\mathrm{d}(\mathbf{r}^i)\right) + \log\left(1 - \mathrm{d}(\mathrm{g}(\mathbf{n}^i))\right), \end{aligned} \tag{4.11}$$

where N_s refers to the number of real or generated samples, as they are originally equal.

An iterative procedure based on two consecutive steps is explored to solve (4.11): First, the discriminator must minimize its cross-entropy cost function (4.3), which is equivalent to the maximization objective in (4.11). Second, the generator must adapt its own parameters to generate outputs such that the discriminator misclassify them by considering they are from the True class.

During the first step, the samples from the real data vector $\mathbf{r}$ are marked as belonging to the True class (label 1), whereas the samples from the generator are considered to be from the False class (label 0). However, a crucial difference occurs in the second step, where the membership of the generated samples is changed to the True class (labels changing accordingly) and they are the only input data presented to the discriminator, as the generator cannot modify the first summation in (4.11). During this second training step, the parameters of the discriminator must remain fixed and the gradients of $\mathrm{J_{GAN}}(\cdot)$ are backpropagated into the generator. Algorithm 3 summarizes this iterative procedure.

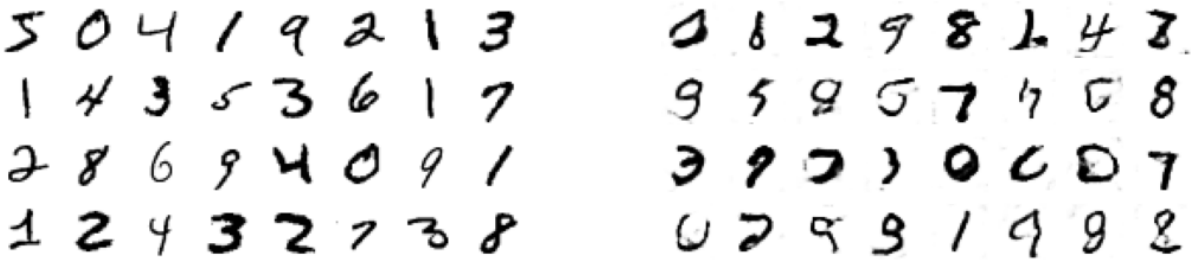
Example 4.3.1. As implemented in (GéRON, 2019) considering the aforementioned algorithm, a Deep Convolutional GAN (DCGAN) (RADFORD; METZ; CHINTALA, 2016) has been trained on the MNIST (LECUN; CORTES; BURGES, 2010) dataset.

Figure 4.6 shows the results after only 50 training epochs. Figure 4.6b demonstrates that the DCGAN is quite able to generate new images with, though not completely,

Algorithm 3: Generative Adversarial Networks algorithm.

Result: Generated samples indistinguishable from the real data.
Initialize all the parameters of the generator and discriminator networks.
for *all training epochs* **do**
  Draw K minibatches r^k of L samples from the real dataset.
  for *each minibatch* r^k **do**
    Step 1: Train the discriminator.
    Draw L samples $n^l \sim \mathrm{p}_{\mathbf{n}}$ and produce a minibatch g^k of generated samples.
    Assemble a minibatch u^k, such that: $\mathrm{u}^k \leftarrow \{\mathrm{r}^k(\text{label 1}),\ \mathrm{g}^k(\text{label 0})\}$.
    Update D to decrease the cross-entropy cost of $\mathrm{d}(\mathrm{u}^k)$.
    Step 2: Train the generator.
    Draw new L samples $n^l \sim \mathrm{p}_{\mathbf{n}}$ and produce a new generated minibatch g^k.
    Assemble a new minibatch u^k, such that: $\mathrm{u}^k \leftarrow \{\mathrm{g}^k(\text{label 1})\}$.
    Update G to decrease the cross-entropy cost of $\mathrm{d}(\mathrm{u}^k)$.
  end
end

convincing hand-written digits that are not mere copies of the original set, which is shown in Figure 4.6a, but that such generated images capture the style of the original set and present some odd details that could well belong to an actual person. ■



(a) Original MNIST samples. (b) Generated MNIST samples.

Figure 4.6 – DCGAN on MNIST.

Training is complete once the training objective (4.11) has converged. An analysis of convergence is also offered in (GOODFELLOW et al., 2014), where the authors show that convergence is attained if and only if the generator learns the actual distribution of the data, i.e., $\mathrm{p}_{\mathbf{g}}(\cdot) = \mathrm{p}_{\mathbf{r}}(\cdot)$, thus producing "new" real samples. In this context, the cost function remains stable around the constant value of $\log(4)$, i.e., $\mathrm{J}_{\mathrm{GAN}}(\mathrm{D},\ \mathrm{G}) \approx \log(4)$, and the discriminator, albeit being very competent, can no longer distinguish between real and fake samples, yielding an output closer to 0.5 for each one, i.e., $\mathrm{d}(\mathbf{u}) \approx 0.5$.

Nonetheless, in (GOODFELLOW, 2017), it is alerted that the aforementioned convergence analysis is only valid when (4.11) is optimized in the "function space", rather than in the "parameter space", as it is usually the case. Additionally, it also indicates that convergence implies the Nash equilibrium of the GAN game, i.e., neither generator nor discriminator benefit from changing their strategies.

Unfortunately, in practice, such equilibrium is not always attainable, as the cost function may end up indefinitely oscillating with no stable value. One phenomenon related to non-convergence of GANs is called mode collapse, which corresponds to the situation where the generator only outputs samples too similar to each other or from a single subset (class) of patterns with common characteristics.

Once the GAN is successfully trained, its generator resembles a latent model, whose input noise $\mathbf{n}$ may be seen as a code for the generated "real" data. Then, the generator and the decoder of an AE share a common behaviour (GéRON, 2019), which will be relevant for the next chapter.

4.4 ADVERSARIAL AUTOENCODERS

Having in view the structure of VAEs, with the idea of attaining a pre-specified PDF at the code layer, and the adversarial training mechanism of GANs, Alireza Makhzani et al. proposed in (2016) an innovative, simple architecture for generative AEs, which they called Adversarial Autoencoders (AAEs).

Similarly to VAEs, AAEs aim at driving the code space towards a chosen target distribution $\mathrm{p}_{\mathbf{t}}(\cdot)$, such that a new code sample $\mathbf{h}^n$ could be drawn from such distribution, i.e., $\mathbf{h}^n \sim \mathrm{p}_{\mathbf{t}}$, thus generating, through the decoder, a new "reconstructed" data sample $\hat{\mathbf{v}}^n = \mathrm{f}_{\mathrm{de}}(\mathbf{h}^n)$. However, the code space is not "pushed" towards the target distribution by the minimization of a regularization latent cost added to the reconstruction cost function, as the Kullback-Leibler divergence latent loss in (4.10), but, it is guided by a discriminator in an adversarial scheme.

Analogously to GANs, AAEs set a competition between the codes $\mathbf{h}$ produced by the encoder, which acts as the generator network and, also implicitly, imposes an overall distribution $\mathrm{p}_{\mathbf{h}}(\cdot)$ on the codes, i.e., $\mathbf{h} \sim \mathrm{p}_{\mathbf{h}}$, and the samples, collected in $\mathbf{r}$, drawn by the chosen target distribution $\mathrm{p}_{\mathbf{t}}$, which behaves similarly to the real data input. Then, a discriminator must judge whether the samples come from the target distribution (True class) or the encoder (False class). Following this analogy, once the discriminator can no longer recognize the membership of the samples, the encoder has learnt to match the distribution of the codes to the target distribution ($\mathrm{p}_{\mathbf{h}}(\cdot) = \mathrm{p}_{\mathbf{t}}(\cdot)$).

Figure shows 4.7 the basic architecture proposed by AAEs with a bivariate Gaussian PDF $\mathcal{N}(0, \mathrm{I})$ as the chosen target PDF $\mathrm{p}_{\mathbf{t}}(\cdot)$.

Although no specific cost function is originally proposed in (MAKHZANI et al., 2016), a general training scheme, involving two phases, is outlined: first, a reconstruction phase, during which a previously defined reconstruction cost function, e.g., the MSE cost function (4.6), is minimized with respect to the parameters of the encoder F_{en} and

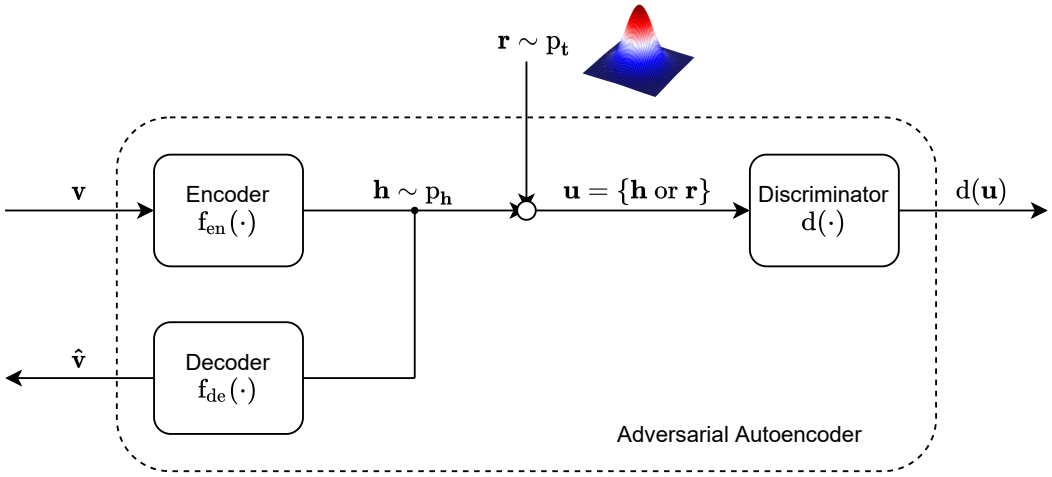


Figure 4.7 – Structure and components of an Adversarial Autoencoder.

decoder $\mathrm{F_{de}}$. Second, a regularization phase, which resembles the two-step training of GANs detailed in the previous section, such that the discriminator first learns to correctly classify among samples and, then, only the encoder updates its parameters in order to fool the discriminator. Algorithm 4 attempts to summarize the aforementioned training phases.

Algorithm 4: Adversarial Autoencoders algorithm.

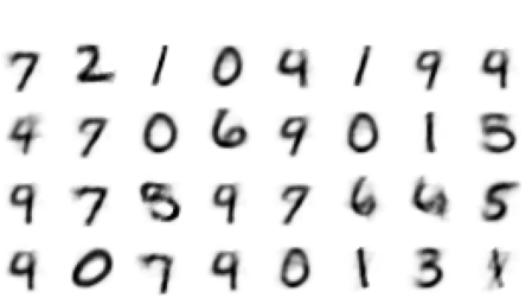
Result: Code space shaped by the target distribution.
Initialize all the parameters of the encoder, decoder and discriminator networks.
for *all training epochs* **do**
  Draw K minibatches v^k of L samples from the input dataset.
  for *each minibatch* v^k **do**
    Phase 1: Train the autoencoder.
    Update $\mathrm{F_{en}}$ and $\mathrm{F_{de}}$ to decrease a reconstruction cost.
    Phase 2 - Step 1: Train the discriminator.
    Compute a minibatch h^k of L code samples.
    Draw L samples $r^l \sim \mathrm{p_t}$ and compose a minibatch r_k.
    Assemble a minibatch u^k, such that: $\mathrm{u}^k \leftarrow \{\mathrm{r}^k(\text{label 1}),\ \mathrm{h}^k(\text{label 0})\}$.
    Update D to decrease the cross-entropy cost of $\mathrm{d}(\mathrm{u}^k)$.
    Phase 2 - Step 2: Train the encoder.
    Assemble a new minibatch u^k, such that: $\mathrm{u}^k \leftarrow \{\mathrm{h}^k(\text{label 1})\}$.
    Update $\mathrm{F_{en}}$ to decrease the cross-entropy cost of $\mathrm{d}(\mathrm{u}^k)$.
  end
end

As reported in (MAKHZANI et al., 2016), AAEs are a powerful and versatile framework that yield astounding results for different tasks, with minor architectural modifications, such as: disentangling style and content of images (in both a supervised and semisupervised fashion), unsupervised clustering and dimensionality reduction.

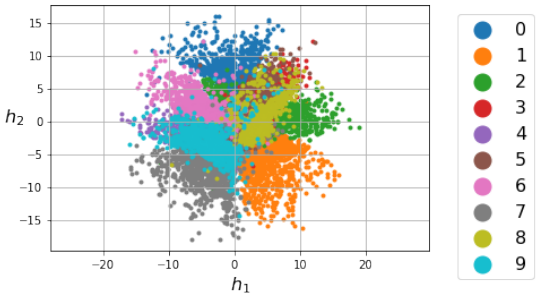
Example 4.4.1. In order to show that AAEs successfully impose a target PDF to the

latent space of the codes, an stochastic AAE has been implemented (similarly to the VAE implemented in (GéRON, 2019)), such that its 2-component code space is visible.

Figure 4.8 shows the results after only 100 training epochs on the MNIST dataset. In Figure 4.8b it is clear that the latent code space tends towards a pre-defined target Gaussian $\mathcal{N}(0, 5\mathrm{I})$. However, such space is not well partitioned with respect to the classes, as many code samples from different classes overlap, as occurs, for example with classes 4 (in purple) and 9 (in cyan). A reason for this can be found in Figure 4.8a, where the overlapping classes correspond to digits that the AE cannot reconstruct differently enough. Consequently, the same overlapping is seen in Figure 4.8c, for which new noise samples have been drawn from the target PDF to, subsequently, "reconstruct" new digits.



(a) Reconstructed MNIST samples.



(b) Latent code space.

(c) "Reconstructions" from noise samples.

Figure 4.8 – AAE on the MNIST dataset.

Importantly, an extensive search of hyperparameters, as well as more training epochs, could well deliver results closer to those reported in (MAKHZANI et al., 2016). ■

Interestingly, from a structural standpoint, AAEs are closely related to ANICA – which is the main focus of this study –, as both are AEs which attempt to retrieve relevant codes by resorting to an adversarial training scheme. However, an important distinction arises when we consider the target distribution: in the context of BSS we are not interested in matching any arbitrary distribution in the code layer, but we explicitly seek to attain the statistical independence between the latent variables, as they are meant to be estimates of the independent sources underlying the available mixtures.

4.5 SUMMARY

This chapter has been dedicated to outline the most relevant topics of machine learning with respect to latent variable retrieval. More specifically, we described how AEs represent non-linear generalizations of PCA and, thus, may go beyond obtaining uncorrelated variables, albeit regularization techniques may be required to guide the internal code towards statistical independence.

In this sense, the adversarial training scheme of GANs proved to be a powerful regularization technique for training AAEs in order to match a previously specified PDF. Hence, it seems plausible to speculate that such training scheme may also be useful to finally allow the retrieval of independent components.

In the following chapter, we will present ANICA, which, similarly to AAEs, includes an adversarial training scheme to shape the distribution of the codes. However, ANICA does not employ a pre-specified distribution for this purpose, but a resampling procedure that simulates samples from the independent distribution of the codes, thus guiding them towards independence.

5 AN ADVERSARIAL APPROACH TOWARDS ICA

In this chapter, we present the adversarial training approach, proposed by Brakel and Bengio, called Adversarial Non-linear Independent Component Analysis (ANICA) to retrieve statistically independent variables within the internal code generated by an AE.

First, we present the general mechanism behind ANICA and, with more detail, the component that differentiates it from regular AAEs: the resampling procedure. Then, we explain its adversarial algorithm and the conditions for convergence. Finally, we briefly discuss the kind of ICA problems it attempts to solve.

5.1 ANICA: ADVERSARIAL NON-LINEAR ICA

In (2017), Philemon Brakel and Yoshua Bengio proposed an adversarial algorithm, which they called ANICA, with the purpose of "teaching" an AE to retrieve, in its code layer, the statistically independent estimates expected by ICA from the available mixture samples the encoder receives. Hence, ANICA offers an alternative solution to the BSS problem for some specific contexts, under a framework similar to AAEs (MAKHZANI et al., 2016), but with some relevant differences.

Figure 5.1 depicts the general structure of ANICA and its most important components. A whitened mixture vector $\mathbf{z}$ is the input, while the encoder must create a code $\mathbf{h}$ such that its PDF tends to be equal to the product of the marginals.

