A Distance-Based Approach to Independent Component Analysis

Debopriya Basu

A Thesis

in

The Department

of

Mathematics and Statistics

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Mathematics) at

Concordia University

Montréal, Québec, Canada

August 2025

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Mr. Debopriya Basu**

Entitled: **A Distance-Based Approach to Independent Component Analysis**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Mathematics)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. S. Chauhan

_____ External Examiner
Dr. A. Khalili

_____ Arms-Length Examiner
Dr. W. Sun

_____ Examiner
Dr. Y. Chaubey

_____ Examiner
Dr. F. Godin

_____ Thesis Supervisor
Dr. A. Sen

Approved by _____
Dr. C. Hyndman, Graduate Program Director
Department of Mathematics and Statistics

_____ 2025

_____
Dr. P. Sicotte, Dean
Faculty of Arts and Science

Abstract

A Distance-Based Approach to Independent Component Analysis

Debopriya Basu, Ph.D.

Concordia University, 2025

Independent Component Analysis (ICA) is a widely used statistical technique for decomposing multivariate signals (mixtures) into their underlying (non-Gaussian) independent components. Mathematically, ICA models observations $\mathbf{Y} \in \mathbb{R}^d, d \geq 2$, as a mixture of unobserved independent source components $\mathbf{X} \in \mathbb{R}^d$, via an unknown nonsingular mixing matrix $\mathbf{A}$, namely, $\mathbf{Y} = \mathbf{AX}$. The goal of ICA is to estimate the unmixing matrix $\mathbf{B} = \mathbf{A}^{-1}$ based on IID data $\mathbf{Y}_i = \mathbf{AX}_i, i = 1, \ldots, n$; to separate the mixed signals into the underlying independent components.

Our work proposes a novel distance-based approach for estimating $\mathbf{B}$. The estimation is performed by minimizing the distance ρ_w between the joint empirical distribution function of $\mathbf{X}_1, \ldots, \mathbf{X}_n$, and the marginal empirical distribution functions of the coordinates. We establish that ρ_w is asymptotically a U-statistic and derive its theoretical properties. Further, we analyze the empirical process to derive an estimation strategy for $\mathbf{B}$. We devise two separate methods to construct confidence intervals. The first one uses the U-statistic and a specialized weight function that makes it independent of the distributions of the sources. The second method relies on the principal components of the empirical process. To make this approach computationally feasible, we propose a Gradient Descent Algorithm (GDA) to compute the estimate, and demonstrate its effectiveness by comparing it to the prevalent FastICA method (cf. Hyvärinen and Oja (2000)).

This work contributes both theoretically and numerically to the field of ICA by introducing a new approach to the estimation problem as well as providing a practical algorithmic implementation procedure.

Acknowledgments

I would like to begin by expressing my deepest gratitude to my advisor, Prof. Arusharka Sen, whose mentorship, encouragement, and patience over these past years have been instrumental in shaping this dissertation. His guidance through both technical challenges and moments of uncertainty has made this journey not only possible but also deeply fulfilling. I am incredibly fortunate to have had him as my advisor.

I am equally grateful to my dissertation committee: Prof. Yogendra P. Chaubey, Prof. Frédéric Godin, and Prof. Wei Sun. Their expertise, thoughtful questions, and generous feedback helped refine the arguments presented here and expanded my thinking in important ways. I am particularly thankful to Prof. Abbas Khalili (as External Examiner) and Prof. Satyaveer S. Chauhan (as Chair) for agreeing to be on my committee at short notice.

To the faculty at Concordia University, I shall forever remain indebted for shaping my education both inside and outside the classroom. Profs. Lisa Kakinami and Debraj Sen have played a key role in helping me grow as a researcher, with their engaging lectures, valuable advice, and most importantly, faith in my abilities. I am grateful to our department staff, especially Carmen Buffone and Judy Thykootathil, for always being willing to help manage my academic and non-academic responsibilities.

To my professors and mentors from the University of Calcutta, chiefly Prof. Asis Kumar Chattopadhyay and Prof. Sugata Sen-Roy, thank you for first sparking my interest in this field and for encouraging me to pursue advanced study. I am particularly grateful to Prof. Uttam Bandyopadhyay and Prof. Tanuka Chattopadhyay, whose passion for teaching left a lasting impression on me. Though they are no longer with us, their guidance and encouragement continue to resonate in my

work and aspirations.

I could not have done any of this without the unwavering love and support of my family. To my parents, Sadhana Basu and Sujit Kumar Basu, thank you for always believing in me. Your sacrifices and encouragement have carried me farther than words can express. To the man I credit for my academic journey — Dr. Samopriya Basu — I am proud to call you my brother. Everything I have achieved in academics has, in some way, been an attempt to measure up to your brilliance. You are the benchmark against which I've challenged myself time and again. And to my extended family — grandparents, uncles, aunts, cousins — thank you all for cheering me on from afar. A special remembrance goes to my grandfather, granduncle, and grandaunt, whom I have lost over the last few years. The Gods know how happy they would be to see this day.

Here in Montréal, I am grateful for the incredible community of friends and colleagues who made daily life during the PhD manageable and full of joy. My colleagues over the years, Grace (my first friend here at Concordia), Emmanuel, Magloire, Samantha, Paul, Sina, Giovanni, and many others, have been like family through it all. From office hours and Math Help Center sessions to academic discussions — you made this journey less solitary and, for want of a better word, human. A heartfelt thanks to the "boys" from other departments who started their Concordia journey with me, Siddharth(s), Subham, and Rishit.

To my batchmates at Calcutta University, particularly Agnideep, Amarnath, Shubhankar, Chiranjib, and Debayan, thank you for being the exact people you are. Don't ever change. I would also like to thank my broader circle of friends across the world, especially Fan Yang, and my gaming buddies — most of whom I've never met in person and whose real names I still don't know. Your late-night conversations, in-game banter, and quiet (and sometimes vocal) support helped me unwind during some of the most stressful stretches of this journey. Thank you for being there, in your own unique way.

Finally, to everyone — named or unnamed — who has offered a kind word, provided advice (academic or otherwise), proofread a paragraph, or simply stood by me during this long journey: thank you. This dissertation may bear my name, but it is the product of many hands, hearts, and minds.

"Non nobis solum nati sumus."

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Independent Component Analysis (ICA) has emerged as a powerful technique in fields like signal processing, neuroscience, and even finance, where extracting signals or components from mixed data is essential. This method is especially valuable when dealing with statistically independent sources but not observable directly, often modeled as latent variables. In this chapter, we introduce ICA, illustrate its foundational principles, mathematical models, and challenges encountered in its framework. We also discuss a few established methods with a focus on estimation. We end the chapter with the proposal for a new method for estimating the unknown source signals modeled under the ICA setup.

1.1 ICA Model Overview

Following the monograph on ICA by Hyvärinen, Karhunen, and Oja (2001), we begin with the classical illustrative example, commonly known as *The Cocktail Party Problem*. Imagine a scenario where two individuals are conversing simultaneously in a room, with two microphones positioned at different locations in the room to capture their speech. The signals recorded by the microphones are indicated as $y_1(t)$ and $y_2(t)$, where $y_i;\ i = 1, 2$; represent the amplitudes and t represents the time index. Each of these recorded signals is assumed to be a weighted linear combination, in other words a *mixture*, of the *source* signals, denoted as $x_i(t);\ i = 1, 2$. Mathematically, this relation can

be represented as a system of two linear equations in two variables:

$$\left.\begin{aligned} y_1(t) &= a_{11}x_1(t) + a_{12}x_2(t) \\ y_2(t) &= a_{21}x_1(t) + a_{22}x_2(t) \end{aligned}\right\} \tag{1}$$

or more compactly in matrix form:

$$\mathbf{Y} = \mathbf{A}\mathbf{X} \tag{2}$$

The matrix $\mathbf{A} = (a_{ij})_{i,j=1,2}$; is called the *mixing matrix*, and its entries are parameters that are typically dependent on factors such as the distances between the microphones and the speakers. To work within a simplified framework while keeping the model tractable, we will adhere to the classical setup, ignoring additional complexities in the model like temporal factors.