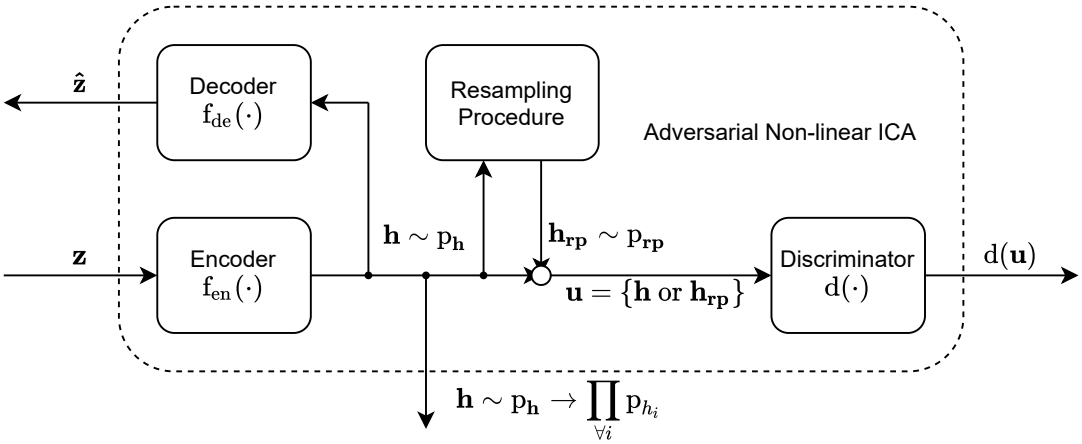


Figure 5.1 – Structure and components of ANICA.

Similarly to AAEs, ANICA consists of an AE whose encoder is in an adversarial game with a discriminator, which must decide whether the samples it receives come from

the encoder or from a target distribution. However, unlike AAEs, whose target distribution is known and fixed from the beginning, or GANs (GOODFELLOW et al., 2014), where the set of real samples does not vary, in ANICA the aforementioned distribution amounts to the joint distribution of the code variables when they are statistically independent, which is equivalent to the product of the marginal distributions. So, the idea of ANICA is to guide the encoder towards creating samples whose joint distribution progressively becomes more similar to the corresponding independent distribution.

The simulation of samples from the unknown independent distribution is at the core of ANICA, as they are completely necessary for its adversarial approach and the subsequent retrieval of independent codes. Unfortunately, in the BSS problem, none of the distributions is known and we do not have access to source samples that could be compared to the codes. Nevertheless, a relatively simple resampling procedure is able to perform such simulation.

5.1.1 RESAMPLING PROCEDURE

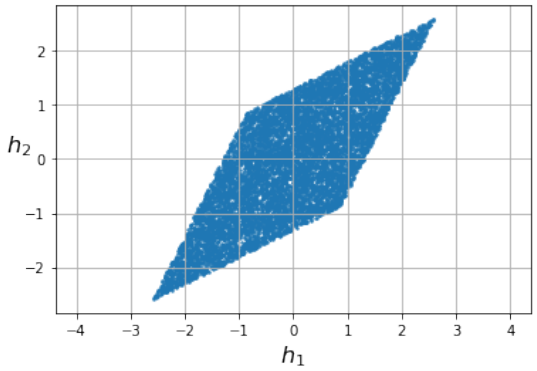
As presented in (BRAKEL; BENGIO, 2017), the resampling procedure attempts to simulate samples from the independent PDF, denoted by $\mathrm{p}_{\mathbf{rp}}(\cdot)$, corresponding to the product of the unknown marginal PDFs of an M-component random code vector $\mathbf{h} \sim \mathrm{p}_{\mathbf{h}}$, without trying to estimate any of the individual PDFs.

In order to do so, the resampling procedure simply reshuffles the set of samples of each random component $h_i \sim \mathrm{p}_{h_i}$ of $\mathbf{h}$, but separately and randomly, in an attempt to break any existing statistical dependence among individual samples across different components, while preserving each marginal PDF $\mathrm{p}_{h_i}(\cdot)$. Thus, the resulting joint samples, which are collected in the M-component random vector $\mathbf{h}_{\mathbf{rp}} \sim \mathrm{p}_{\mathbf{rp}}$, simulate those that could be drawn from the product of the marginals, i.e., $\mathrm{p}_{\mathbf{rp}}(\cdot) = \prod_i^M \mathrm{p}_{h_i}(\cdot)$.

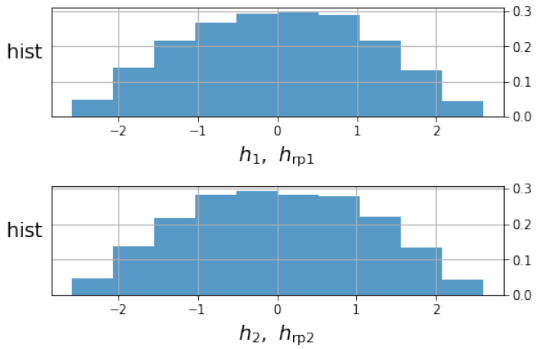
Importantly, the resampling procedure is performed at each training iteration, after the encoder has projected a corresponding new set of codes. Hence, the independent PDF $\mathrm{p}_{\mathbf{rp}}(\cdot)$ changes according to the new marginals, instead of being a fixed imposition on them.

Example 5.1.1. For illustrative purposes only, Figure 5.2 shows the effects of the resampling procedure on a 2-component vector $\mathbf{h}$ produced by a mixture of samples drawn from two independent random uniform distributions. Figure 5.2a indicates that the samples of such vector are correlated, whereas Figure 5.2c shows the resulting set of joint samples from vector $\mathbf{h}_{\mathbf{rp}}$, after the resampling procedure. By comparing the histograms along each component, in Figure 5.2b, to the joint histogram of $\mathbf{h}_{\mathbf{rp}}$, in Figure 5.2d, it is possible to verify that the joint density tends to the product of the marginals. ■

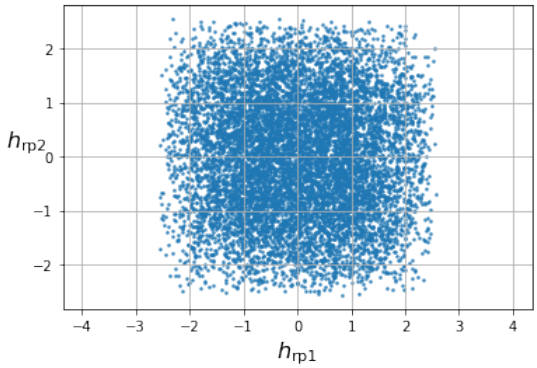
Another strategy is proposed in (BRAKEL; BENGIO, 2017). In simple terms,



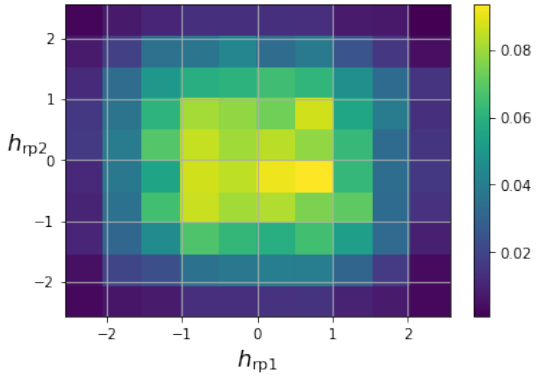
(a) 2-component code.



(b) Histogram of each component.



(c) Resampled code.



(d) Joint histogram after resampling.

Figure 5.2 – Resampling procedure and its effect on statistical dependence.

it consists of an array of M separate one-to-one generator networks $g_i(\cdot)$, such that each receives a different one-component input signal v_i, from a selected random distribution, thus, producing an array of M one-component outputs $g_i(v_i)$, which should be mutually statistically independent. However, such alternative is not explored in this dissertation, as it involves a higher computational cost and the reported results are not consistently better than those achieved by the resampling procedure. Still, as the authors indicate, it could be of interest when dealing with temporal structures in the mixtures or to attempt to turn ANICA into a generative model.

5.2 GENERAL ALGORITHM AND CONVERGENCE ANALYSIS

Despite the differences presented in the previous section, the adversarial training algorithm followed by ANICA remains similar to that of AAEs. However, in ANICA, the discriminator is initially trained to predict a True class membership for the samples from the resampling procedure and a False one for those from the encoder. Then, the AE is trained to minimize a cost function $J_{AE}(\cdot)$, which comprises a reconstruction and an adversarial regularization term. Such cost function, as implemented by the authors, may

be expressed as:

$$\begin{aligned} \mathrm{J}_{\mathrm{AE}}(\mathrm{F}_{\mathrm{en}},\ \mathrm{F}_{\mathrm{de}}) &= \mathbb{E}[|\mathbf{z}-\hat{\mathbf{z}}|] + \lambda\mathbb{E}[-\log(\mathrm{d}(\mathbf{h}))] \\ &= \frac{1}{N_s}\sum_{\forall i}\left|z^i - \mathrm{f}_{\mathrm{de}}(\mathrm{f}_{\mathrm{en}}(z^i))\right| - \lambda\log(\mathrm{d}(\mathrm{f}_{\mathrm{en}}(z^i))), \end{aligned} \tag{5.1}$$

where $|\cdot|$ is the l_1 norm function, λ is a constant whose default value is equal to 0.1, and N_s refers to the number of samples.

In (5.1), the adversarial regularization term, i.e., $\mathrm{J}^{\mathrm{adv}}_{\mathrm{AE}}(\mathrm{F}_{\mathrm{en}}) = \mathbb{E}[-\log(\mathrm{d}(\mathbf{h}))]$, is only dependent on the encoder and clearly represents its attempt to fool the discriminator, as it simply is the cross-entropy cost computed considering that the code samples belong to the True class (label 1). On a side note, the reconstruction cost $\mathrm{J}^{\mathrm{rec}}_{\mathrm{AE}}(\mathrm{F}_{\mathrm{en}},\ \mathrm{F}_{\mathrm{de}}) = \mathbb{E}[|\mathbf{z}-\hat{\mathbf{z}}|]$ was initially tested considering an Euclidean norm, but it did not show any experimental improvement relative to the l_1 norm selected by the authors.

It is important to mention that in ANICA the set of retrieved code samples should be normalized immediately in order to produce a normalized set after the resampling procedure, which helps the training process of the discriminator. In (BRAKEL; BENGIO, 2017), the authors also indicate the necessity of a form of Batch Normalization (IOFFE; SZEGEDY, 2015) on the codes, before reconstruction, to avoid trivial solutions. However, such technique showed no impact after convergence.

Algorithm 5 summarizes the main steps required to train ANICA, where a whitening transformation, when possible, is considered to speed up the process. In the overdetermined ICA case, it could be replaced by a simple PCA (Section 2.5.1).

Algorithm 5: Adversarial Non-linear ICA algorithm.

Result: Code space of statistically independent components
Apply a preprocessing procedure to the mixtures, if possible whitening: $\mathbf{z} = \mathrm{B}\mathbf{x}$.
Initialize all the parameters of the encoder, decoder and discriminator networks.
for *all training epochs* **do**
  Draw K minibatches z^k of L samples from the whitened dataset.
  for *each minibatch* z^k **do**
    Compute a minibatch of h^k of L code samples and normalize.
    Draw a minibatch $\mathrm{h}^k_{\mathrm{rp}}$ of L samples from the resampling procedure.
    Assemble a minibatch u^k, such that: $\mathrm{u}^k \leftarrow \{\mathrm{h}^k_{\mathrm{rp}}(\text{label 1}),\ \mathrm{h}^k(\text{label 0})\}$.
    Update D to decrease the cross-entropy cost of $\mathrm{d}(\mathrm{u}^k)$.
    Assemble a different minibatch u^k, such that: $\mathrm{u}^k \leftarrow \{\mathrm{h}^k(\text{label 1})\}$.
    Update F_{en} and F_{de} to decrease the cost function $\mathrm{J}_{\mathrm{AE}}(\mathrm{F}_{\mathrm{en}},\ \mathrm{F}_{\mathrm{de}})$.
  end
end

5.2.1 CONVERGENCE ANALYSIS

Directly based on the theorems and lemmas formulated in (GOODFELLOW et al., 2014) to prove the global convergence of GANs, in a similar manner it follows that the cost functions minimized by ANICA, i.e., (4.3) and (5.1), converge to a common stable value of $\log 2$ if, and only if, the encoder learns to retrieve a code $\mathbf{h} \sim \mathrm{p}_{\mathbf{h}}$ whose distribution is equal to the distribution after the resampling procedure, i.e., a code with statistically independent components. This conclusion is demonstrated in Theorem 5.2.1.

Theorem 5.2.1. Convergence of ANICA is achieved if, and only if, $\mathrm{p}_{\mathbf{h}}(\cdot) = \mathrm{p}_{\mathbf{rp}}(\cdot)$, which occurs only when the cross-entropy cost of the discriminator and the adversarial regularization term of the AE converge to $\log 2$. In this case, the discriminator can no longer distinguish among classes, so that $\mathrm{d}(\mathbf{u}) = 0.5$ for all input sample.

Proof. As proved in (GOODFELLOW et al., 2014), for a fixed AE and assuming there is access to the PDFs, the optimal mapping the discriminator may learn corresponds to:

$$\mathrm{d}_{\mathrm{opt}}(\mathbf{u}) = \frac{\mathrm{p}_{\mathbf{rp}}(\mathbf{u})}{\mathrm{p}_{\mathbf{rp}}(\mathbf{u}) + \mathrm{p}_{\mathbf{h}}(\mathbf{u})} \tag{5.2}$$

Then, by maintaining such optimal discriminator unchanged, the adversarial term in (5.1) may be expressed as:

$$\begin{aligned} \mathrm{J}^{\mathrm{adv}}_{\mathrm{AE}}(\mathrm{F}_{\mathrm{en}}) &= \mathbb{E}[-\log(\mathrm{d}_{\mathrm{opt}}(\mathbf{h}))] \\ &= \int \mathrm{p}_{\mathbf{h}}(\mathbf{h}) \log\left(\frac{\mathrm{p}_{\mathbf{rp}}(\mathbf{h}) + \mathrm{p}_{\mathbf{h}}(\mathbf{h})}{\mathrm{p}_{\mathbf{rp}}(\mathbf{h})}\right) \mathrm{d}\mathbf{h} \end{aligned} \tag{5.3}$$

It is possible to more conveniently rewrite (5.3) in terms of Kullback-Leibler divergences $\mathrm{D}_{\mathrm{KL}}(\cdot)$, by some algebraic manipulations, as follows:

$$\begin{aligned} \mathrm{J}^{\mathrm{adv}}_{\mathrm{AE}}(\mathrm{F}_{\mathrm{en}}) &= \int \mathrm{p}_{\mathbf{h}}(\mathbf{h}) \log\left(2 \cdot \frac{1}{2} \cdot \frac{\mathrm{p}_{\mathbf{rp}}(\mathbf{h}) + \mathrm{p}_{\mathbf{h}}(\mathbf{h})}{\mathrm{p}_{\mathbf{rp}}(\mathbf{h})}\right) \mathrm{d}\mathbf{h} \\ &= \int \mathrm{p}_{\mathbf{h}}(\mathbf{h}) \log(2) \mathrm{d}\mathbf{h} + \int \mathrm{p}_{\mathbf{h}}(\mathbf{h}) \log\left(\frac{\mathrm{p}_{\mathbf{rp}}(\mathbf{h}) + \mathrm{p}_{\mathbf{h}}(\mathbf{h})}{2\mathrm{p}_{\mathbf{rp}}(\mathbf{h})}\right) \mathrm{d}\mathbf{h} \\ &= \log 2 + \int \left[2 \cdot \frac{\mathrm{p}_{\mathbf{h}}(\mathbf{h}) + \mathrm{p}_{\mathbf{rp}}(\mathbf{h})}{2} - \mathrm{p}_{\mathbf{rp}}(\mathbf{h})\right] \log\left(\frac{\mathrm{p}_{\mathbf{rp}}(\mathbf{h}) + \mathrm{p}_{\mathbf{h}}(\mathbf{h})}{2\mathrm{p}_{\mathbf{rp}}(\mathbf{h})}\right) \mathrm{d}\mathbf{h} \\ &= \log 2 + 2 \int \frac{\mathrm{p}_{\mathbf{h}}(\mathbf{h}) + \mathrm{p}_{\mathbf{rp}}(\mathbf{h})}{2} \log\left(\frac{\mathrm{p}_{\mathbf{h}}(\mathbf{h}) + \mathrm{p}_{\mathbf{rp}}(\mathbf{h})}{2} \cdot \frac{1}{\mathrm{p}_{\mathbf{rp}}(\mathbf{h})}\right) \mathrm{d}\mathbf{h} + \cdots \\ &\quad \cdots + \int \mathrm{p}_{\mathbf{rp}}(\mathbf{h}) \log\left(\mathrm{p}_{\mathbf{rp}}(\mathbf{h}) \cdot \frac{2}{\mathrm{p}_{\mathbf{h}}(\mathbf{h}) + \mathrm{p}_{\mathbf{rp}}(\mathbf{h})}\right) \mathrm{d}\mathbf{h} \end{aligned} \tag{5.4}$$

From (5.4), the two last terms may be identified as Kullback-Leibler divergencies among a pair of PDFs in opposite orders. Finally, the adversarial term is conveniently expressed as:

$$\mathrm{J}^{\mathrm{adv}}_{\mathrm{AE}}(\mathrm{F}_{\mathrm{en}}) = \log 2 + 2\mathrm{D}_{\mathrm{KL}}\left(\frac{\mathrm{p}_{\mathbf{h}} + \mathrm{p}_{\mathbf{rp}}}{2} \parallel \mathrm{p}_{\mathbf{rp}}\right) + \mathrm{D}_{\mathrm{KL}}\left(\mathrm{p}_{\mathbf{rp}} \parallel \frac{\mathrm{p}_{\mathbf{h}} + \mathrm{p}_{\mathbf{rp}}}{2}\right) \tag{5.5}$$

Since $\mathrm{D_{KL}}(\cdot)$ is always non-negative, the minimum value of $\mathrm{J}^{\mathrm{adv}}_{\mathrm{AE}}(\cdot)$ may attain is $\log 2$, which occurs, if and only if, both divergences are null or, equivalently, when $\mathrm{p_{\mathbf{h}}}(\cdot) = \mathrm{p_{\mathbf{rp}}}(\cdot)$. As a consequence, $\mathrm{d_{opt}}(\mathbf{u}) = 0.5$ and, finally, (5.2) becomes equal to $\log 2$, as well, thus concluding the proof. □

Once that ANICA has converged and $\mathrm{p_{\mathbf{h}}}(\cdot) = \mathrm{p_{\mathbf{rp}}}(\cdot)$, it follows that the components $h_i \sim \mathrm{p}_{h_i}$ of the random code vector $\mathbf{h} \sim \mathrm{p_{\mathbf{h}}}$ are statistically independent due to the statistical independence of the resampling procedure, thus $\mathrm{p_{\mathbf{h}}}(\cdot) = \prod_{\forall i} \mathrm{p}_{h_i}(\cdot)$.

5.3 ANICA AND ITS LIMITS FOR ICA

As already explained in Section 2.17 and Section 2.5, the linear and instantaneous ICA case only requires the statistical independence of the retrieved estimates $\mathbf{y}$ in order to guarantee that they capture an scaled and permuted representation of the independent sources $\mathbf{s}$, which may only contain one Gaussian source at maximum. Hence, it directly follows that such ICA case can be solved by ANICA, once it has attained convergence and, thus, the encoder has learnt to project a code of statistically independent estimates from the (pre processed or not) mixtures $\mathbf{x}$ it receives.

ANICA would simply need to be modelled with a LAE (Section 4.2.1), such that the trained linear encoder would fulfill the role of the separation matrix W, while the linear decoder would resemble the mixing matrix A. Considering a centered input mixture vector $\mathbf{x}$, thus all vectors would also be centered, no bias term would be necessary, and the trained linear model could be expressed as:

$$\left.\begin{array}{lll} \mathbf{y} & = \Theta_{\mathrm{en}}\mathbf{x} & = \mathrm{WA}\mathbf{s} \\ \hat{\mathbf{x}} & = \Theta_{\mathrm{de}}\mathbf{y} & = \mathrm{A}\mathbf{s} \end{array}\right\} \hat{\mathbf{x}} = \Theta_{\mathrm{de}}\Theta_{\mathrm{en}}\mathbf{x}, \tag{5.6}$$

where $\Theta_{\mathrm{en}} = \mathrm{W} \in \Re^{M \times N}$ and $\Theta_{\mathrm{de}} = \mathrm{A} \in \Re^{N \times M}$.

5.3.1 UNDERDETERMINEDNESS AND ANICA

Unfortunately, the aforementioned linear model, by itself, does not have the capacity to deal with the underdetermined ICA case ($N < M$), even when the number of sources M is known, due to the inherit column-rank-deficiency of the matrix product WA, for any linear encoder with a weight matrix Θ_{en}, which limits the number of sources to possibly be recovered to a maximum of $N - 1$ independent components.

This only happens in the rare and exceptional case when the $M - N$ linearly dependent columns of the mixing matrix A are all multiples of a single one of those N linearly independent, then, the respective $M - N + 1$ sources would be mixed into a different "source" statistically independent from the rest and, thus, they would be

inseparable. Nevertheless, such case is highly unlikely, then it could not be possible for ANICA to retrieve any independent estimates in a more typical underdetermined setting.

5.3.2 POST-NONLINEAR ICA

In addition to linear instantaneous ICA, ANICA is also well fit for Post-Nonlinear (PNL) ICA (TALEB; JUTTEN, 1997; TALEB; JUTTEN, 1999). Very briefly, the PNL model considers that the linear instantaneous mixtures are each followed by a non-linear invertible mapping $f_i(\cdot)$. Then, a separation system attempts to invert each of them by a corresponding mapping $g_i(\cdot)$, such that a final separation matrix W should yield uncorrupted independent estimates. The PNL model may be expressed as:

$$\begin{aligned} \mathbf{x} &= \mathbf{f}(\mathrm{A}\mathbf{s}) \\ \mathbf{y} &= \mathrm{W}\mathbf{g}(\mathbf{x}), \end{aligned} \tag{5.7}$$

where $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ are arrays containing all the $f_i(\cdot)$ and $g_i(\cdot)$ functions, respectively.

As explained in (JUTTEN; BABAIE-ZADEH; KARHUNEN, 2010), unlike other non-linear ICA models, PNL mainly requires the statistical independence of the retrieved estimates to guarantee they represent the sources, with no more than one Gaussian PDF, which is the same idea on which ANICA is based. There is a caveat nonetheless: the mixing system must be invertible, which raises some particularities that must be carefully taken into account.

A very interesting approach to solve the PNL problem is proposed in (DUARTE et al., 2006), where the authors employ concepts drawn from the field of evolutionary computation, specifically artificial immune systems (CASTRO; ZUBEN, 2002), to locate the global optimal of a cost function based on the minimization of the mutual information.

5.4 SUMMARY

In this chapter, we presented the main ideas of ANICA, exploring the previously discussed concepts of AAEs and GANs, and addressing convergence and the effectiveness of the obtained solutions. Additionally, we have also discussed its applicability to underdetermined and nonlinear scenarios.

In order to test our theoretical analysis, in the following chapter we present a series of experiments designed to prove the behaviour of ANICA under different scenarios and to see how it performs when compared with FastICA and JADE (Section 2.5.2).

6 EXPERIMENTS

In this chapter, we will analyze the behavior, the limitations and the potential advantages of ANICA applied to the linear and instantaneous BSS problem. We designed a set of scenarios to cover different aspects of the task, as well as to pose different challenges for the studied method. Additionally, we established a comparison between ANICA and two well-known algorithms: FastICA (Section 3.2.3) and JADE (Section 3.3.1), considering the Normalized Amari error (NAE) (AMARI; CICHOCKI; YANG, 1996) as the main performance measure.

We also propose a set of metrics, based on the criteria for ICA already presented in Chapter 3, for the blind selection of the training epoch that produces the best code, which are presented in the sequence.

6.1 METRICS AND METHODOLOGY

In (BRAKEL; BENGIO, 2017), the performance of ANICA was evaluated by the computation of all the Pearson correlation coefficients, for every training epoch, between each code h_i and every source s_i, such that the absolute value of the maximum coefficient should tend towards one to indicate that the given code was representative of a unique source. Although this choice was useful for demonstrating the effectiveness of the proposed method, from a practical standpoint it suffers from a major drawback, as it requires the knowledge of the individual sources.

Here, having as inspiration the criteria explained in Chapter 3, we propose a set of unsupervised metrics, i.e., which do not use any information regarding the sources, to monitor the evolution of the codes h_i during the training process in order to identify the most adequate epoch to retrieve the source estimates. In particular, we resorted to the notions of mutual information (see Equation (3.3)) and non-gaussianity, either measured by the negentropy and the normalized kurtosis (see Equations (3.10) and (3.8)), and defined the following blind metrics:

$$\begin{aligned} \mathrm{SMI}(\mathbf{h}) &= \mathrm{MI}(\mathbf{h}) + \mathrm{H}(\mathrm{p}_{\mathbf{z}}) \\ &= \left(\sum_{\forall i} \mathrm{H}(\mathrm{p}_{h_i})\right) - \log(|\det(\Theta_{\mathrm{en}})|) \end{aligned} \tag{6.1}$$

$$\mathrm{MNK}(\mathbf{h}) = \sqrt[2]{\sum_{\forall i} \mathrm{kurt}(h_i)^2} \tag{6.2}$$

$$\mathrm{MN}(\mathbf{h}) = \sqrt[2]{\sum_{\forall i} \mathrm{J}(h_i)^2} \tag{6.3}$$

Equation (6.1) considers a shifted mutual information (SMI) due to the joint entropy of the whitened mixtures $\mathrm{H}(\mathrm{p}_{\mathbf{z}})$ being constant through out the training iterations of ANICA, thus, its calculation is not necessary, while expressions (6.2) and (6.3) are only the modules (Euclidean norms) of the normalized kurtosis (MNK) and the negentropy (MN), respectively.

Additionally, and only for corroboration purposes, we also employ the Normalized Amari error (NAE) (AMARI; CICHOCKI; YANG, 1996), which is basically a metric that measures the capability of a separation matrix W to "invert" the mixing matrix A, such that its value grows from 0 to 1 as the separation becomes worse:

$$\mathrm{NAE}(\mathrm{W},\ \mathrm{A}) = \frac{1}{2M(M-1)} \left(\sum_{i=1}^{M} \left(\frac{\sum_{j=1}^{M} |wa_{i,j}|}{\max_j |wa_{i,j}|} - 1 \right) + \sum_{j=1}^{M} \left(\frac{\sum_{i=1}^{M} |wa_{i,j}|}{\max_i |wa_{i,j}|} - 1 \right) \right), \quad (6.4)$$

where each $wa_{i,j}$ represents the coefficient of matrix WA at a row i and column j, and M is the number of sources.

As already mentioned, the focus of this dissertation lies on the linear and instantaneous ICA case. Hence, a linear model of ANICA has been implemented to deal with such problem. Similarly to (BRAKEL; BENGIO, 2017), the discriminator was implemented with a single hidden layer of 64 ReLu activated neurons. The model was initialized with Xavier initialization (GLOROT; BENGIO, 2010) for the discriminator, and random orthogonal matrices for the LAE. Then, we trained ANICA using the RMSProp algorithm (TIELEMAN; HINTON, 2012), with default parameters, for only 4000 epochs and minibatches of 1024 samples, in contrast to the 500000 epochs and minibatches of 64 samples originally considered by (BRAKEL; BENGIO, 2017).

After training, we selected the training epochs that optimized the aforementioned blind metrics and, then, we calculated the NAE by using the encoder weight matrix for such epochs as the separation matrix W. The computed NAE, i.e. $\mathrm{NAE}(\Theta_{\mathrm{en}},\ \mathrm{A})$, is used to compare the quality of the separation achieved by ANICA to that of FastICA and JADE. In most experiments, we have repeated training 10 times, in an attempt to compensate for the effects of initialization and offer more robust results. Thus, the reported NAEs are an average across such repetitions.

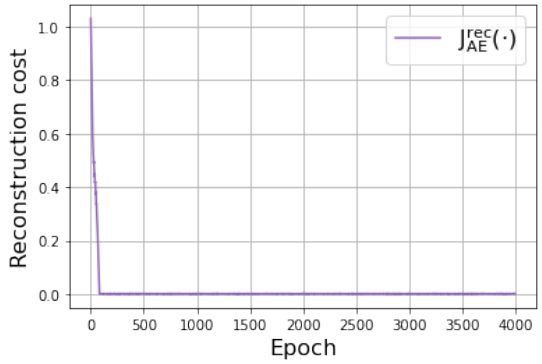
Unless stated otherwise, for each experiment a synthetic dataset of mixtures was created from zero-mean, unit-variance sources and mixing matrices, each drawn from uniform and Gaussian PDFs, respectively. As already indicated, the mixtures were whitened before training.

6.2 INITIAL EXPERIMENT: METRICS AND CONVERGENCE

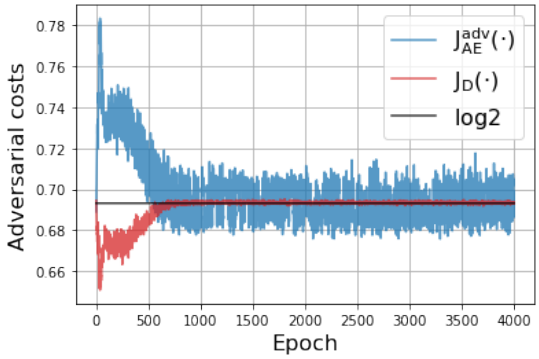
A warm-up experiment is designed for a dataset of 2^{14} samples and two sources in order to show how the blind metrics indeed capture the same information as the NAE,

as well as to illustrate the results yielded by ANICA and the training process in general. For this example, we have considered no repetitions.

Figure 6.1 shows the training curves along the 4000 epochs. In Figure 6.1a it is noticeable that the LAE learns very rapidly to reconstruct the mixtures at the output layer, while Figure 6.1b confirms the theoretical results obtained in Section 5.2.1 regarding convergence around the value of $\log 2$ for the adversarial costs, i.e., the cross-entropy of the discriminator $\mathrm{J_D}(\cdot)$ and the adversarial term $\mathrm{J_{AE}^{adv}}(\cdot)$.



(a) Reconstruction cost of the AE.



(b) Adversarial convergence to $\log 2$.

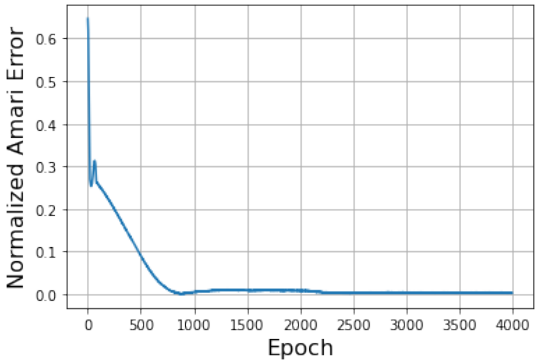
Figure 6.1 – Training curves and convergence of ANICA.

For each training epoch, we also computed all the metrics – NAE, SMI, MNK and MN – considering the encoder and/or the code components at that moment. The corresponding curves are exhibited in Figure 6.2.

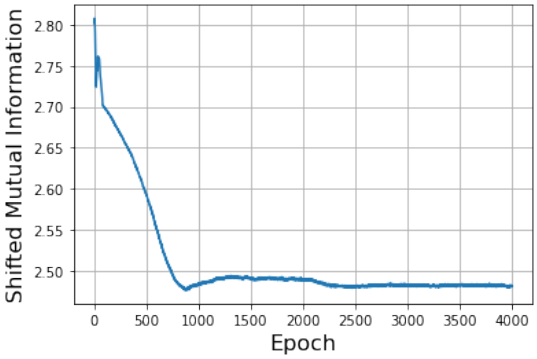
We can clearly notice in Figure 6.2 that they all behave similarly, almost capturing every value fluctuation and, most importantly, they are optimal around the same epoch. Figure 6.2a depicts the NAE, which will be our reference for the following experiments, while Figure 6.2b is the SMI, which basically behaves as an scaled version of the previous metric. Figures 6.2c and 6.2d correspond to the MNK and MN, respectively, and, due to their maximization objective, capture a vertical reflection of the NAE.

Finally, the data on which ANICA has been trained and the results are shown in Figure 6.3. Figure 6.3a contains the unknown sources, and Figure 6.3b the input data samples. After training, we can see that the encoder has learnt to capture a meaningful code that actually resembles the sources, as portrayed in Figures 6.3c, while the complete AE has succeeded in its "original" task of reconstruction, which is shown in Figure 6.3d.

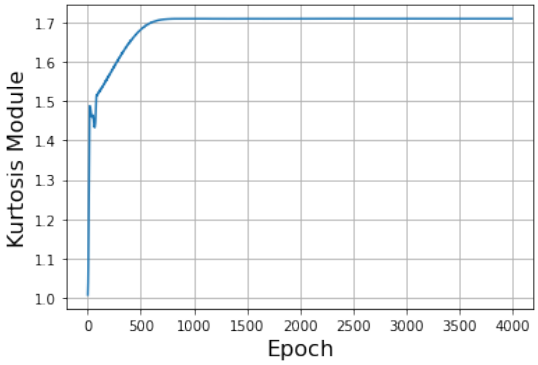
Additionally, Figure 6.4 shows the PDFs involved in the training of ANICA. Figures 6.4a and 6.4c reveal the independence of the uniform PDF of the sources and the correct matching on the code vector, respectively, while Figures 6.4b and 6.4d show that the decoder reconstructs the input similarly to a whitened mixing matrix $\mathbf{A_z}$.