In general, the ICA setup is a **latent variables** model where we observe d random variables $Y_1, \ldots, Y_d$, each modeled as a linear combinations of d latent, statistically independent random variables $X_1, \ldots, X_d$. It is also a **generative** model as it describes how the observed data arise via linear mixing from underlying independent sources.

The relationship between observed mixtures and source components is written as:

$$\mathbf{Y} = \mathbf{A}\mathbf{X} = \sum_j \mathbf{a}_j X_j; \tag{3}$$

where $\mathbf{X} = (X_1, \ldots, X_d)^\top$ is a vector of independent but latent[1] components. These *sources* are called the *Independent Components* (IC). The matrix $\mathbf{A}$ is the *mixing matrix*, with unknown entries a_{ij}, and the j^{th} column of $\mathbf{A}$ denoted by $\mathbf{a}_j$.

The goal of ICA is to recover the original sources $\mathbf{X}$ from the observed mixtures $\mathbf{Y}$; in other words, to estimate the matrix $\mathbf{B} := \mathbf{A}^{-1}$ and to predict $\hat{\mathbf{X}} = \hat{\mathbf{B}}\mathbf{Y}$, where $\hat{\mathbf{B}}$ is an estimate of $\mathbf{B}$.

Let $f(x_1, \ldots, x_d)$ be the joint density function of the independent components of $\mathbf{X}$ and $f_j(x_j)$, $j = 1, \ldots, d$; be the marginal density functions. Then by independence assumption of the components of $X_1, ..., X_d$, the joint probability density function factorizes into the product of marginal

[1] They can't be observed directly.

densities

$$f(x_1, \ldots, x_d) = \prod_{j=1}^{d} f_j(x_j). \tag{4}$$

Since $\mathbf{X} = \mathbf{BY}$, let $\mathbf{b}_j^\top$ denote its rows. Consequently, the joint density of $Y_1, \ldots, Y_d$ is

$$f(\mathbf{y}) = |\det(\mathbf{B})| \prod_{j=1}^{d} f_j(\mathbf{b}_j^\top \mathbf{y}). \tag{5}$$

1.2 Basic Assumptions

The fundamental assumption of ICA revolves entirely around the independence of its components, i.e., each component in $\mathbf{X}$ is assumed to be statistically independent. It is also assumed that these components are non-Gaussian[2], or more precisely, at most one of the independent components can be assumed to have a Gaussian distribution. At least under these two assumptions, the ICA model is identifiable with some trivial indeterminacy (discussed in the next section). Additional assumptions are made as required regarding the distribution of these components.

Although not strictly necessary, assuming that the number of independent components equals the number of observed mixtures, i.e., the mixing matrix $\mathbf{A}$ is square, greatly simplifies the estimation procedure. This *square* linear system allows $\mathbf{Y} = \mathbf{AX}$ to be inverted (up to scaling and permutation) facilitating straightforward source recovery using linear algebra and simple statistical techniques. To that end, we assume that the square mixing matrix $\mathbf{A}$ is non-singular, i.e.,

$$\mathbf{X} = \mathbf{BY}, \;\; \mathbf{B} = \mathbf{A}^{-1}. \tag{6}$$

The matrix $\mathbf{B}$ will be a cornerstone in the discussions to follow, and will referred to as the *unmixing matrix*.

Non-square Systems

When the system isn't square, the number of mixtures m differs from the number of sources d, and from it two distinct scenarios arise:

[2]The rationale for excluding more than one Gaussian components is elaborated upon in Section 1.3.3.

- **Overcomplete case**: (more sources than mixtures, $d > m$) The system is underdetermined. There are infinitely many possible combinations of sources that could produce the observed mixtures. In this setting, additional assumptions have to be made (e.g., sparsity of sources) typically employ more sophisticated techniques such as sparse ICA or nonlinear methods to identify the components.

- **Undercomplete case**: (more sources than mixtures, $d < m$) The system is overdetermined. Although there is more information available than strictly necessary, standard ICA algorithms usually do not directly apply, and preprocessing such as dimensionality reduction (e.g., via PCA) is required to reduce the system to a square form before applying ICA.

1.3 Indeterminacies and Drawbacks

It is well known that the independent components model suffers from certain inherent identifiability problems. They are briefly discussed under the next three headings.

1.3.1 Indeterminacy of Variance (Scaling Ambiguity)

As neither the mixing matrix $\mathbf{A}$ nor the independent components $\mathbf{X}$ are known, the variance (or scale) of each component $X_j,\ j = 1, 2, ..., d$; can't be determined. This is because any scaling of the j^{th} independent component X_j can be exactly counteracted by an inverse scaling in $\mathbf{a}_j$, the j^{th} column of $\mathbf{A}$, by the same scalar, say k_j,

$$\mathbf{Y} = \sum_j \frac{\mathbf{a}_j}{k_j}\, k_j X_j.$$

Consequently, the ICA model can only identify the sources up to an arbitrary multiplicative constant. In practice, this is resolved by fixing the magnitudes of the independent components arbitrarily. Capitalizing on this, it is common to assume that each component has unit variance, i.e., $\mathbb{V}(X_j) = 1\ \forall j = 1, 2, \ldots, d$. Of course, matrix $\mathbf{A}$ is modified in the ICA solution methods to account for this additional constraint. However, it still leaves the **sign** problem — any independent component can be multiplied by -1 without affecting the model.

1.3.2 Lack of Natural Ordering

An inherent indeterminacy in ICA is the absence of a natural or meaningful order among the estimated independent components[3]. The ICA model treats all components symmetrically without prioritizing one over another. Mathematically, this corresponds to the fact that permuting the columns of the mixing matrix $\mathbf{A}$ and the corresponding entries of the source $\mathbf{X}$ does not change the mixture $\mathbf{Y} = \mathbf{AX}$. For any permutation matrix $\mathbf{P}$, one can express $\mathbf{Y} = \mathbf{APP}^{-1}\mathbf{X}$, with the entries of $\mathbf{P}^{-1}\mathbf{X}$ representing the components in a different order, while $\mathbf{AP}$ serves as the new mixing matrix. Any ICA algorithm capable of solving the original problem can be employed to solve the new permuted problem. Thus, ICA algorithms are only capable of estimating components up to a permutation.

In view of the discussions in Sections 1.3.1 and 1.3.2, we consider estimation of $\mathbf{B}$ using a loss function of the form $\mathcal{L}(\hat{\mathbf{B}}^{-1}\mathbf{B} - \mathbf{I})$. Note that such a loss is invariant under both component-wise change of scale and permutation of components. This is explored in Chapter 3, Section 3.3.

1.3.3 Limited to Non-Gaussian Components

A major limitation of ICA is its reliance on the non-Gaussianity of the independent components. When two or more[4] latent sources are Gaussian, the ICA model becomes fundamentally unidentifiable. This stems from the property that any linear combination of Gaussian variables remains Gaussian, and that multivariate Gaussian distributions are invariant under orthogonal transformations. As a consequence, it is impossible to distinguish the original sources from any orthogonal mixture constructed from them using ICA techniques.

To demonstrate, consider the Cocktail Party scenario with two speakers (sources) and two recorders (mixtures). Assume that an orthogonal[5] mixing matrix and the two independent sources are Gaussian. Then Y_1 and Y_2 are also Gaussian, uncorrelated, and have unit variance. The joint

[3] As opposed to estimated *principal components* in PCA (cf. Section 1.4).

[4] It's worth emphasizing that if only one of the independent components is Gaussian, the ICA model can still be estimated. Refer to Section 3.3.3 to see how having more than one Gaussian density as the source signal affects the estimation in our proposed approach.

[5] The orthogonality consideration is elaborated on later.

density function is given by

$$f(y_1, y_2) = \frac{1}{2\pi} \exp\left(-\frac{y_1^2 + y_2^2}{2} \right), y_1, y_2 \in \mathbb{R}.$$

Their joint distribution is illustrated below in Figure 1.1.