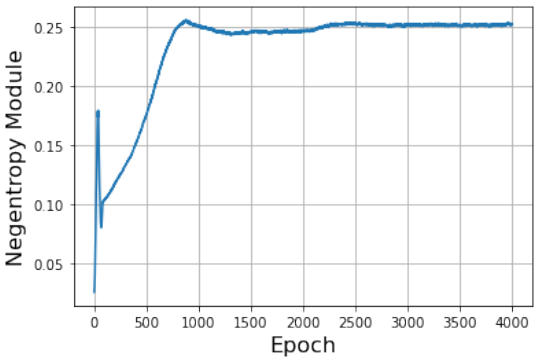
(a) Normalized Amari Error.



(b) Shifted Mutual Information.

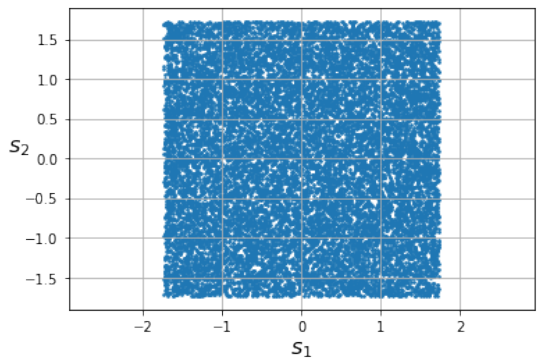


(c) Module of the Normalized Kurtosis.

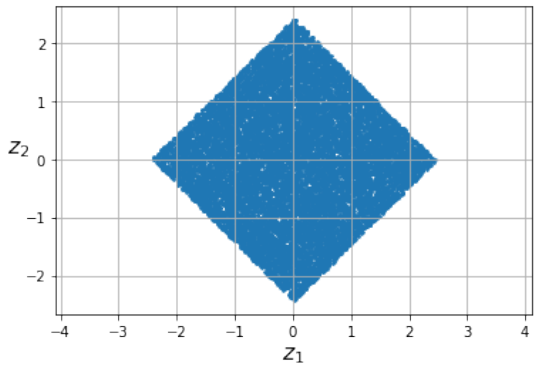


(d) Module of the Negentropy.

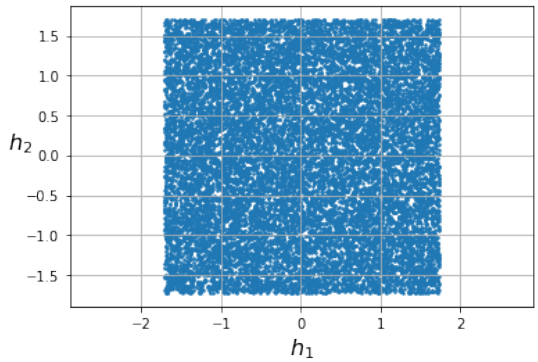
Figure 6.2 – Proposed blind metrics to evaluate independence.



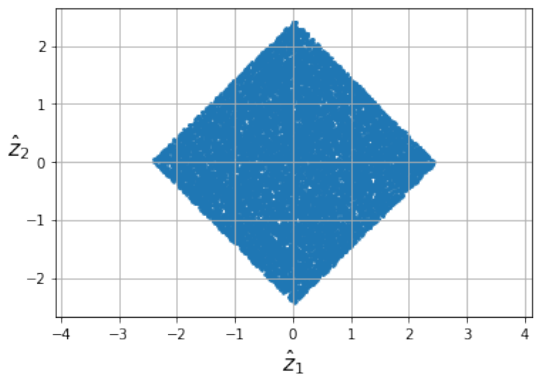
(a) Uniform sources $\mathbf{s}$.



(b) Whitened mixtures $\mathbf{z}$.

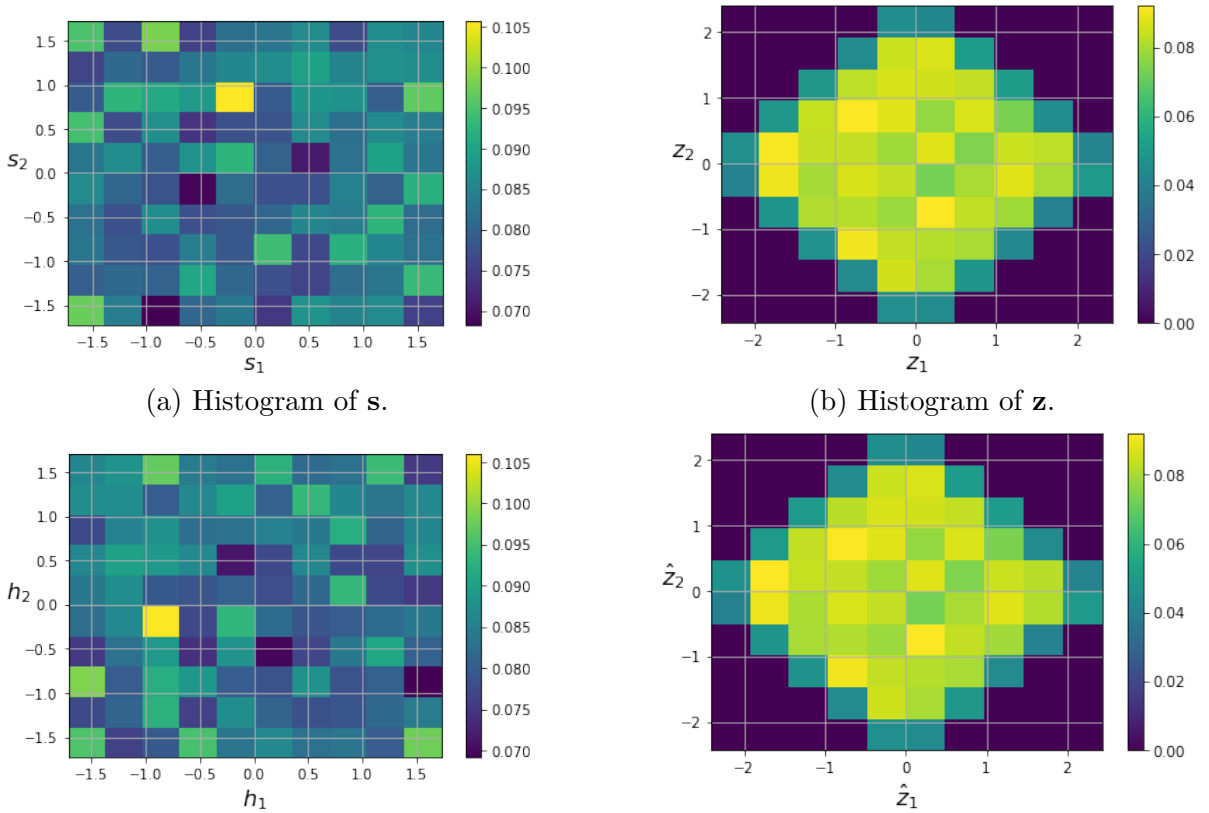


(c) Independent codes $\mathbf{h}$.



(d) Reconstructions $\hat{\mathbf{z}}$.

Figure 6.3 – Retrieval of independent estimates from whitened mixtures by ANICA.



(a) Histogram of $\mathbf{s}$.

(b) Histogram of $\mathbf{z}$.

(c) Histogram of $\mathbf{h}$.

(d) Histogram of $\hat{\mathbf{z}}$.

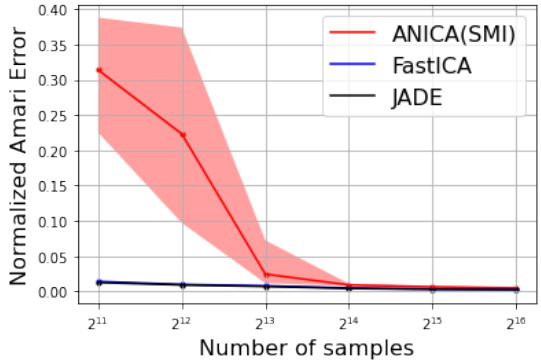
Figure 6.4 – Matching of the PDFs of the data performed by ANICA.

6.3 EXPERIMENT 1: NUMBER OF SAMPLES

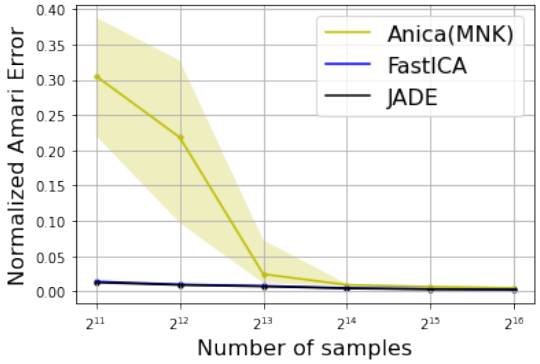
The first relevant scenario to analyze refers to the impact of the size of the training dataset. For this experiment, the number of whitened mixtures is fixed to $M = 9$ components, while the size of the dataset increases by a factor of 2 during the experiment, such that the number of samples ranges from 2^{11} to 2^{16} samples. The mixture matrix A is fixed and the samples of the sources are not replaced, but only extended.

Figure 6.5 shows the progression of the NAE, obtained by each of the studied algorithms with respect to the number of samples, such that Figures 6.5a, 6.5b and 6.5c consider the epochs selected with respect to the SMI, MNK and MN, respectively. As we can observe, increasing the size of the dataset reduces both the average NAE (represented by dots connected in a continuous line), improving the performance, and the dashed region between the minimum and maximum NAE across all repetitions, making the model more robust. Nonetheless, even in Figure 6.5d, where the aforementioned dashed region thins, the NAE values associated with ANICA are always above those attained by FastICA and JADE.

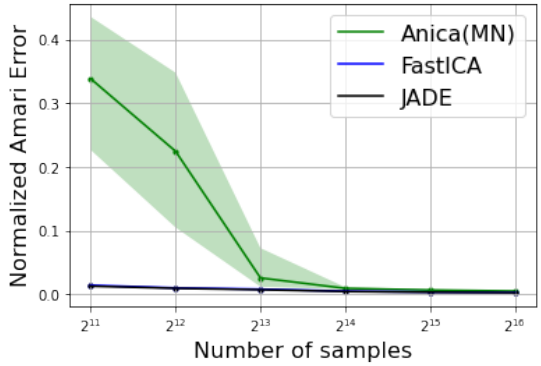
The effect on convergence is shown in Figure 6.6. Initially, for the smallest training set of 2^{11} samples, ANICA does not effectively converge, as can be seen in Figure



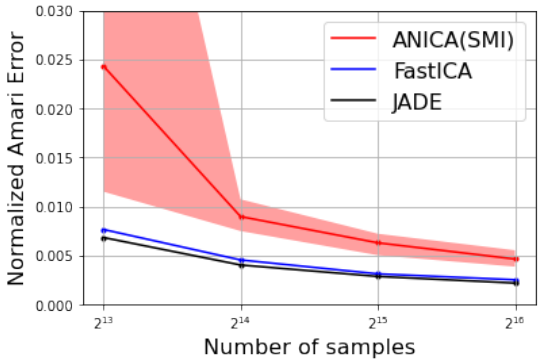
(a) NAE on the epochs with optimal SMI.



(b) NAE on the epochs with optimal MNK.



(c) NAE on the epochs with optimal MN.



(d) Section of interest.

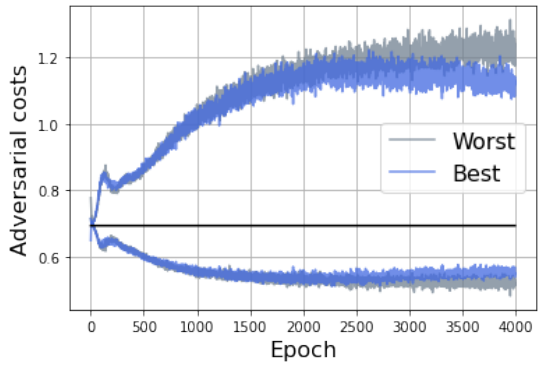
Figure 6.5 – Experiment 1: Impact of the number of samples on the attained independence.

6.6a considering the best and the worst training repetition. In both cases, more training epochs would be necessary for a complete convergence. Nonetheless, the adversarial costs do start to converge for 2^{13} samples, as indicated in Figure 6.6b, which helps to explain why the dashed regions in Figure 6.5 start to thin when the dataset is larger. Figures 6.6c and 6.6d confirm the critical impact of the number of samples on convergence and, thus, on the independence attained by ANICA. Based on these results, all the subsequent experiments shall consider $2^{14} = 16384$ samples.

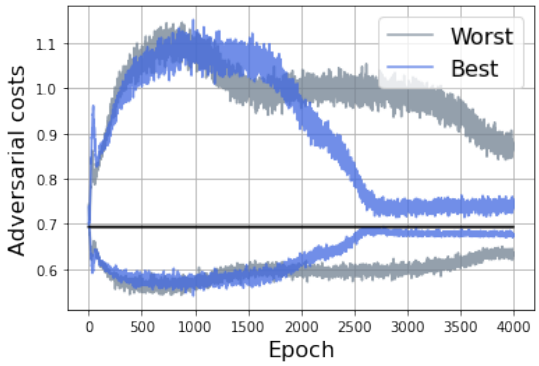
6.4 EXPERIMENT 2: NUMBER OF COMPONENTS

Secondly, we analyzed how ANICA is influenced by the number of components of the input whitened mixture vector $\mathbf{z}$. In this experiment, we initially created a 9-component source vector with a corresponding mixture matrix $A \in \Re^{9\times9}$, with the intention of producing mixtures by the subsequent multiplication between a submatrix of A and the source vector, such that both increase their dimensionality from 2 to finally 9. In this fashion, we try to uniquely modify the number of components and avoid to train on completely new samples.

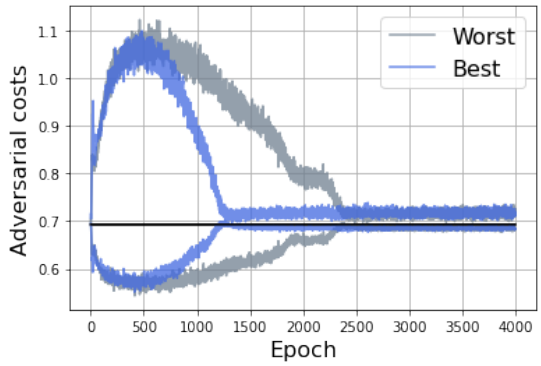
Figure 6.7 shows the average NAE as a function of the number of components. In general, according to all metrics, the performance deteriorates as the number of sources



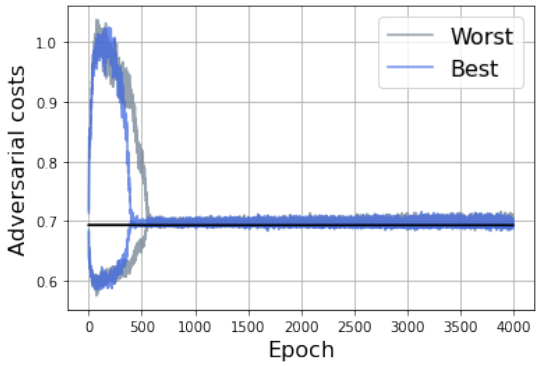
(a) Convergence for 2^{11} samples.



(b) Convergence for 2^{13} samples.



(c) Convergence for 2^{14} samples.



(d) Convergence for 2^{15} samples.

Figure 6.6 – Experiment 1: Impact of the number of samples on convergence.

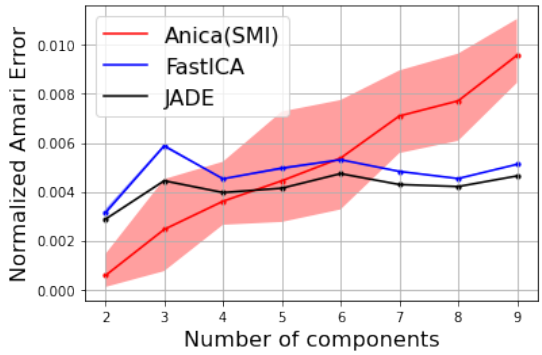
increases, as can be seen in Figures 6.7a, 6.7b and 6.7c. Nonetheless, we can see that ANICA performs surprisingly better than FastICA and JADE in the cases up to $M = 6$ sources, as Figure 6.7d demonstrates.