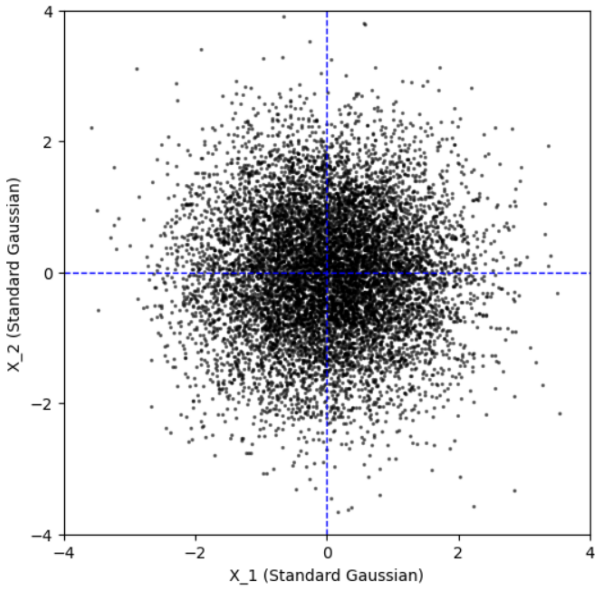

Figure 1.1: Simulation of the joint density of two independent standard Gaussian random variables. Simulated on `python`, sample size 10000.

The joint density is symmetric, as evident from the figure, and consequently lacks any directional information regarding the columns of the mixing matrix $\mathbf{A}$. Thus, it is a standard assumption in ICA that at most one source is Gaussian; otherwise, the model can only estimate the components up to an orthogonal transformation.

To summarize, assuming at most one independent component is Gaussian, scale transformations, permutations, and changes in sign of the independent components are the primary and only sources of unidentifiability for the estimation of the unmixing matrix $\mathbf{B}$ (or the mixing matrix $\mathbf{A}$).

1.4 Distinguishing ICA and PCA

Below we present a detailed comparison of two popular techniques in data analysis and signal processing: *Independent Component Analysis* (ICA) and *Principal Component Analysis* (PCA).

These methods are often confused due to similarities in their names, yet they differ fundamentally in their objectives, assumptions, and areas of application. PCA and ICA address distinct problems and rely on different statistical principles, which makes each suitable for particular contexts. In this section, we highlight why ICA is often more appropriate than PCA in scenarios involving source separation or data with strong non-Gaussian characteristics.

1.4.1 Primary Objective

Principal Component Analysis (PCA) is designed to reduce the dimensionality of data while preserving as much total variance as possible. It transforms the original data into a new set of uncorrelated (though not necessarily independent) variables called *principal components*. These components are ordered such that the first principal component captures the maximum possible variance, the second captures the next largest variance subject to orthogonality with the first, and so on.

Independent Component Analysis (ICA), on the other hand, is used to separate observed multivariate data into *independent components*. It is useful when the underlying data is (or at least can be assumed to be) composed of independent sources. The goal of ICA is to *unmix* the data into statistically independent components. ICA does not prioritize variance in any way, nor does it impose any ordering on the recovered components (cf. Section 1.3.2).

1.4.2 Assumptions

PCA assumes that the structure in the data is represented by variance and is thus best explained by projections along directions of maximal variance. It presumes that the underlying structure is captured through second-order statistics (i.e., covariances) and assumes orthogonality of components. PCA only removes correlations and does not account for other dependencies arising in data, such as statistical independence.

In contrast, ICA attempts to unmix observed data into statistically independent sources, owing to the main assumption that the observations are generated by a linear combination of independent, non-Gaussian sources. ICA explicitly minimizes the statistical dependencies between components,

going beyond the scope decorrelation (as in PCA). Unlike PCA, ICA does not require orthogonality of components, allowing it to capture independent structures that may be related in a non-orthogonal[6] manner. This makes ICA more suited for data where the independence assumption is vital, such as blind source separation (BSS) problems.

1.4.3 Output Components

The components obtained from PCA are orthogonal and ordered by the amount of variance they explain. This makes PCA particularly useful for dimensionality reduction. However, as each principal component is a linear combination of features from the original variables, they lack straightforward interpretability.

ICA produces components that are statistically independent in the sense that knowledge of one provides no information about the others. These components are neither constrained to be orthogonal nor ordered (by variance). ICA operates on the assumption of independence of the components, which is a much stronger criterion than uncorrelatedness, making it better for scenarios where the goal is to extract distinct underlying sources.

1.4.4 Scope/Applicability

PCA transforms the data into directions of maximum variance, but the principal components are still mixtures of the underlying sources. It is primarily used when the goal is to capture the variance in the data than to recover (independent) sources. As such it finds more use as a tool for dimensionality reduction, noise reduction, and data compression.

ICA is more specialized and computationally intensive, but excels in when the goal is to separate mixed signals into their constituent independent sources. When the independence assumption is justified, ICA can recover the original signals more effectively than PCA, as it is designed to "unmix" the observations. This makes ICA a preferred choice for a variety of blind source separation problems such as in signal processing and imaging (e.g., audio signals, EEG/MEG recordings, etc.).

[6] Standard PCA is not suited for capturing non-linear relationships. It is usually advised to turn non-linear relations into linear ones, by the use of data transformations (e.g., log transforms), or use non-linear extensions like Kernel-PCA (cf. Schölkopf, Smola, and Müller (1998)).

1.4.5 Why use ICA instead of PCA?

Although PCA is effective for dimensionality reduction and identifying orthogonal directions of maximal variance, it fails to recover the underlying independent sources when such sources exist. Its assumption does not always hold, especially when dealing with non-Gaussian data. Real-world data often contain non-Gaussian distributions (e.g., audio, images, biomedical data), where variance alone is not the best criterion for separating meaningful signals. ICA is more appropriate when the goal is to recover latent, independent sources, particularly in contexts involving signal mixtures or generative modeling. ICA is hence preferred over PCA in tasks like blind source separation or uncovering latent factors in non-Gaussian settings, where statistical independence — rather than mere decorrelation — is crucial.

Table 1.1: Comparison between PCA and ICA

Aspect	**PCA**	**ICA**
Primary Objective	Find orthogonal directions that maximize variance	Find components that maximize statistical independence
Assumptions	Components are uncorrelated; relies on second-order statistics (covariance)	Components are statistically independent; assumes non-Gaussianity of sources
Output Components	Orthogonal and uncorrelated; ordered by explained variance	Statistically independent; not necessarily orthogonal or ordered
Identifiability	Unique up to sign and scaling	Unique up to permutation and scaling
Computation	Closed-form solution via eigenvalue decomposition (EVD); Computationally fast and not intensive	Iterative, nonlinear optimization algorithms (e.g., FastICA); more computationally intensive
Use Case	Dimensionality reduction, decorrelation, noise reduction	Separation of independent underlying sources from observations

In the next section, we discuss the established standard approach to Independent Component Analysis, which always starts with preprocessing the data of mixtures, by centering and whitening. In the following section, we provide an overview of some existing methods for ICA estimation.

1.5 Preprocessing the data

Preprocessing plays a critical role in data analysis tasks by improving data quality, reducing noise, and enhancing the performance of models. Through techniques such as data cleaning, normalization, feature scaling, and dimensionality reduction, data preprocessing ensures that the input data is well-structured, consistent, and suitable for the chosen algorithms. Assuming complete data, we will discuss the important preprocessing aspects of the ICA model in particular.

1.5.1 Centering the mixtures

Without loss of generality, the independent components $\mathbf{X}$ are assumed to have zero means, i.e., $\mathbb{E}(\mathbf{X}) = \mathbf{0}$. This can be enforced by centering the data. If the data isn't zero-mean, one can always center them by subtracting the mean from each observed variable. Suppose the data $\mathbf{Y}^*$ has a non-zero mean, $\mathbb{E}(\mathbf{y}^*) \neq \mathbf{0}$. The centering transformation $\mathbf{Y} = \mathbf{Y}^* - \mathbb{E}(\mathbf{Y}^*)$, can be applied to yield $\mathbb{E}(\mathbf{Y}) = \mathbf{0}$. As a result, $\mathbb{E}(\mathbf{X}) = \mathbf{B}\mathbb{E}(\mathbf{Y}) = \mathbf{0}$. Of course, this will not have any effect on the mixing matrix $\mathbf{A}$ or its inverse $\mathbf{B}$.