It is worth mentioning that it was necessary to repeat training once for the 3-component dataset in order to have 10 instances of convergence between the adversarial costs and to compute the average NAE.

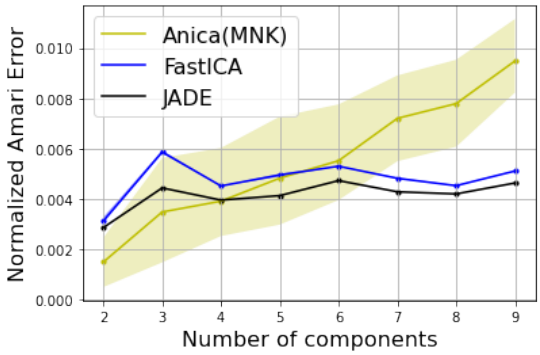
6.5 EXPERIMENT 3: ANICA WITHOUT THE DECODER

Following the same methodology employed in Section 6.4 for an increasing number of components, in this experiment we have decided to test how well the adversarial elements in ANICA behave without the decoder, i.e., how successful would the encoder be in fooling the discriminator when there is no reconstruction task for which to adapt its parameters.

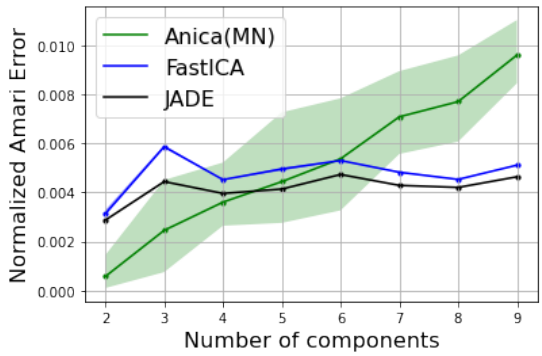
Figure 6.8 shows the average NAE with respect to the number of components for the ANICA model without the decoder. In comparison to the previous analogous experiment, the results shown in Figures 6.8a, 6.8b and 6.8c indicate that ANICA does not reach the same level of independence at the code layer, though the NAE values are sufficiently small to consider that it was able to separate the sources. Similarly, this reduced



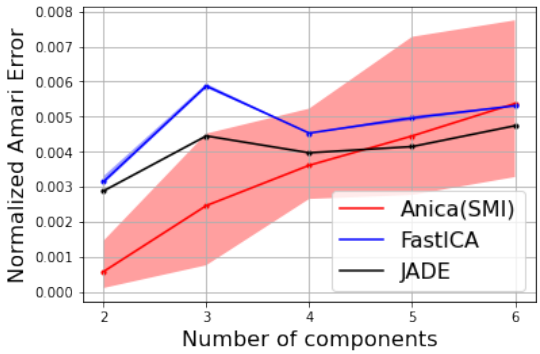
(a) NAE on the epochs with optimal SMI.



(b) NAE on the epochs with optimal MNK.



(c) NAE on the epochs with optimal MN.



(d) Section of interest.

Figure 6.7 – Experiment 2: Impact of the number of components on the attained independence.

ANICA model performs slightly better than FastICA and JADE only when $M = 2$ or $M = 3$ sources are considered, as Figure 6.8d demonstrates.

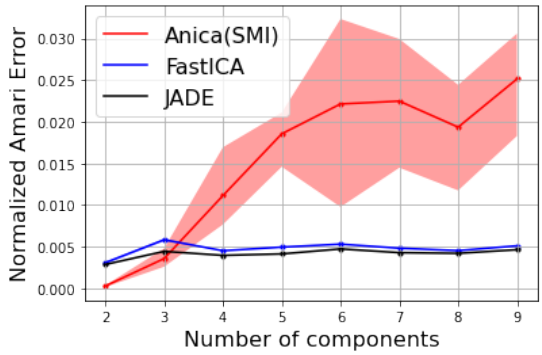
These results suggest that the presence of the decoder within ANICA, whose structure mimics that of the actual mixing system, encourages the encoder towards a refined separation matrix, as the entire model also is trained to minimize the reconstruction error of the mixtures.

6.6 EXPERIMENT 4: SIGNAL-TO-NOISE RATIO

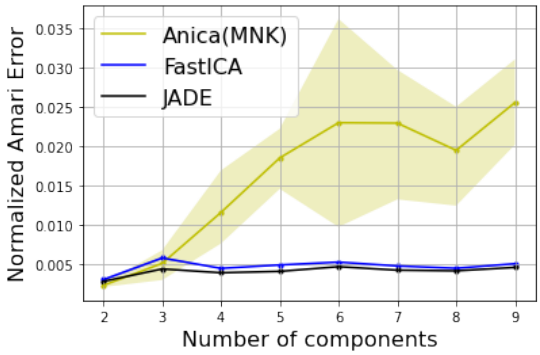
In this experiment, we assessed how ANICA behaves when the original mixture vector $\mathbf{x}$ is corrupted by a random noise signal $\mathbf{n}$. For this purpose, we generated White Gaussian Noise (WGN) samples with variance $\sigma^2_{\mathbf{n}}$, an added them to the 9-component $\mathbf{x}$ to produce a corrupted mixture vector $\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{n}$. Then, whitening and ANICA follow as usual for different noise variances, measured by the Signal-To-Noise Ratio (SNR) in decibels, which is expressed as:

$$\mathrm{SNR}_{\mathrm{dB}} = 10 \log \left(\frac{\sigma^2_{\mathbf{s}}}{\sigma^2_{\mathbf{n}}} \right), \tag{6.5}$$

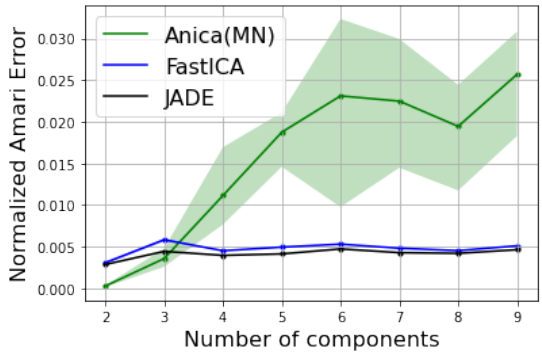
where σ^2_s indicates the common variance of the sources, which is assumed to be unitary.



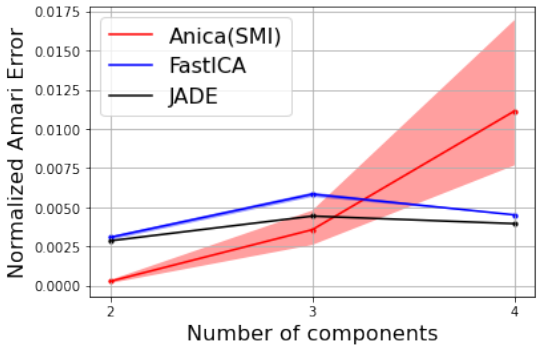
(a) NAE on the epochs with optimal SMI.



(b) NAE on the epochs with optimal MNK.



(c) NAE on the epochs with optimal MN.

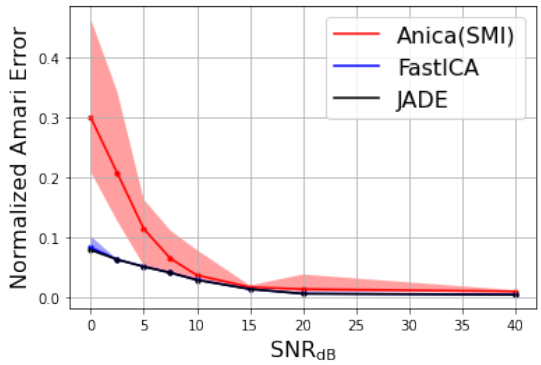


(d) Section of interest.

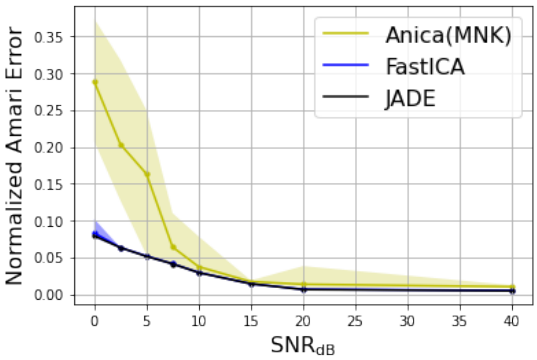
Figure 6.8 – Experiment 3: Impact of the lack of decoder on the attained independence.

The obtained NAE values are presented in Figure 6.9 with respect to the SNR_{dB}, where we can observe that ANICA is more susceptible to initialization when the noise variance is large, as Figures 6.9a, 6.9b and 6.9c indicate for the blind metrics. For example, for $\text{SNR}_{dB} = 0$, i.e., $\sigma_{\mathbf{n}}^2 = \sigma_{\mathbf{s}}^2 = 1$, there is wide dashed region which only becomes sufficiently thin after $\text{SNR}_{dB} = 7.5$, i.e., $\sigma_{\mathbf{n}}^2 = 0.1778$. Conversely, initialization is less important once such noise variance becomes less dominant, as can be seen in Figure 6.9d, which means that the independent executions of ANICA (starting from different initial parameters) led to NAE values closer to those attained by JADE and FastICA.

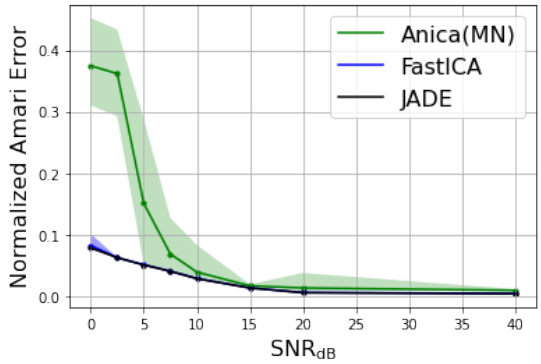
Figure 6.10 complements the previous explanation regarding the wide variations observed in the dashed regions. In Figure 6.10a, we can see that even the best training instance delivers parallel adversarial costs that may not be able to achieve the ideal value of $\log 2$, which confirms that retrieving independent estimates of the sources is more difficult for noisy datasets. Once the SNR_{dB} increases to 15 ($\sigma_{\mathbf{n}}^2 = 0.0316$), as in Figure 6.10b, all repetitions converge, albeit with some delay, so better results could be obtained. However, convergence is not guaranteed, as Figure 6.10c with $\sigma_{\mathbf{n}}^2 = 0.01$ indicates, where one training instance that does not fully converge, thus widening its respective dashed region. Finally, for $\text{SNR}_{\text{dB}} = 40$ and $\sigma_{\mathbf{n}}^2 = 0.0001$, as shown in Figure 6.10d, the effect of noise on convergence is still important, but no longer critical.



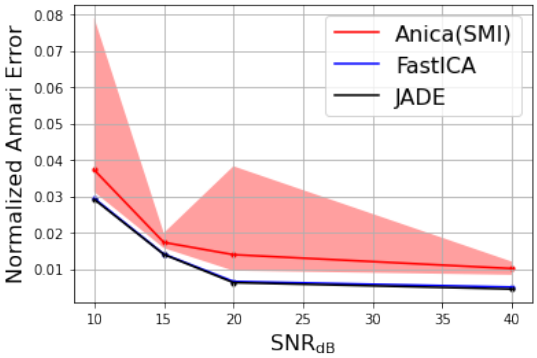
(a) NAE on the epochs with optimal SMI.



(b) NAE on the epochs with optimal MNK.

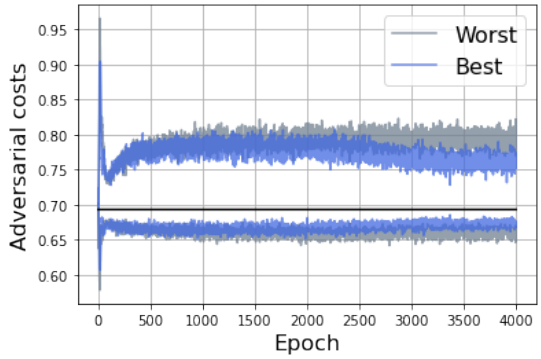


(c) NAE on the epochs with optimal MN.

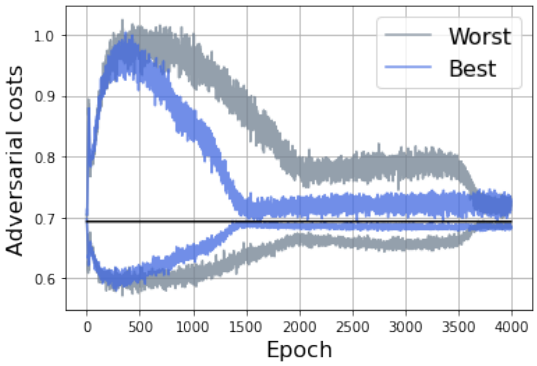


(d) Section of interest.

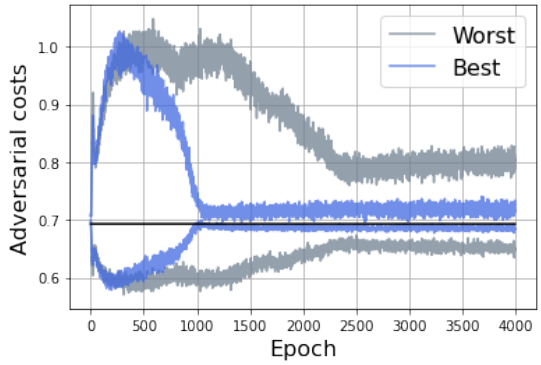
Figure 6.9 – Experiment 4: Impact of the SNR_{dB} on the attained independence.



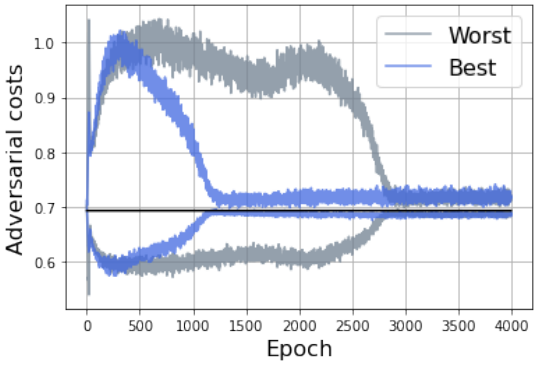
(a) Convergence for $SNR_{dB} = 0$.



(b) Convergence for $SNR_{dB} = 15$.



(c) Convergence for $SNR_{dB} = 20$.



(d) Convergence for $SNR_{dB} = 40$.

Figure 6.10 – Experiment 4: Impact of the SNR_{dB} on convergence

6.7 EXPERIMENT 5: OVERDETERMINED CASE

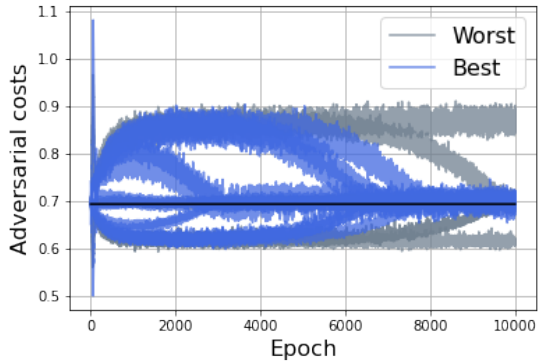
In order to evaluate ANICA in the overdetermined case, we designed this experiment such that the number of sources was kept fixed at $M = 2$, while the number of available mixtures grew from $N = 3$ to $N = 9$. Unlike previous experiments, the dataset was reduced to $2^{13} = 8192$ samples and, thus, it was necessary to train the model for 10000 epochs.

For the overdetermined ICA case we have considered two approaches to retrieve estimates with ANICA: (i) the reduction of dimensionality to remain in the well-determined setting; and (ii) the use of an underdetermined LAE to automatically attain a code with only $M = 2$ components.

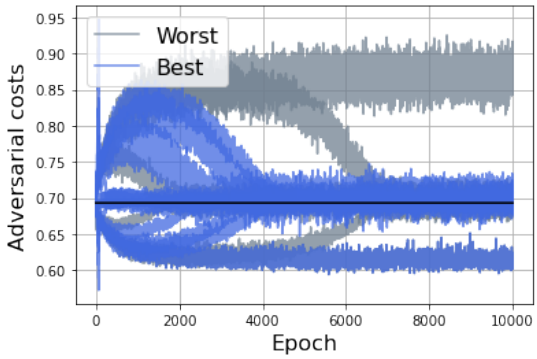
6.7.1 EXPERIMENT 5.1: DIMENSIONALITY REDUCTION BY PCA

In the first approach, similarly to (JOHO; MATHIS; LAMBERT, 2000) (Section 2.5.1), we identified the number of sources by calculating the rank of the covariance matrix of the mixtures, as the rank and the number of sources must coincide, and then reduced the dimensionality with PCA to deal with a well-determined problem.