1.5.2 Whitening the mixture data

A zero-mean random vector (of mixtures) $\mathbf{y}$ is said to be *whitened* if its components are uncorrelated and their variances equal unity, i.e. the dispersion matrix of the mixtures is $\mathbb{E}(\mathbf{Y}\mathbf{Y}^\top) = \mathbf{I}$. For any data, a whitening transformation[7] is always possible. A standard and simple approach is using the *eigenvalue decomposition*[8] (EVD) of the dispersion matrix.

Whitening via eigenvalue decomposition is a straightforward procedure. Rewrite the dispersion matrix of the mixtures $\mathbf{Y}$ (already assumed to have zero means) as follows:

$$\mathbb{E}(\mathbf{Y}\mathbf{Y}^\top) = \mathbf{P}\mathbf{D}\mathbf{P}^\top,$$

where $\mathbf{P}$ is the orthonormal matrix of eigenvectors of $\mathbb{E}(\mathbf{Y}\mathbf{Y}^\top)$, and $\mathbf{D} = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_d)$ is a diagonal matrix of its corresponding eigenvalues. Whitening can be performed by pre-multiplying

[7] The transformation is called "whitening" because it changes the input vector into a white noise vector.

[8] cf. Appendix A.1

the mixtures $\mathbf{Y}$ by $\mathbf{S} = \mathbf{P}\mathbf{D}^{-\frac{1}{2}}\mathbf{P}^\top$.

1.5.3 Orthogonality

Once the data of mixtures $\mathbf{Y}$ have been *whitened* with $\mathbf{S}$ (as above), the new mixture data would be

$$\mathbf{Z} = \mathbf{S}\mathbf{Y} = \mathbf{S}\mathbf{A}\mathbf{X} = \mathring{\mathbf{A}}\mathbf{X},$$

where $\mathring{\mathbf{A}}$ is the new mixing matrix, and with $\mathbf{Z}$ having mean $\mathbb{E}(\mathbf{Z}) = \mathbf{0}$ and dispersion matrix $\mathbb{E}(\mathbf{Z}\mathbf{Z}^\top) = \mathbf{I}$. It is important to note that, in such cases, $\mathring{\mathbf{A}}$ is always orthogonal as shown below:

Lemma 1.5.1. *For the preprocessed mixtures* $\mathbf{Z}$ *and the corresponding ICs* $\mathbf{X}$*, the mixing matrix* $\mathring{\mathbf{A}}$ *can always be assumed to be orthogonal.*

Proof. Under the usual assumptions (Section 1.2), $\mathbb{E}(\mathbf{X}\mathbf{X}^\top) = \mathbf{I}$. So,

$$\mathbf{I} = \mathbb{E}(\mathbf{Z}\mathbf{Z}^\top) = \mathring{\mathbf{A}}\mathbb{E}(\mathbf{X}\mathbf{X}^\top)\mathring{\mathbf{A}}^\top = \mathring{\mathbf{A}}\mathring{\mathbf{A}}^\top.$$

■

It is usually advised to pre-process the data of mixtures $\mathbf{Y}$ to have zero means and the identity matrix as its dispersion matrix. Whitening, in particular, provides the added benefit of transforming the new mixing matrix orthogonal, allowing the search for the mixing matrix $\mathbf{A}$ (or its inverse $\mathbf{B}$, the so-called unmixing matrix) to be restricted to only the space of orthogonal matrices for the (new) whitened data. Most, if not all, ICA algorithms and estimation techniques work assuming the data is already centered and whitened.

1.6 Established Procedures and Algorithms

Here we discuss some of the preexisting methods of Independent Component Analysis, from the most popular ones to some less-used, yet novel, approaches.

1.6.1 FastICA

FastICA is perhaps the most popular and simple ICA algorithm in use. It is a method built on projection pursuit, seeking to find an orthogonal rotation of preprocessed data, via a fixed-point iterative method by optimizing a measure of non-Gaussianity (usually kurtosis or negentropy) or mutual information (entropy).

Under the ICA model, we have $\mathbf{Y} = \mathbf{AX} \iff \mathbf{X} = \mathbf{BY}$. For pre-processed data $\mathbf{Y}$, we search for a linear combination $\mathbf{b}^\top \mathbf{Y}$ that maximizes some predefined cost function (in general, optimizes based on the measure used). For the purpose of illustration, we summarize the procedure addressed in Hyvärinen (1999) and Hyvärinen et al. (2001), where the authors use a combination of *information-theoretic* and *projection-pursuit* approaches to decompose mixed signals into statistically independent components.

The observed data $\mathbf{Y}$ is a linear mixture of independent source signals $\mathbf{X}$ via a mixing matrix $\mathbf{A}$, i.e.,

$$\mathbf{Y} = \mathbf{AX}.$$

The goal is to find a matrix B such that

$$\mathbf{X} = \mathbf{BY},$$

where the components of $\mathbf{X}$ are statistically independent. We start with the linear combination $\mathbf{b}^\top \mathbf{Y}$ which is the *projection* of the data vector on the line spanned by the vector $\mathbf{b}$, constrained to be on the unit sphere. It captures the component of the data $\mathbf{Y}$ in the direction specified by the vector $\mathbf{b}$.

The iterative procedure aims at finding the direction of the vector $\mathbf{b}$ that maximizes the measure of non-Gaussianity of the projection $\mathbf{b}^\top \mathbf{Y}$. FastICA algorithms using measures of non-Gaussianity require a function $G(z)$, and its first and second derivatives $G^{(1)}(z)$ and $G^{(2)}(z)$. Recommended choices for this function $G(\cdot)$, as suggested by Hyvärinen (1999), are:

$$\begin{aligned}
G_1(z) &= \frac{1}{a_1} \log\cosh\left(a_1 z\right),\ a_1 \in [1,2],\ \text{usually } a_1 = 1; \\
G_2(z) &= -\frac{1}{a_2} \exp\left(\frac{-a_2 z^2}{2}\right),\ a_2 \approx 1; \\
G_3(z) &= \frac{z^4}{4}.
\end{aligned}$$

Each function has its own merits and demerits. G_1 is general-purpose; G_2 is more robust and works better with super-Gaussian[9] components, while G_3 (and kurtosis) is better suited for sub-Gaussian[10] components.

FastICA algorithms can be used to extract one or multiple components. The steps of the FastICA algorithm to estimating one independent component, as outlined in Hyvärinen et al. (2001), are provided below in Algorithm 1. Of course, the expectations in Step 4 cannot be computed analytically and are instead replaced by their empirical (sample) estimates from your observed data.

Algorithm 1 FastICA: Estimating One Independent Component

Require: Observed data matrix $\mathbf{Y}$, nonlinearity function G, tolerance ε

1: Center and whiten the data.

2: Randomly initialize a unit-norm vector $\mathbf{b}$.

3: **repeat**

4: Compute: $\mathbf{b} \leftarrow \mathbb{E}\left[\mathbf{y} \cdot G^{(1)}(\mathbf{b}^\top \mathbf{y})\right] - \mathbb{E}\left[G^{(2)}(\mathbf{b}^\top \mathbf{y})\right]\mathbf{b}$

5: Normalize: $\mathbf{b} \leftarrow \dfrac{\mathbf{b}}{\|\mathbf{b}\|}$

6: **until** convergence (i.e., change in $\mathbf{b}$ below threshold ε)

7: **Return** $\mathbf{b}$ as the estimated independent component direction.

Estimating other (mutually) independent components just requires the algorithm to be repeated to obtain other independent projections. The steps for estimating several independent components are outlined in Algorithm 2. The method of orthogonalization used in step (6) in Algorithm 2 is called symmetric orthogonalization (formulated by Löwdin (1970)). There are several other ways to carry out orthogonalization of the matrix, namely, Gram–Schmidt process, Householder transformation, etc.

[9] These have heavier tails than a Gaussian; extreme deviations are more likely, e.g., Laplace distribution.

[10] These have lighter tails compared to a Gaussian distribution, like uniform distribution.

Algorithm 2 FastICA: Estimating Multiple Independent Components

Require: Observed data matrix $\mathbf{Y}$, number of components m, nonlinearity function G, tolerance ε

1: Center and whiten the data.

2: Choose m: the number of independent components to estimate.