Unfortunately, the model did not always converge, thus, we were forced to select only the 7 best training curves for each value of N with respect to the lowest SMI. Figure 6.11 depicts all the convergence curves for the most problematic instances: when N was equal to 3, in Figure 6.11a, and when $N = 6$, in Figure 6.11b, such that the selected curves are in blue.



(a) Training curves for $N = 3$.

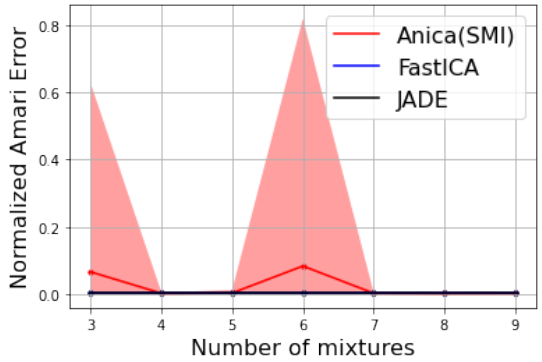


(b) Training curves for $N = 6$.

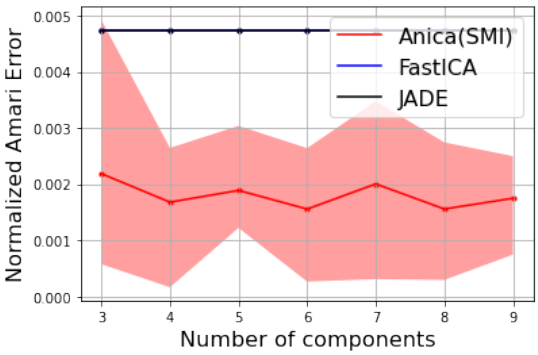
Figure 6.11 – Experiment 5.1: All convergence curves.

The attained results, measured by the average NAE, as usual, are displayed in Figure 6.12. Initially, in Figure 6.12a, we observe that, depending on the number of mixtures and on the repetition of ANICA, a poor NAE value can be obtained. For instance, the worst execution of ANICA for $N = 6$ led to a NAE slightly above 0.8, which means that the sources were not separated. However, if we remove the trials in which ANICA did

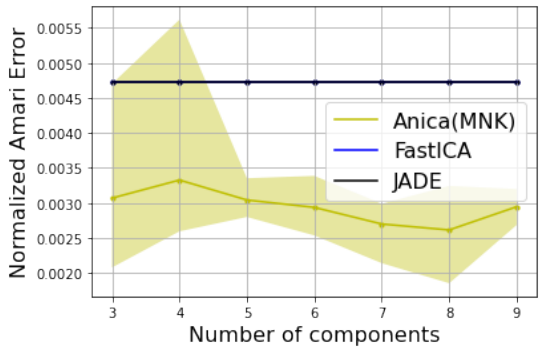
not converge, the results are outstanding and much better than those attained by FastICA and JADE, as Figures 6.12b, 6.12c and 6.12d prove.



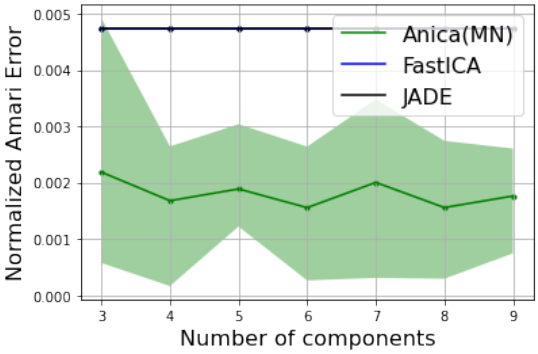
(a) All training curves.



(b) NAE on the epochs with optimal SMI.



(c) NAE on the epochs with optimal MNK.



(d) NAE on the epochs with optimal MN.

Figure 6.12 – Experiment 5.1: Impact of the number of mixtures on the attained independence considering the overdetermined scenario.

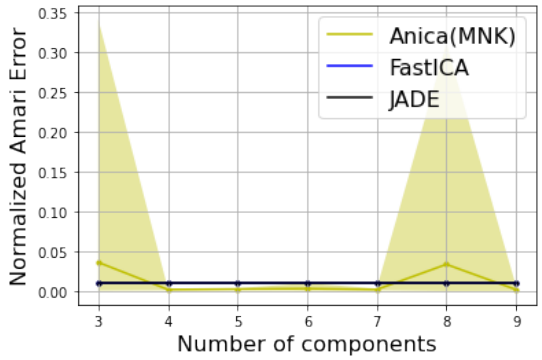
6.7.2 EXPERIMENT 5.2: UNDERCOMPLETE AUTOENCODER

In the second approach, we also considered the rank to identify the number of sources M and, then, we simply define an undercomplete LAE to produce a code of M components, having as input the set of N mixtures. Importantly, due to the rectangular shape of the encoder matrix Θ_{en}, it was not possible to calculate the SMI.

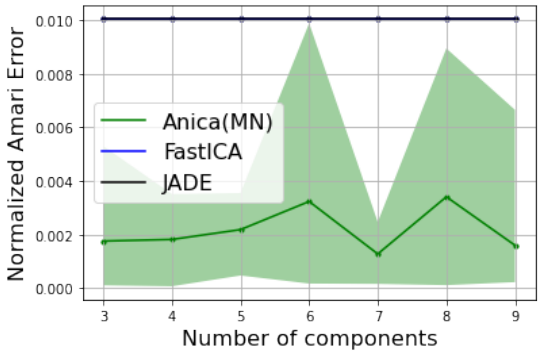
Interestingly, unlike the previous approach, ANICA adequately converged in all trials when its AE is undercomplete. Figure 6.13 shows that, although the MNK in Figure 6.13a may not be the right metric for this approach, the results are still encouraging, as reported in Figure 6.13b, where for all training curves the average NAE is still lower than those associated with both FastICA and JADE.

6.8 EXPERIMENT 6: GAUSSIANITY

Now that we have analyzed the behaviour of ANICA in different scenarios, there only remains the task of testing the quality of the independent code to capture Gaussian



(a) NAE on the epochs with optimal MNK.



(b) NAE on the epochs with optimal MN.

Figure 6.13 – Experiment 5.2: Impact of the undercomplete autoencoder on the attained independence.

and super-Gaussian sources. In this experiment, we simply produce three mixtures from a Gaussian, Laplacian (super-Gaussian) and uniform (sub-Gaussian) distributions and, then, we compare the results attained by ANICA and perform the aforementioned comparison to FastICA and JADE by means of the NAE. Since we already studied the performance of ANICA with respect to the number of sources and samples, and in the presence of noise, in this experiment we do not increase the challenge and no training repetition was deemed necessary.

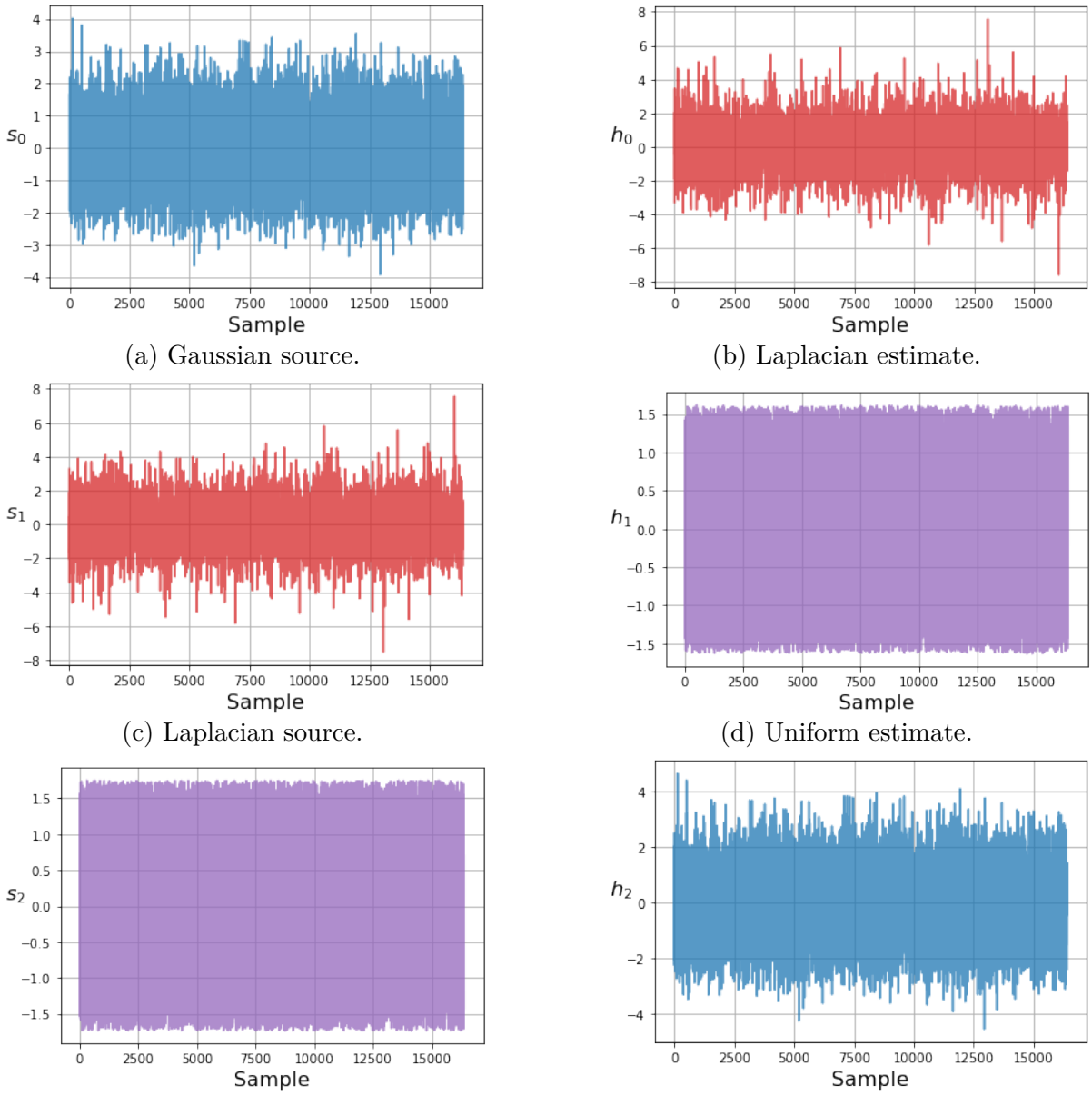
Figure 6.14 shows the final results attained by ANICA in the code layer after 4000 epochs, as usual. Figures 6.14a, 6.14c and 6.14e depict the unknown original sources, while Figures 6.14b, 6.14d and 6.14f hold the captured codes. As we can notice, there is a clear correspondence between sources and codes of the same colour, which is made evident in Figure 6.15 due to the almost perfect correlation between source and estimate, such that the absolute value of the Pearson coefficient in each case is approximately equal to 1.

The best possible NAE directly measured on the training epochs was equal to NAE = 0.004, which indicates that ANICA may outperform FastICA (NAE = 0.0089) and JADE (NAE = 0.0138) in this scenario.

6.9 EXPERIMENT 7: REAL AUDIO SIGNALS

Finally, we considered the application of ANICA in the context of real signals. In particular, we created four mixture signals given an equal number of audio sources selected from the audio database reported in (KABAL, 2002).

For this case, we have only considered 2000 epochs with no training repetitions. Figure 6.16 shows every audio source signal and all the codes retrieved by ANICA, such that there is a clear correspondence figures of the same colour. The success of ANICA is also confirmed by the NAE, which was equal to NAE = 0.0007, being considerably



(a) Gaussian source.
(b) Laplacian estimate.
(c) Laplacian source.
(d) Uniform estimate.
(e) Uniform source.
(f) Gaussian estimate.

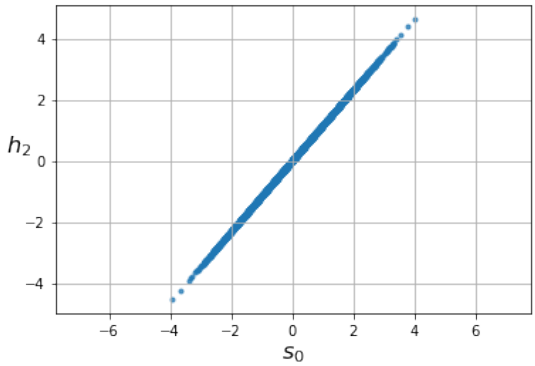
Figure 6.14 – Experiment 6: Impact of the Gaussianity of the sources on ANICA.

smaller than those attained by FastICA and JADE, which were equal to NAE = 0.0099 and NAE = 0.0164, respectively.

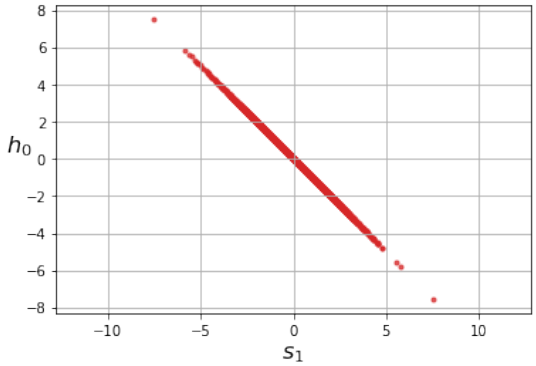
Similarly to the previous experiment, we have included Figure 6.17 to indicate the almost perfect correlation (with an absolute Pearson correlation approximately equal to 1) between sources and their corresponding estimates in the code layer.

6.10 SUMMARY

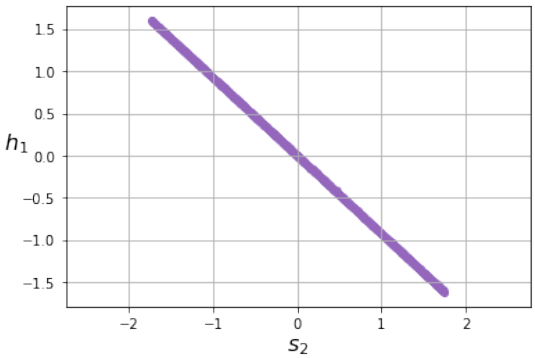
As already mentioned, the presented experiments were designed and executed in order to allow a better understanding of the potential and the limitations of ANICA in



(a) Gaussian source and code.



(b) Laplacian source and estimate.

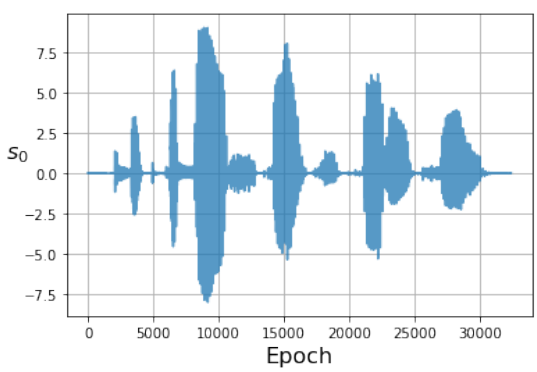


(c) Uniform source and estimate.

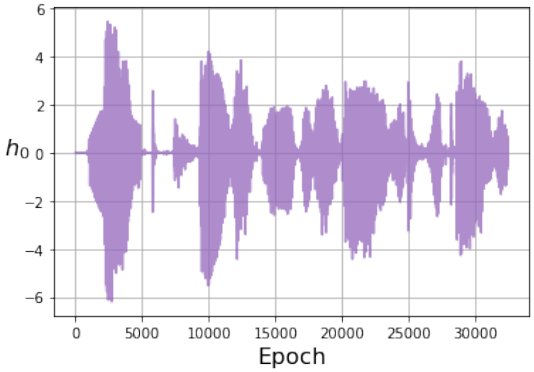
Figure 6.15 – Experiment 6: Best correspondence among signals.

different scenarios. Having in view the obtained results, we consider that ANICA yields quite interesting results, overcoming in certain cases the classical algorithms FastICA and JADE, albeit it usually requires a significantly higher number of samples to converge.

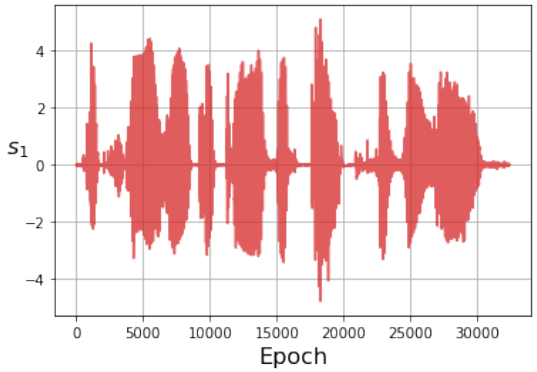
Interestingly, the main ideas of ANICA seem to be directly applicable to other BSS environments. By properly modifying the architecture of the AE, ANICA may be adapted to address different scenarios of the BSS problem, such as those involving non-linear mixtures and convolutive mixtures. In addition, other properties regarding the code variables could be encouraged, instead of the statistical independence, which could lead to different adversarial networks for similar tasks, such as sparse component analysis (SCA). So, we believe to still be far from the exhaustion of all the advantages ANICA has to offer.



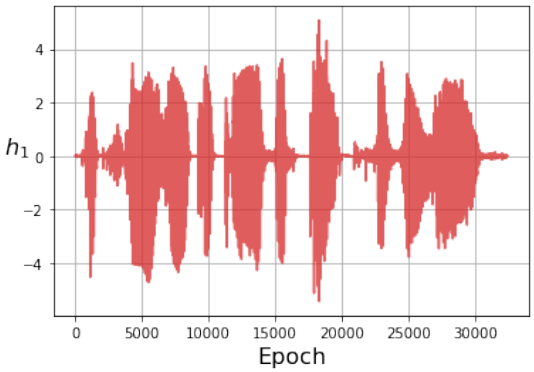
(a) Source signal.



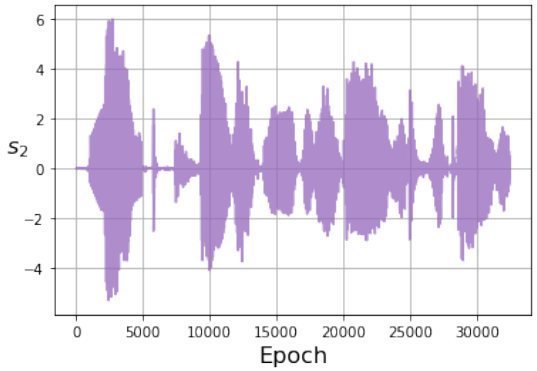
(b) Code.



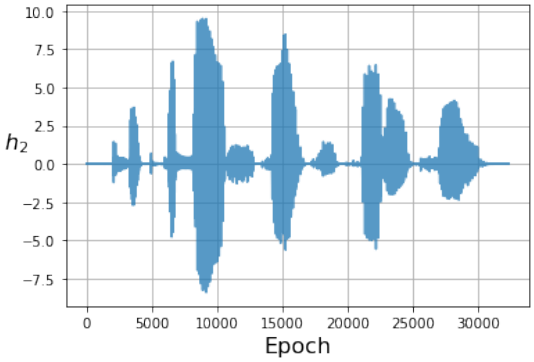
(c) Source signal.



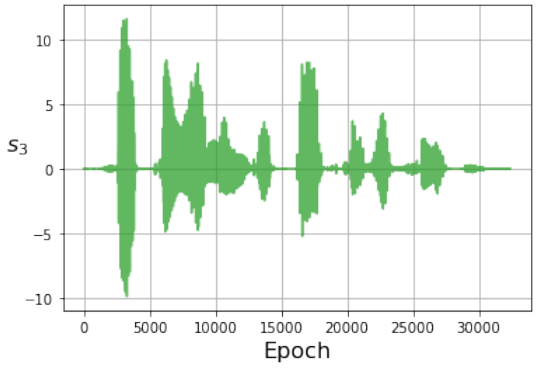
(d) Code.



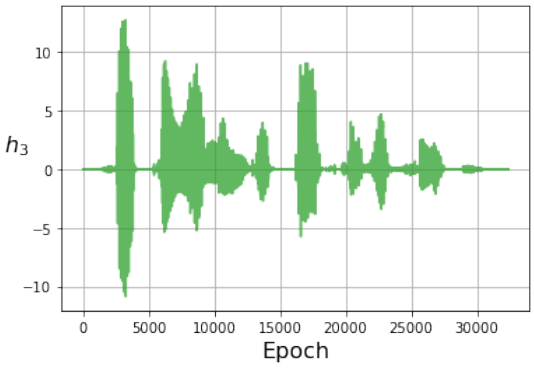
(e) Source signal.



(f) Code.

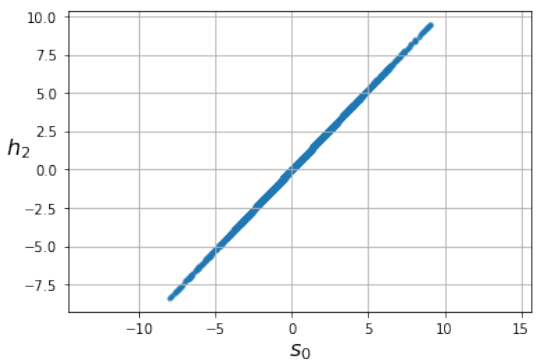


(g) Source signal.

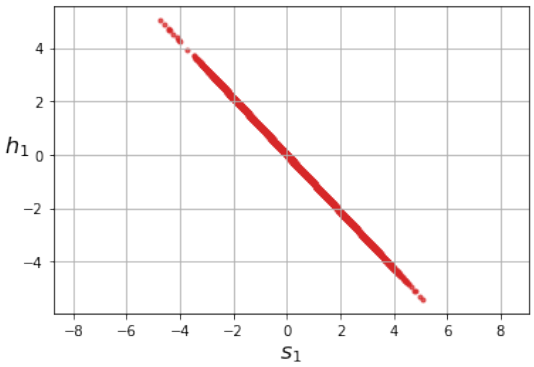


(h) Code.

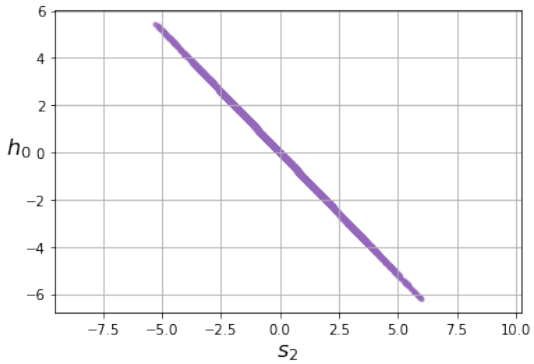
Figure 6.16 – Experiment 7: Audio signals separation with ANICA.



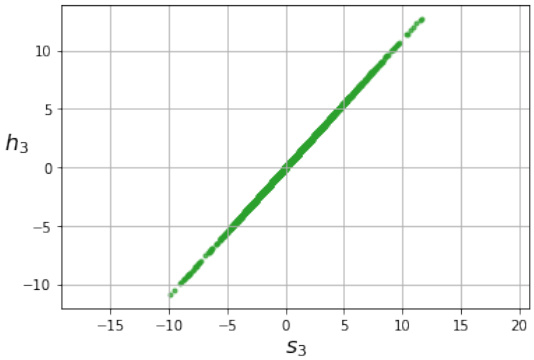
(a) Source and corresponding code.



(b) Source and corresponding code.



(c) Source and corresponding code.



(d) Source and corresponding code.

Figure 6.17 – Experiment 7: Best correspondence among audio signals.

7 CONCLUSIONS

As the results of our experiments demonstrate, ANICA has proven to be an effective ICA algorithm for challenging scenarios. In the cases of many sources, (Section 6.4), and a noisy dataset (Section 6.6), its performance is comparable to that of well-established algorithms, such as FastICA and JADE, once there is a large enough dataset onto which to train the model, as Section 6.3 makes explicit in its results.

The access to a large number of samples is crucial for ANICA in order to help the adversarial costs converge faster, and in a more stable manner, towards the ideal value of $\log 2$, as proven in Section 5.2.1. Also, it helps to mitigate difficulties related to initialization and the selection of hyperparameters. Nonetheless, convergence is never guaranteed due to the effects of the random initialization of the parameters of ANICA. As has been pointed out in most experiments, there is a wide variation among the lowest and highest NAE values, which is not the case for FastICA and JADE, which offer more consistent results. Notwithstanding, even when the dataset is relatively small, training for more epochs and reducing the size of the minibatch may lead to an adequate separation matrix in the encoder.

With no intention of minimizing the problem of non-convergence, it is valid to train ANICA multiple times, on the same set of mixtures, and yet retrieve very interesting results with the aid of the proposed blind metrics. As was performed in Section 6.7, for the overdetermined case, we simply removed the worst training instances based on the SMI they attained, for each number of additional components. Thus, in a completely blind manner, we discarded the training instances that did not attain favourable results and, as a consequence, there only remained the most robust instances.

The importance of the decoder was undoubtedly demonstrated in Section 6.5. There, we simply eliminated the decoder, such that the encoder and discriminator were connected in an architecture almost identical to GANs. However, the dynamic between such components is not exactly the same, due to the "inverse" mapping the encoder tries to learn, i.e., the encoder creates a latent code from existing observable mixtures, while the generator in a GAN attempts to create new "real" samples from a latent noise. Additionally, although it seemed to be that the reconstruction task of the decoder did not help the whole system to retrieve independent components, the results indicate otherwise. Once the decoder is removed, the performance of ANICA worsens more quickly the more components it has to retrieve in comparison to its original behavior reported in Section 6.4. Our hypothesis is that the joint training of the encoder and decoder maintains the code within the space defined by PCA, and similarly to AAEs, they are both equally important

to shape the distribution of the latent code space.

The adoption of the SMI, MNK and NM did not only proved to be useful as criteria for blindly selecting an epoch with a corresponding low NAE, but it also confirms the success of the AE to generate a code of independent components, rather than just learning a trivial solution for reconstruction.

Finally, it is important to highlight that ANICA offers a flexible architecture that can be straightforwardly modified to deal with other BSS scenarios, as well as those involving convolutive mixtures and nonlinear models, besides PNL. It is our belief that these cases will benefit, to a more significant extent, from the generality of the discussed approach. It even seems attainable a solution for the underdetermined case ($N < M$), under its adversarial framework. These are key topics for future investigations.

Although ANICA has proven to be an effective algorithm for ICA, it is also computationally expensive, requiring many hours for the completion of every experiment (including the 10 repetitions and varying samples, sources or noise). We initially speculated that forcing the parameters of the encoder and decoder to be always close to orthogonal matrices would reduce the required number of training epochs due to the reduction on the search space to yield independent components. As already explained in Section 2.4.3, whitening is invariant to orthogonal transformations. Thus, for an input set of whitened mixtures, a proper separation matrix could be found within the orthogonal space. Unfortunately, the orthogonalization steps implemented over the course of this dissertation did not offer any significant improvement, quite the opposite. Forcing an orthogonalization step on the parameters, which are continuously adapted by gradient descent to optimize a cost function, only delayed convergence or made it impossible to be attained because each orthogonalization step was, in a certain sense, equivalent to forgetting the information just learnt. A hope for a more natural orthogonalization process, such that it does not changes abruptly the encoder matrix, resides on a technique called Tying weights, as explained in Example 4.8, which we did not were able to implement until the writing process of this dissertation.

BIBLIOGRAPHY

ALPAYDIN, E. *Introduction to Machine Learning*. 2. ed. [S.l.]: The MIT Press, 2010. ISBN 026201243X. Cited in page 48.

AMARI, S.; CICHOCKI, A.; YANG, H. H. A New Learning Algorithm for Blind Signal Separation. In: TOURETZKY, D.; MOZER, M. C.; HASSELMO, M. (Ed.). *Advances in Neural Information Processing Systems*. MIT Press, 1996. v. 8, p. 757–763. Available from Internet: <https://proceedings.neurips.cc/paper/1995/file/e19347e1c3ca0c0b97de5fb3b690855a-Paper.pdf>. Cited 2 times in pages 71 and 72.

ARBERET, S.; GRIBONVAL, R.; BIMBOT, F. A Robust Method to Count and Locate Audio Sources in a Stereophonic Linear Instantaneous Mixture. *Proc. of the Int'l. Workshop on Independent Component Analysis and Blind Signal Separation (ICA 2006)*, mar. 2006. Cited in page 34.

BACK, A. D.; WEIGEND, A. S. A First Application of Independent Component Analysis to Extracting Structure from Stock Returns. *International Journal of Neural Systems*, v. 08, n. 04, p. 473–484, 1997. Available from Internet: <https://doi.org/10.1142/S0129065797000458>. Cited in page 37.

BELL, A. J.; SEJNOWSKI, T. J. An Information-Maximization Approach to Blind Separation and Blind Deconvolution. *Neural Computation*, v. 7, n. 6, p. 1129–1159, nov. 1995. ISSN 0899-7667. Available from Internet: <https://doi.org/10.1162/neco.1995.7.6.1129>. Cited in page 40.

BELOUCHRANI, A. et al. A blind source separation technique using second-order statistics. *IEEE Transactions on Signal Processing*, v. 45, n. 2, p. 434–444, 1997. Available from Internet: <https://ieeexplore.ieee.org/document/554307>. Cited in page 33.

BISHOP, C. *Pattern Recognition and Machine Learning*. 1. ed. [S.l.]: Springer-Verlag New York, 2006. (Information Science and Statistics). Cited in page 48.

BRAKEL, P.; BENGIO, Y. *Learning Independent Features with Adversarial Nets for Non-linear ICA*. 2017. Available from Internet: <https://arxiv.org/abs/1710.05050>. Cited 9 times in pages 8, 9, 17, 48, 64, 65, 67, 71, and 72.

CARDOSO, J.-F. Blind signal separation: statistical principles. *Proceedings of the IEEE*, v. 86, n. 10, p. 2009–2025, 1998. Cited in page 40.

CARDOSO, J. F.; SOULOUMIAC, A. Blind beamforming for non-gaussian signals. *IEE Proceedings F (Radar and Signal Processing)*, v. 140, p. 362–370(8), dez. 1993. ISSN 0956-375X. Available from Internet: <https://digital-library.theiet.org/content/journals/10.1049/ip-f-2.1993.0054>. Cited 2 times in pages 39 and 45.

CASTELLA, M.; CHEVREUIL, A.; PESQUET, J.-C. Chapter 8 - Convolutive mixtures. In: COMON, P.; JUTTEN, C. (Ed.). *Handbook of Blind Source Separation*. Oxford: Academic Press, 2010. p. 281–324. ISBN 978-0-12-374726-6. Available from Internet: <https://www.sciencedirect.com/science/article/pii/B9780123747266000138>. Cited in page 21.

CASTRO, L. de; ZUBEN, F. V. Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 3, p. 239–251, 2002. Cited in page 70.

CHERRY, E. C. Some Experiments on the Recognition of Speech, with One and with Two Ears. *The Journal of the Acoustical Society of America*, v. 25, n. 5, p. 975–979, 1953. Available from Internet: <https://doi.org/10.1121/1.1907229>. Cited in page 37.

COMON, P. Independent Component Analysis. In: LACOUME, J.-L. (Ed.). *Higher-Order Statistics*. Elsevier, 1992. p. 29–38. Available from Internet: <https://hal.archives-ouvertes.fr/hal-00346684>. Cited 2 times in pages 27 and 28.

COMON, P. Independent component analysis, A new concept? *Signal Processing*, v. 36, n. 3, p. 287–314, 1994. ISSN 0165-1684. Higher Order Statistics. Available from Internet: <https://www.sciencedirect.com/science/article/pii/0165168494900299>. Cited 2 times in pages 27 and 28.

COMON, P.; CARDOSO, J. F. Eigenvalue Decomposition of a Cumulant Tensor with Applications. *Proceedings of SPIE - The International Society for Optical Engineering*, v. 1348, p. 361–372, nov. 1990. Cited in page 46.

COVER, T. M.; THOMAS, J. A. *Elements of Information Theory*. 2. ed. Hoboken, USA: Wiley-Interscience, 2006. v. 2nd. (Wiley Series in Telecommunications and Signal Processing, 3). Cited 2 times in pages 39 and 42.

DIAMANTARAS, K. I.; KUNG, S. Y. *Principal Component Neural Networks: Theory and Applications*. 1. ed. USA: John Wiley & Sons, Inc., 1996. (Adaptive and Cognitive Dynamic Systems: Signal Processing"" Learning"" Communications and Control). ISBN 0471054364. Cited in page 54.

DOUKAS, N.; KARADIMAS, N. V. A Blind Source Separation Based Cryptography Scheme for Mobile Military Communication Applications. *WTOC*, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, v. 7, n. 12, p. 1235–1245, dez. 2008. ISSN 1109–2742. Cited in page 37.