3: Initialize m unit-norm random vectors $\{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$ and form matrix $\mathbf{B} = [\mathbf{b}_1 \ \ldots \ \mathbf{b}_m]$.

4: Orthogonalize: $\mathbf{B} \leftarrow (\mathbf{B}\mathbf{B}^\top)^{-\frac{1}{2}}\mathbf{B}$

5: **repeat**

6: **for** $i = 1$ to m **do**

7: Update each $\mathbf{b}_i$: $\mathbf{b}_i \leftarrow \mathbb{E}\left[\mathbf{y} \cdot G^{(1)}(\mathbf{b}_i^\top \mathbf{y})\right] - \mathbb{E}\left[G^{(2)}(\mathbf{b}_i^\top \mathbf{y})\right]\mathbf{b}_i$

8: **end for**

9: Orthogonalize: $\mathbf{B} \leftarrow (\mathbf{B}\mathbf{B}^\top)^{-1/2}\mathbf{B}$

10: **until** convergence of all $\mathbf{b}_i$ (change below threshold ε)

11: **Return** estimated unmixing matrix $\mathbf{B}$

For other fixed-point algorithm variants, and differences in the approaches depending on the measure used (non-Gaussianity or mutual information) see Chapters 8 and 10 in Hyvärinen et al. (2001).

1.6.2 ICA by Maximum Likelihood Estimation

The density of the mixtures $\mathbf{Y}$ can be expressed as a function of the unmixing matrix $\mathbf{B} = [\mathbf{b}_1 \, \mathbf{b}_2 \ldots \mathbf{b}_d]^\top$:

$$f(\mathbf{y}) = |\det(\mathbf{B})| \prod_{j=1}^{d} f_j(\mathbf{b}_j^\top \mathbf{y}).$$

Considering n observations $\mathbf{y}_i; i = 1, 2, \ldots, n$, the log-likelihood function can be set up as follows:

$$\begin{aligned} l^n &= l^n(B, f_1, \ldots, f_d) = l^n(B, f_1, \ldots, f_d; \mathbf{y}_1, \ldots, \mathbf{y}_n) \\ &= \log|\det(\mathbf{B})| + \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{d} \log f_j(\mathbf{b}_j^\top \mathbf{y}_i) \end{aligned} \tag{7}$$

Parametric Approach

The parametric maximum likelihood approach assumes a parametric family for the distributions $f_1, \ldots, f_d$ and then optimizes the contrast function (7). This is tackled using fixed-point algorithms. The problem then becomes almost identical to what has been discussed in the previous subsection. A FastICA algorithm can be directly applied to the problem of maximization of the likelihood function. In such a case, a part of the log-likelihood expression, viz. $\log|\det(\mathbf{B})|$, is constant and the log-likelihood function is just the sum of n terms, each of the form optimized by FastICA algorithms.

The main drawback to this approach, and other methods that rely on the optimization of a contrast function, represented by measures of non-gaussianity (kurtosis, negentropy) or mutual information (entropy) is the presumption of a parametric family. Indeed, even most non-parametric approaches assume that the family of densities is *smooth* as well as rely on *tuning parameters* which are notoriously difficult to choose. For example, see A. Chen and Bickel (2006), Ilmonen and Paindaveine (2011).

Non-parametric Approach

Samworth and Yuan (2012) proposed a non-pamarmetric MLE method free of smoothness assumption and nuisance tuning parameters. The only assumption being that the source densities $f_1, \ldots, f_d$ are log-concave. Using what they term the *log-concave ICA projection* of the empirical distribution function, the estimating procedure is constructed. The algorithm they proposed is concisely presented in Algorithm 3.

Algorithm 3 Nonparametric MLE Approach to ICA

The unmixing matrix and the marginal densities are estimated by maximizing (7), where the matrix $\mathbf{B} \in \mathcal{O}(d)$: set of all orthogonal $d \times d$ matrices with determinant 1.

Require: Centered and whitened data $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$, tolerance η

1: **Initialize** $\mathbf{B}$: Generate a random $d \times d$ matrix $\mathbf{Z}$ with IID standard Gaussian entries, compute QR-factorization $\mathbf{Z} = \mathbf{QR}$, and set $\mathbf{B} \leftarrow \mathbf{Q}$ (authors' suggestion).

2: **repeat**

3: **for** $j = 1$ to d **do**

4: Project data: $z_{ij} \leftarrow \mathbf{b}_j^\top \mathbf{y}_i$ for $i = 1, \ldots, n$

5: Estimate f_j as the univariate log-concave MLE of $\{z_{1j}, \ldots, z_{nj}\}$ (e.g., using the `logcondens` package in R)

6: **end for**

7: Update $\mathbf{B} \leftarrow \mathbf{B} \exp(\varepsilon \mathbf{S}^*)$. Here, $\mathbf{S}^*$ is a special skew-symmetric matrix and ε is a step size constant, the details for which can be found in Samworth and Yuan (2012).

8: Compute log-likelihood $\ell^{(k)}$ using:

$$\ell^{(k)} = \log |\det \mathbf{B}| + \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{d} \log f_j(\mathbf{b}_j^\top \mathbf{y}_i)$$

9: **until** $\dfrac{l^{n(k)} - l^{n(k-1)}}{|l^{n(k-1)}|} < \eta$

10: **Return** final unmixing matrix $\mathbf{B}$ and densities $f_1, \ldots, f_d$

See Appendix A.2 for a brief explanation of QR-factorization mentioned in step (2) of the Algorithm 3.

1.6.3 Semiparametric One-Step Estimation based on Ranks

The work of Ilmonen and Paindaveine (2011) interestingly suggests a signed-rank-based approach to finding an estimator for the mixing matrix $\mathbf{A}$. Inference is drawn on the mixing matrix by considering a *normalized* representative $\mathbf{L}$ within an equivalence class. They point out that an estimate of this matrix $\mathbf{L}$ allows for the recovery of the Independent Components $X_1, \ldots, X_d$ as well

as any other estimate $\mathbf{A}$ as long as $\mathbf{L} \sim \mathbf{A}$.[11]. While their paper also outlines the testing procedure for the null hypothesis $\mathbf{H}_0 : \mathbf{L} = \mathbf{L}_0$ against $\mathbf{H}_1 : \mathbf{L} \neq \mathbf{L}_0$, for some fixed $\mathbf{L}_0$, we prioritize looking at the point estimate of the normalized matrix $\mathbf{L}$. Unlike most other methods of ICA estimation (including the ones discussed in this section) that require a converging algorithm, they propose a "one-step rank estimator".

Under some conditions and assumptions (symmetric components, ULAN[12] family of target densities, etc), the asymptotic properties and the explicit form of this *One-step R-estimator* are derived and stated in the form of two theorems (Theorem 4.1 and Theorem 4.2 respectively) in Ilmonen and Paindaveine (2011).

Below is a simplified statement for the construction of the estimator starting with the primary assumption:

Assumption: For all $r \neq s \in \{1, \ldots, p\}$, dispose of sequences of estimators $\hat{\gamma}_{rs}(f)$ and $\hat{\varrho}_{rs}(f)$, of the so-called cross-information coefficients[13] γ_{rs} and ϱ_{rs}, that:

i. are locally asymptotically discrete; and,

ii. for any $g \in \mathfrak{F}_{\text{ULAN}}$, satisfy $\hat{\gamma}_{rs}(f) = \gamma_{rs}(f, g) + o_p(1)$ and $\hat{\varrho}_{rs}(f) = \varrho_{rs}(f, g) + o_p(1)$ as $n \longrightarrow \infty$, under the proposed semi-parametric model.

To deal with the estimation the cross-information coefficients $\gamma_{rs}(f, g)$ and $\varrho_{rs}(f, g)$, they suggest a solution based entirely on ranks, treating g as a nuisance parameter. This is explored in detail in Section 4.2 in their paper. Define the statistics $\hat{\alpha}_{rs}(f)$ and $\hat{\beta}_{rs}(f)$, for $r \neq s \in \{1, \ldots, p\}$, obtained by plugging in the estimators $\hat{\gamma}_{rs}$ and $\hat{\varrho}_{rs}$ (from the *assumption*) into

$$\alpha_{rs}(f, g) = \frac{\gamma_{rs}(f, g)}{\gamma_{rs}(f, g)\gamma_{sr}(f, g) - \varrho_{rs}(f, g)\varrho_{sr}(f, g)},$$
$$\beta_{rs}(f, g) = \frac{-\varrho_{rs}(f, g)}{\gamma_{rs}(f, g)\gamma_{sr}(f, g) - \varrho_{rs}(f, g)\varrho_{sr}(f, g)};$$

with $\hat{\alpha}_{rr}(f) := 0 =: \hat{\beta}_{rr}(f),\ r = 1, 2, ..., p$.

[11] The symbol $\sim$ stands for *similar matrices*.

[12] ULAN refers to Uniform Local and Asymptotically Normal.

[13] The exact forms of the the cross-information coefficients are defined in Section 3 (Hypothesis Testing) of Ilmonen and Paindaveine (2011).

In the light of the *assumption* and statements made, and for a symmetric fixed target density (of the components) $f \in \mathfrak{F}_{\text{ULAN}}$, define

$$\hat{N}_f := (\hat{A}_f^\top \odot T_{v,f}^\top) + (\hat{B}_f^\top \odot T_{v,f}^\top), \text{ where } \hat{A}_f = (\hat{\alpha}_{rs}(f)) \text{ and } \hat{B}_f = (\hat{\beta}_{rs}(f)),$$

where v is a $\sqrt{n}$-consistent and locally asymptotic discrete estimator assumed to be available[14], T is a special off-diagonal matrix based on the signed ranks. The symbol $\odot$ stands for the *Hadamard product*, the entry-wise multiplication of equal-sized matrices.

Then the estimator is given by

$$\hat{\mathbf{L}}_f = \mathbf{L} + \frac{1}{\sqrt{n}}\mathbf{L}\left[\hat{N}_f - \text{diag}(\mathbf{L}\hat{N}_f)\right]$$

where $\mathbf{L}$ is a preliminary estimator, and the notation $\text{diag}(\mathbf{M})$ means a diagonal matrix with the same diagonal entries as the matrix $\mathbf{M}$. The initial estimator $\mathbf{L}$ can be obtained using other methods like FastICA.

The estimator $\hat{\mathbf{L}}_f$ is shown to enjoy certain desirable asymptotic properties (Theorem 4.1 in Ilmonen and Paindaveine (2011)), for instance,

i. it maintains $\sqrt{n}$-consistency and asymptotic normality for a wide range of densities;

ii. it is semi-parametrically efficient under correctly specified densities;

iii. its asymptotic covariance matrix can easily be estimated consistently.

However, this approach is not $\sqrt{n}$-consistent when even one of the component densities violates the symmetry assumption. A more recent work, by Hallin and Mehta (2015), addresses this drawback and extends into asymmetric densities, by considering ranks instead of signed ranks.

1.6.4 Efficient Independent Component Analysis

A. Chen and Bickel (2006) suggest an "efficient" semi-parametric approach to the ICA estimation problem. The ICA model assumes that the observed data $\mathbf{y}$ is a linear mixture of independent

[14] They show that the asymptotic properties of the estimator are not affected by its choice.

components $\mathbf{x}$, i.e.,

$$\mathbf{y} = \mathbf{A}\mathbf{x} \Longleftrightarrow \mathbf{x} = \mathbf{B}\mathbf{y}. \tag{8}$$

The likelihood function of $\mathbf{x}$ under (8) is given by

$$p_{\mathbf{x}}(\mathbf{y}, \mathbf{B}, f_i) = |\det(\mathbf{B})| \prod_{i=1}^{m} f_i(\mathbf{b}_i^\top \mathbf{y}),$$

where $\mathbf{B}$, the unmixing matrix, is the parameter of interest, and $f_i, i = 1, 2, ..., d$; are the marginal density functions of the independent components, which are unknown and treated as nuisance parameters.

To derive an estimate for $\mathbf{B}$, they define an *efficient score function*:

$$\mathbb{I}^*(\mathbf{y}, \mathbf{B}, \Phi_{\mathbf{B}}) = \text{vec}(\mathbf{M}(\mathbf{B}\mathbf{y})(\mathbf{B}^{-1})^\top), \tag{9}$$

where $\mathbf{M}(\mathbf{x})$ is an $m \times m$ matrix with entries

$$m_{ij}(\mathbf{x}) = \begin{cases} -\phi(x_i)x_j, & \text{if } i \neq j \\ \alpha x_i + \beta\eta(x_i), & \text{if } i = j. \end{cases}$$

The terms α and β are constants that depend on moments of f_i, and $\eta(x_i)$ is a function that enforces scaling constraints. See Amari and Cardoso (1997) for the derivation of these terms. The efficient score function (9) depends on f_i only through the density scores ϕ_i, where

$$\phi_i(x_i) = -\frac{\mathrm{d}}{\mathrm{d}x_i} \log f_i(x_i). \tag{10}$$

$\Phi_{\mathbf{B}}$ is a vector containing the density score functions. To obtain an efficient estimator of $\mathbf{B}$, first construct an estimate $\hat{\Phi}_{\mathbf{B}}$ of $\Phi_{\mathbf{B}}$ and then solve the efficient score equation

$$\int \mathbb{I}^*(\mathbf{x}, \mathbf{B}, \hat{\Phi}_{\mathbf{B}}) \mathrm{d}P_n = 0. \tag{11}$$

Here, the estimate $\hat{\Phi}_{\mathbf{B}}$ is a data-driven function of $\mathbf{B}$, and so $\mathbb{I}^*(\mathbf{y}, \mathbf{B}, \hat{\Phi}_{\mathbf{B}})$ is an approximation to

the efficient score function.

Starting with an initial point estimate of the matrix $\mathbf{B}$ the entire estimation procedure can be condensed into a pseudo-algorithm (Algorithm[15] 4) to obtain $\hat{\mathbf{B}}$. Once such initialization is provided from their previous work A. Chen and Bickel (2005), referred to as PCFICA.

The authors show that the limit exists, and where the sequence $\hat{\mathbf{B}}^{(j)}, j = 0, 1, ...$ converges, the limit $\hat{\mathbf{B}}^{(\infty)} =: \hat{\mathbf{B}}$ is taken as the final estimate of the unmixing matrix.

1.6.5 Non-parametric ICA

Samarov and Tsybakov (2004) present a non-parametric framework for estimation in an ICA setup. It is based on the diagonalization of matrix functionals derived from the data, and achieves $\sqrt{n}$-consistency for the principal directions of the IC estimates. The procedure is focused on estimating the mixing matrix $\mathbf{A}$ which relates the observed data $\mathbf{y}$ to the independent components $\mathbf{X}$ as in (3).

Suppose $\mathbf{Y}_1, ..., \mathbf{Y}_n$ are IID $\mathbb{R}^d$-valued $(d \geq 2)$ random variables with common density function p, such that

$$p(\mathbf{y}) = |\det(\mathbf{A})| \prod_{j=1}^{d} p_j(\mathbf{y}^\top \mathbf{a}_j), \ \mathbf{y} \in \mathbb{R}^d, \tag{12}$$

where $\mathbf{A} = (\mathbf{a}_1, ..., \mathbf{a}_d)$ is the unknown (but assumed to be orthogonal) mixing matrix with linearly independent unit length columns $\mathbf{a}_j \in \mathbb{R}^d$, and $p_j(\cdot), \ j = 1, 2, ..., d$ are unknown probability densities, defined on $\mathbb{R}$, of the independent components. p_j are assumed to be smooth densities. The authors develop a two-step approach to estimate the mixing matrix $\mathbf{A}$, using the properties of the dispersion matrix of the observed data and the gradient of the joint density function (12).