DUA, D.; GRAFF, C. *UCI Machine Learning Repository*. 2017. Available from Internet: <http://archive.ics.uci.edu/ml>. Cited in page 53.

DUARTE, L. T. et al. Blind Source Separation of Post-nonlinear Mixtures Using Evolutionary Computation and Order Statistics. In: ROSCA, J. et al. (Ed.). *Independent Component Analysis and Blind Signal Separation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 66–73. ISBN 978-3-540-32631-1. Cited in page 70.

DUDA R. O. HART, P. E.; STORK, G. *Pattern Classification*. 2. ed. [S.l.]: John Wiley & Sons, 2000. Cited in page 50.

FAN, W.; ZHANG, X.; JIANG, B. A New Passive Sonar Bearing Estimation Algorithm Combined with Blind Source Separation. In: *2010 Third International Joint Conference on Computational Science and Optimization*. [s.n.], 2010. v. 1, p. 15–18. Available from Internet: <https://ieeexplore.ieee.org/abstract/document/5532922>. Cited in page 37.

GLOROT, X.; BENGIO, Y. Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, PMLR*, v. 9, p. 249–256, 2010. Available from Internet: <http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf?hc_location=ufi]>. Cited in page 72.

GOODFELLOW, I. *NIPS 2016 Tutorial: Generative Adversarial Networks*. 2017. Cited in page 59.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Cited 7 times in pages 48, 49, 51, 52, 53, 54, and 55.

GOODFELLOW, I. et al. Generative Adversarial Nets. In: GHAHRAMANI, Z. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014. v. 27, p. 2672–2680. Available from Internet: <https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf>. Cited 6 times in pages 48, 57, 58, 59, 65, and 68.

GRIBONVAL, R.; ZIBULEVSKY, M. Chapter 10 - Sparse component analysis. In: COMON, P.; JUTTEN, C. (Ed.). *Handbook of Blind Source Separation*. Oxford: Academic Press, 2010. p. 367–420. ISBN 978-0-12-374726-6. Available from Internet: <https://www.sciencedirect.com/science/article/pii/B9780123747266000151>. Cited in page 35.

GéRON, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. 2. ed. [S.l.]: O'Reilly Media, Inc., 2019. ISBN 9781492032649. Cited 9 times in pages 48, 49, 51, 54, 55, 57, 58, 60, and 62.

HAYKIN, S.; CHEN, Z. The Cocktail Party Problem. *Neural Computation*, v. 1875–1902, n. 9, set. 2005. Available from Internet: <https://doi.org/10.1162/0899766054322964>. Cited in page 37.

HE, Z. et al. K-hyperline clustering learning for sparse component analysis. *Signal Processing, Elsevier*, v. 89, n. 6, p. 1011–1022, 2009. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0165168408003836>. Cited in page 34.

HYVäRINEN, A. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, v. 10, n. 3, p. 626–634, maio 1999. Available from Internet: <https://ieeexplore.ieee.org/abstract/document/761722?casa_token=wnLI-tBR1H8AAAAA:X45kYVKNBkVzpJe3oRORbgGfvfy2c5opxR-DFA_S_s33121MNXVHUQHtv4CGZH-tm8vPzdM0SA>. Cited 2 times in pages 39 and 43.

HYVäRINEN, A.; KARHUNEN, J.; OJA, E. *Independent Component Analysis*. John Wiley & Sons, Ltd, 2001. ISBN 9780471221319. Available from Internet: <https://onlinelibrary.wiley.com/doi/abs/10.1002/0471221317>. Cited 7 times in pages 25, 36, 39, 41, 43, 45, and 46.

HYVäRINEN, A.; OJA, E. A fast fixed-point algorithm for independent component analysis. *Neural computation, MIT Press*, v. 9, n. 7, p. 1483–1492, jul. 1997. ISSN 0899-7667. Available from Internet: <https://www.mitpressjournals.org/doi/abs/10.1162/neco.1997.9.7.1483>. Cited 2 times in pages 39 and 43.

IOFFE, S.; SZEGEDY, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: BACH, F.; BLEI, D. (Ed.). *Proceedings of the 32nd International Conference on Machine Learning*. Lille, France: PMLR, 2015. (Proceedings of Machine Learning Research, v. 37), p. 448–456. Available from Internet: <http://proceedings.mlr.press/v37/ioffe15.html>. Cited in page 67.

JAFARI, M. G.; CHAMBERS, J. A. Fetal electrocardiogram extraction by sequential source separation in the wavelet domain. *IEEE Transactions on Biomedical Engineering*, v. 52, n. 3, p. 390–400, 2005. Available from Internet: <https://ieeexplore.ieee.org/document/1396379>. Cited in page 37.

JOHO, M.; MATHIS, H.; LAMBERT, R. H. Overdetermined Blind Source Separation: Using More Sensors Than Source Signals In A Noisy Mixture. In: *in Proc. International Conference on Independent Component Analysis and Blind Signal Separation*. [s.n.], 2000. p. 81–86. Available from Internet: <https://www.researchgate.net/publication/2635667_Overdetermined_Blind_Source_Separation_Using_More_Sensors_Than_Source_Signals_In_A_Noisy_Mixture>. Cited 2 times in pages 29 and 81.

JUTTEN, C. *Calcul neuromimétique et traitement du signal: analyse en composantes indépendantes*. Tese (Doutorado) — Grenoble INPG, 1987. Cited in page 27.

JUTTEN, C.; BABAIE-ZADEH, M.; KARHUNEN, J. Chapter 14 - Nonlinear mixtures. In: COMON, P.; JUTTEN, C. (Ed.). *Handbook of Blind Source Separation*. Oxford: Academic Press, 2010. p. 549–592. ISBN 978-0-12-374726-6. Available from Internet: <https://www.sciencedirect.com/science/article/pii/B9780123747266000199>. Cited in page 70.

JUTTEN, C.; COMON, P. Chapter 1 - Introduction. In: COMON, P.; JUTTEN, C. (Ed.). *Handbook of Blind Source Separation*. Oxford: Academic Press, 2010. p. 1–22. ISBN 978-0-12-374726-6. Available from Internet: <https://www.sciencedirect.com/science/article/pii/B9780123747266000060>. Cited in page 23.

JUTTEN, C.; HERAUT, J. Independent Components Analysis versus Principal Components Analysis. *Signal Processing IV, Theories and Applications EUSIPCO'88*, v. 2, p. 643–646, set. 1988. Cited in page 27.

JUTTEN, C.; TALEB, A. Source separation: From dusk till dawn. *Proc. 2nd Int. Workshop on Independent Component Analysis and Blind Source Separation, ICA2000*, p. 15–26, 2000. Available from Internet: <https://www.researchgate.net/publication/2399949_Source_Separation_From_Dusk_Till_Dawn>. Cited in page 27.

KABAL, P. Tsp speech database. McGill University, Database Version. set. 2002. Available from Internet: <http://www-mmsp.ece.mcgill.ca/Documents/Downloads/TSPspeech/TSPspeech.pdf>. Cited in page 83.

KAGAN, A. M.; LINNIK, Y. V.; RAO, C. R. *Characterization problems in mathematical statistics*. 1. ed. [S.l.]: Wiley, 1973. (Wiley series in probability and mathematical statistics). Cited in page 30.

KAY, S. M. *Intuitive Probability and Random Processes Using MATLAB*. 1. ed. [S.l.]: Springer US, 2006. (Communications Engineering, Networks). ISBN 978-0-387-24157-9. Cited in page 19.

KIM, S. G.; YOO, C. D. Underdetermined independent component analysis by data generation. In: . [S.l.: s.n.], 2004. p. 445–452. ISBN 978-3-540-30110-3. Cited in page 23.

KIM, S. G.; YOO, C. D. Underdetermined blind source separation based on subspace representation. *IEEE Transactions on Signal Processing*, v. 57, n. 7, p. 2604–2614, 2009. Cited in page 23.

KINGMA, D. P.; WELLING, M. Auto-Encoding Variational Bayes. In: BENGIO, Y.; LECUN, Y. (Ed.). *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. [s.n.], 2014. Available from Internet: <http://arxiv.org/abs/1312.6114>. Cited in page 56.

KIVILUOTO, K.; OJA, E. Independent Component Analysis for Parallel Financial Time Series. In: *International Conference on Neural Information Processing (ICONIP'98), Kitakyushu, Japan, October 21-23, 1998*. [S.l.: s.n.], 1998. p. 895–898. Cited in page 37.

KNUTH, K. H. A bayesian approach to source separation. *Proceedings of the First International Workshop on Independent Component Analysis and Signal Separation: ICA'99, Aussois, France*, p. 283–288, jan. 1999. Available from Internet: <https://arxiv.org/abs/physics/0205032>. Cited in page 35.

LATHAUWER, L. D.; MOOR, D. D.; VANDEWALLE, J. Fetal electrocardiogram extraction by blind source subspace separation. *IEEE Transactions on Biomedical Engineering*, v. 47, n. 5, p. 567–572, 2000. Available from Internet: <https://ieeexplore.ieee.org/abstract/document/841326>. Cited in page 37.

LECUN, Y.; CORTES, C.; BURGES, C. MNIST handwritten digit database. 2010. Available from Internet: <http://yann.lecun.com/exdb/mnist/>. Cited in page 58.

LEI, P.; CHEN, M.; WANG, J. Speech enhancement for in-vehicle voice control systems using wavelet analysis and blind source separation. *IET Intelligent Transport Systems*, Institution of Engineering and Technology, v. 13, p. 693–702(9), abr. 2019. ISSN 1751-956X. Available from Internet: <https://digital-library.theiet.org/content/journals/10.1049/iet-its.2018.5094>. Cited in page 37.

LU, C.-J.; LEE, T.-S.; CHIU, C.-C. Financial time series forecasting using independent component analysis and support vector regression. *Decision Support Systems*, v. 47, n. 2, p. 115–125, 2009. ISSN 0167-9236. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0167923609000323>. Cited in page 38.

LUO, Z.; LI, C.; ZHU, L. A Comprehensive Survey on Blind Source Separation for Wireless Adaptive Processing: Principles, Perspectives, Challenges and New Research Directions. *IEEE Access*, PP, p. 1–26, nov. 2018. Cited in page 37.

MAKHZANI, A. et al. *Adversarial Autoencoders*. 2016. Available from Internet: <http://arxiv.org/abs/1511.05644>. Cited 5 times in pages 48, 60, 61, 62, and 64.

MINDIKOGLU, A. F.; VEEN, A. Van der. Separation of overlapping RFID signals by antenna arrays. In: *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. [S.l.: s.n.], 2008. p. 2737–2740. Cited in page 37.

MOHAMMAD-DJAFARI, A. A bayesian approach to source separation. *AIP Conference Proceedings*, v. 567, n. 1, p. 221–244, 2001. Available from Internet: <https://aip.scitation.org/doi/abs/10.1063/1.1381859>. Cited in page 35.

MOHAMMAD-DJAFARI, A.; KNUTH, K. H. Chapter 12 - Bayesian approaches. In: COMON, P.; JUTTEN, C. (Ed.). *Handbook of Blind Source Separation*. Oxford: Academic Press, 2010. p. 467–513. ISBN 978-0-12-374726-6. Available from Internet: <https://www.sciencedirect.com/science/article/pii/B9780123747266000175>. Cited in page 35.

MOVAHEDI, F. et al. Estimating the mixing matrix in Sparse Component Analysis (SCA) based on partial k-dimensional subspace clustering. *Neurocomputing for Vision Research Advances in Blind Signal Processing, Elsevier*, v. 71, n. 10, p. 2330–2343, 2008. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0925231208001033>. Cited in page 34.

O'GRADY, P. D.; PEARLMUTTER, B. A.; RICKARD, S. T. Survey of sparse and non-sparse methods in source separation. *International Journal of Imaging Systems and Technology*, v. 15, n. 1, p. 18–33, 2005. Available from Internet: <https://onlinelibrary.wiley.com/doi/abs/10.1002/ima.20035>. Cited in page 35.

PAPOULIS, A.; PILLAI, S. U. *Probability, Random Variables, and Stochastic Processes*. 4. ed. [S.l.]: McGrawHill International, 2002. (McGraw-Hill Higher Education). ISBN 0-07-366011-6. Cited 8 times in pages 18, 19, 24, 27, 28, 35, 40, and 45.

PLUMBLEY, M. D. et al. Automatic Music Transcription and Audio Source Separation. *Cybernetics and Systems*, Taylor & Francis, v. 33, n. 6, p. 603–627, 2002. Available from Internet: <https://doi.org/10.1080/01969720290040777>. Cited in page 37.

PRINCEN, J.; BRADLEY, A. Analysis/synthesis filter bank design based on time domain aliasing cancellation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 34, n. 5, p. 1153–1161, 1986. Cited in page 35.

RADFORD, A.; METZ, L.; CHINTALA, S. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 2016. Cited in page 58.

ROMANO, J. et al. *Unsupervised signal Processing: Channel Equalization and Source Separation*. 1st. ed. USA: CRC Press, 2011. Cited 2 times in pages 33 and 41.

RUMELHART, D.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, v. 323, p. 533–536, 1986. Cited in page 49.

SANEI, S. et al. Incorporating frequency warping into sparse component analysis. In: *2005 13th European Signal Processing Conference*. [S.l.: s.n.], 2005. p. 1–4. Cited in page 35.

SHLENS, J. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014. Available from Internet: <https://arxiv.org/abs/1404.1100>. Cited in page 25.

SUYAMA, R. *Proposta de m´etodos de separa¸c~ao cega de fontes para misturas convolutivas e n~ao-lineares*. Tese (Doutorado) — FEEC-UNICAMP, 2007. Cited in page 28.

TALEB, A.; JUTTEN, C. Nonlinear source separation: the post-nonlinear mixtures. In: VERLEYSEN, M. (Ed.). *ESANN 1997, 5th Eurorean Symposium on Artificial Neural Networks, Bruges, Belgium, April 16-18, 1997, Proceedings*. [S.l.]: D-Facto public, 1997. p. 279–284. Cited in page 70.

TALEB, A.; JUTTEN, C. Source separation in post-nonlinear mixtures. *IEEE Transactions on Signal Processing*, v. 47, n. 10, p. 2807–2820, 1999. Cited in page 70.

TIELEMAN, T.; HINTON, G. *Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude*. 2012. 26–31 p. Available from Internet: <https://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6e.mp4>. Cited in page 72.

TONG, L. et al. AMUSE: A New Blind Identification Algorithm. *IEEE International Symposium on Circuits and Systems*, v. 3, n. 1784–1787, 1990. Available from Internet: <https://ieeexplore.ieee.org/document/111981>. Cited in page 32.

VINCENT, P. et al. Extracting and Composing Robust Features with Denoising Autoencoders. In: . New York, NY, USA: Association for Computing Machinery, 2008. (ICML '08), p. 1096—-1103. ISBN 9781605582054. Available from Internet: <https://doi.org/10.1145/1390156.1390294>. Cited in page 55.

VINCENT, P. et al. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research*, v. 11, n. 110, p. 3371–3408, 2010. Available from Internet: <http://jmlr.org/papers/v11/vincent10a.html>. Cited in page 55.

WERBOS, P. J. Applications of advances in nonlinear sensitivity analysis. In: DRENICK, R. F.; KOZIN, F. (Ed.). *System Modeling and Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1982. p. 762–770. ISBN 978-3-540-39459-4. Cited in page 49.

YEREDOR, A. Chapter 7 - Second-order methods based on color. In: COMON, P.; JUTTEN, C. (Ed.). *Handbook of Blind Source Separation*. Oxford: Academic Press, 2010. p. 227–279. ISBN 978-0-12-374726-6. Available from Internet: <https://www.sciencedirect.com/science/article/pii/B9780123747266000126>. Cited in page 33.

YI, T.-H. et al. Clustering Number Determination for Sparse Component Analysis during Output-Only Modal Identification . *Journal of Engineering Mechanics*, v. 145, n. 1, 2019. Available from Internet: <https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29EM.1943-7889.0001557>. Cited in page 34.

ZARZOSO, V.; NANDI, A. K. Noninvasive fetal electrocardiogram extraction: blind separation versus adaptive noise cancellation. *IEEE Transactions on Biomedical Engineering*, v. 48, n. 1, p. 12–18, 2001. Available from Internet: <https://ieeexplore.ieee.org/document/900244>. Cited in page 37.

ZIEHE, A.; MULLER, K. R. TDSEP—An efficient algorithm for blind separation using time structure. *Proceedings of the International Conference on Artificial Neural Networks (ICANN '98), Springer*, p. 675–680, 1998. Available from Internet: <https://link.springer.com/chapter/10.1007/978-1-4471-1599-1_103>. Cited in page 33.