The first step involves the relation between the dispersion matrices of the observations and the independent components. Note that the dispersion matrix of $\mathbf{Y}$, $\mathbf{\Sigma}_Y = \mathbb{E}[\mathbf{Y}\mathbf{Y}^\top]$ is related to that of $\mathbf{X}$, denoted by $\mathbf{\Sigma}_X$, as follows:

$$\mathbf{\Sigma}_X = \mathbb{E}[\mathbf{X}\mathbf{X}^\top] = \mathbb{E}[\mathbf{A}^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{A}] = \mathbf{A}^\top \mathbb{E}[\mathbf{Y}\mathbf{Y}^\top]\mathbf{A} = \mathbf{A}^\top \mathbf{\Sigma}_Y \mathbf{A}. \tag{13}$$

This implies that $\mathbf{A}$ diagonalizes the matrix $\mathbf{\Sigma}_Y$ as $\mathbf{\Sigma}_X$ is a diagonal matrix containing the variances

[15] The procedure is named *Efficient ICA* due to the fact that it utilizes the efficient score function as defined in (10).

Algorithm 4 Efficeint ICA

1: **Input:** Observed data: $\mathbf{y}^1, ..., \mathbf{y}^n$, an initial estimate $\hat{\mathbf{B}}^{(0)}$, B-spline basis functions $\mathbf{S}^{(k)} \equiv (\mathbf{S}_1^{(k)}, ..., \mathbf{S}_{n_k}^{(k)})$.

2: **Initialize Parameters:**

   i. Initialize $\hat{\mathbf{B}}^{(0)}$. Can be obtained by using any consistent method, e.g., FastICA, PCFICA.

   ii. Define a *sieve* estimator $\hat{\Phi}_{\mathbf{B}}$ for the density scores $\Phi_{\mathbf{B}}$.

3: **B-Spline Approximation:** For each $k = 1, ..., m$, select a sieve for $\hat{\phi}_{\mathbf{B}_k}$, the density score of the k^{th} component

   i. Select an interval $[s^l_{(n_k)}, s^u_{(n_k)}] \in \mathbb{R}$ containing most of the mass of f_k.

   ii. Choose $n_k + 4$ equi-spaced knots over this interval.

   iii. Construct n_k cubic B-spline basis functions over the knot sequence $\mathbf{S}_n^{(k)} \equiv (\mathbf{S}_{n,1}^{(k)}, ..., \mathbf{S}_{n,n_k}^{(k)})$.

4: **Estimate Density Score Functions:** Using the random sample $\mathbf{B}_k\mathbf{y}^i, i = 1, 2, ..., n$ from the density function $f_{\mathbf{B}_k}$ estimate (see Jin (1992)) the score function

$$\hat{\phi}_{\mathbf{B}_k} = [\gamma_n(\mathbf{B}_k)]^\top \mathbf{S}_n^{(k)},$$

   where $\gamma_n(\mathbf{B}_k)$ is calculated empirically from sample moments.

5: **Estimated score function:**

   i. Define $\hat{\Phi}_{\mathbf{B}} = \left(\hat{\phi}_{\mathbf{B}_1}(x_1), ..., \hat{\phi}_{\mathbf{B}_1}(x_m)\right)^\top$.

   ii. Replace the efficient score function (9) with $\mathbb{I}^*(\mathbf{y}, \mathbf{B}, \hat{\Phi}_{\mathbf{B}})$.

6: **Plug-in estimates:** Replace parameters α_i, β_i and σ_i^2 in the score function by their plug-in estimates.

7: **Efficient score function:**

   i. The empirical efficient score function equation is given by

$$e_n(\mathbf{B}) = \int \mathbb{I}^*(\mathbf{y}, \mathbf{B}, \hat{\Phi}_{\mathbf{B}}) \mathrm{d}P_n.$$

   ii. Solve the equation $e_n(\mathbf{B}) = 0$ for an efficient estimate $\hat{\mathbf{B}}$.

8: **Newton-Raphson iteration:**

   i. Update $\hat{\mathbf{B}}$ iteratively using the rule

$$\hat{\mathbf{B}}^{(j+1)} = \hat{\mathbf{B}}^{(j)} + \left[\int \mathbb{I}^* \mathbb{I}^{*t}(\mathbf{y}, \hat{\mathbf{B}}^{(j)})\right]^{-1} e_n(\hat{\mathbf{B}}^{(j)}), j = 0, 1, \dots .$$

   ii. Iterate until convergence, i.e., $\hat{\mathbf{B}}^{(j+1)}$ nad $\hat{\mathbf{B}}^{(j)}$ are close enough.

9: **Output**: Final estimate $\hat{\mathbf{B}}$.

of the independent components.

The second step involves estimating a matrix functional derived from the gradient of the joint density of $\mathbf{Y}$ (12). They define the functional $T(p)$ based on the outer product of the gradient, viz.,

$$T(p) \coloneqq \mathbb{E}[\nabla p(\mathbf{X}) \, (\nabla p(\mathbf{X}))^\top]. \tag{14}$$

For densities obeying (12), and under a mild assumption, it can be shown that $T(p)$ takes the form

$$T(p) = \sum_{j=1}^{d} c_{jj} \mathbf{a}_j \mathbf{a}_j^\top$$

or, equivalently in matrix notation,

$$\mathbf{T} =: T(p) = \mathbf{A}\mathbf{C}\mathbf{A}^\top, \tag{15}$$

where $\mathbf{C}$ is a diagonal matrix[16] with entries $c_{11}, ..., c_{dd}$ (> 0). Thus, $\mathbf{T}$ is positive definite and so,

$$\mathbf{C}^{-1} = \mathbf{A}^\top \mathbf{T}^{-1} \mathbf{A}. \tag{16}$$

If we define the matrices $\mathbf{P} = \mathbf{A}\mathbf{C}^{\frac{1}{2}}$ and $\mathbf{\Lambda} = \mathbf{C}^{\frac{1}{2}} \mathbf{\Sigma}_X \mathbf{C}^{\frac{1}{2}}$, then (13) and (16) imply that

$$\mathbf{P}^\top \mathbf{\Sigma}_Y \mathbf{P} = \mathbf{\Lambda}, \text{ and } \mathbf{P}^\top \mathbf{T}^{-1} \mathbf{P} = \mathbf{I}_d. \tag{17}$$

A result in matrix algebra (see Section 6.7 in Lewis (1991)) states that for any two positive semi-definite symmetric matrices $\mathbf{\Sigma}_Y$ and $\mathbf{T}^{-1}$, if at least one of them is positive definite, then there exists a non-singular matrix $\mathbf{P}$ and a diagonal matrix $\mathbf{\Lambda}$ such that (17) holds.

Here, the diagonal entries of $\mathbf{\Lambda}$, denoted by λ_j, are the eigenvalues of $\mathbf{T}\mathbf{\Sigma}_Y$, and the columns of $\mathbf{P}$ are the corresponding eigenvectors. Therefore, if λ_j are all different, columns of $\mathbf{P}$ can be

[16] $T(p)$ is more accurately represented as $T(p) = \sum_{i=1}^{d} \sum_{j=1}^{d} c_{ij} \mathbf{a}_i \mathbf{a}_j^\top$. However, the assumption mentioned (but not elaborated here) ensures that $c_{jk} = 0, \, \forall \, j \neq k$, resulting in what is obtained above.

uniquely identified as vectors $\mathbf{p}_j$ from

$$\mathbf{T}\boldsymbol{\Sigma}Y\mathbf{p}_j = \lambda_j\mathbf{p}_j, j = 1, 2, ..., d.$$

For $\mathbf{q}_j \coloneqq \mathbf{T}^{-\frac{1}{2}}\mathbf{p}_j$, the above equation can be rewritten as

$$\mathbf{T}^{\frac{1}{2}}\boldsymbol{\Sigma}_Y\mathbf{T}^{\frac{1}{2}}\mathbf{q}_j = \lambda_j\mathbf{q}_j, j = 1, 2, ..., d,$$

where $\mathbf{T}^{\frac{1}{2}}$ is the symmetric square root of $\mathbf{T}$. This in turn implies that $\mathbf{T}^{\frac{1}{2}}\boldsymbol{\Sigma}_Y\mathbf{T}^{\frac{1}{2}}$ is symmetric, and hence $\mathbf{q}_j$ are orthogonal vectors. So, $\mathbf{a}_j$ can be estimated by just following these steps:

i. Construct $\sqrt{n}$-consistent estimators of $\boldsymbol{\Sigma}_Y$ and $\mathbf{T}$.

ii. Use them to estimate $\mathbf{T}^{\frac{1}{2}}\boldsymbol{\Sigma}_Y\mathbf{T}^{\frac{1}{2}}$, and hence $\mathbf{q}_j, j = 1, 2, ..., d$.

iii. From the estimates of $\mathbf{q}_j$, obtain the estimates of $\mathbf{p}_j = \mathbf{T}^{\frac{1}{2}}\mathbf{q}_j$.

iv. Obtain $\mathbf{a}_j$ estimates using the relation

$$\mathbf{a}_j = c_{jj}^{-\frac{1}{2}}\mathbf{p}_j = \frac{\mathbf{p}_j}{||\mathbf{p}_j||}.$$

1.7 Proposed approach

Here we introduce the intuition behind our approach to the estimation problem in ICA — one which is based on distance and relies on the empirical distribution as obtained from the mixed observations. An outline of the idea is provided as follows.

Define $F_n(\mathbf{y} \mid \mathbf{B})$ and $F_n^{\perp}(\mathbf{y} \mid \mathbf{B})$, the joint and product of the marginal empirical distribution of the independent components $\mathbf{X}$, respectively. These are expressed as functions of the observations $\mathbf{Y}$ and the unmixing matrix $\mathbf{B}$, as the independent components X_i themselves are not observable and hence are replaced with the observations as per relation (6), as below:

$$F_n(\mathbf{y} \mid \mathbf{B}) = \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}\{\mathbf{B}\mathbf{Y}_i \leq \mathbf{y}\} \tag{18}$$

$$F_n^{\perp}(\mathbf{y} \mid \mathbf{B}) = \prod_{j=1}^{d} \left[\frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{\{\mathbf{b}_j^{\top} \mathbf{Y}_i \leq y_j\} \right] \tag{19}$$

where $\mathbf{b}_j^{\top}$ denotes the j^{th} row of the matrix $\mathbf{B}$, and $\mathbf{y}$ is a vector consisting of $y_1, ..., y_d$. The entire rationale hinges on the assumption of independence of the components, under which we should have

$$\mathbb{E}[F_n(\mathbf{Y} \mid \mathbf{B})] = F(\mathbf{y} \mid \mathbf{B}) = \prod_{j=1}^{d} F_j(y_j \mid \mathbf{B}) = \mathbb{E}[F_n^{\perp}(\mathbf{Y} \mid \mathbf{B})].$$

As such, we propose to estimate $\mathbf{B}$ based on some predefined metric ρ_w by minimizing the *distance* between (18) and (19); i.e., to estimate $\mathbf{B}$ from

$$\arg\min_{\mathbf{B}} \rho_w \Big(F_n(\mathbf{y} \mid \mathbf{B}), F_n^{\perp}(\mathbf{y} \mid \mathbf{B}) \Big).$$

The following weighted *squared-error statistic* is taken into consideration for theoretical derivations, calculations and simulation studies:

$$\rho_w \coloneqq \rho_w \Big(F_n(\mathbf{y} \mid \mathbf{B}), F_n^{\perp}(\mathbf{y} \mid \mathbf{B}) \Big) = \int_{\mathbb{R}^2} \Big[F_n(\mathbf{y} \mid \mathbf{B}) - F_n^{\perp}(\mathbf{y} \mid \mathbf{B}) \Big]^2 w(\mathbf{y}) \, \mathrm{d}\mathbf{y} \tag{20}$$

with some suitable weight function $w(\cdot)$ of vector $\mathbf{y}$.

Organization of this dissertation

The remainder of the dissertation is structured as follows:

Chapter 2 is devoted to the analysis of the squared-error statistic ρ_w, under the framework of U-statistics. Following the well-established asymptotic theory for U-statistics, we derive a closed-form expression for the asymptotic distribution of the statistic $n\rho_w$. Furthermore, we propose a novel methodology for constructing a confidence region for matrix-valued parameters. Traditional techniques for interval estimation do not readily generalize to the matrix setting, necessitating an alternative approach.

In **Chapter 3**, we investigate the limiting distribution of the proposed estimator through an asymptotic analysis of the associated empirical process. Specifically, we derive an estimator for the

unmixing matrix, highlighting the challenges that arise for higher dimensions as well as when more than one component is Gaussian — an issue that impedes identifiability in Independent Component Analysis (ICA). We further examine the principal components of the empirical process and construct a confidence region using a similar methodology introduced in Chapter 2. Notably, the resulting confidence intervals demonstrate parity between the two approaches.

Finally, **Chapter 4** presents the estimation algorithm to recover the unmixing matrix and, ultimately, to separate the independent components. Given the structure of our estimation procedure — formulated as the minimization of a cost function — we propose an adaptation of the Gradient Descent algorithm tailored to the context of U-statistics. Several modifications are incorporated to account for theoretical considerations and to enhance computational efficiency. The chapter concludes with practical applications of the algorithm, including both visual and quantitative comparisons with the widely used FastICA method.

In the final chapter, we look at the challenges of implementation in higher dimensions and larger datasets and suggest avenues to explore in further research.

Chapter 2

Analysis of the Metric ρ_w as a U-statistic

In this chapter, we rigorously examine the squared-error statistic ρ_w defined in (20), utilizing the well-established theory of U-statistics to derive its asymptotic properties. As with many statistics of this form, like the Cramér–von Mises statistic, we first show that ρ_w is indeed a U-statistic. This provides a clear foundation for developing the theory, deriving the asymptotic distribution, and reflecting on the advantages and disadvantages of using this approach in our setting. We begin by defining U-statistics and concepts associated with them, such as projections, degeneracy problems, and their asymptotic behaviors. We follow a streamlined treatment to derive the asymptotic distribution of ρ_w, based on the usual approach detailed in Serfling (2009).

2.1 U-Statistics preliminaries

Let $X_1, X_2, \ldots$ be independent observations, possibly vector-valued, from a distribution F. Consider a functional $\theta(F)$ and an associated function $\phi(\cdot)$ (which can be assumed to be a symmetric function without any loss of generality) called the *kernel*, such that

$$\mathbb{E}_F(\phi(X_1, \ldots, X_m)) = \theta(F). \tag{21}$$

For any kernel ϕ, the corresponding *U-statistic* is defined as:

$$U_n \equiv U_n(X_1, \ldots, X_n) \coloneqq \frac{1}{\binom{n}{m}} \sum_{\mathcal{C}} \phi(X_{i_1}, \ldots, X_{i_m}),\ m \leq n, \tag{22}$$

where $\mathcal{C}$ denotes the summation over $\binom{n}{m}$ combinations of m distinct elements $i_1, \ldots, i_m$ from $1, \ldots, n$. Clearly, U_n is an unbiased estimator of the parameter θ.

Corresponding to a U-statistc, the associated von Mises statistic, or simply called *V-statistic*, is defined as

$$V_n \equiv V_n(X_1, \ldots, X_n) \coloneqq \frac{1}{n^m} \sum_{i_1=1}^{n} \cdots \sum_{i_m=1}^{n} \phi(X_{i_1}, \ldots, X_{i_m}). \tag{23}$$

2.1.1 Projection

By Hoeffding Decomposition (for more details, see Dynkin and Mandelbaum (1983)), a symmetric kernel $\phi(X_1, \ldots, X_m)$ can be decomposed into mutually uncorrelated terms called *projections*:

$$\phi(X_1, \ldots, X_m) = \mathbb{E}\phi + \sum_{j=1}^{m} \phi^{(1)}(X_j) + \sum_{1 \leq j < k \leq m} \phi^{(2)}(X_j, X_k) + \cdots + \phi^{(m)}(X_1, \ldots, X_m) \tag{24}$$

where $\mathbb{E}\phi \coloneqq \mathbb{E}_F(\phi(X_1, \ldots, X_m))$. We focus only on the second and third terms of (24), as shown later to be the only terms of interest. They are henceforth called the *first-order* and *second-order projections*, respectively, and can be mathematically expressed as:

$$\phi^{(1)}(X_j) = \Big(\mathtt{I} - \mathbb{E}_j\Big) \prod_{l \neq j} \mathbb{E}_l(\phi), \quad \text{and} \tag{25}$$

$$\phi^{(2)}(X_j, X_k) = \Big(\mathtt{I} - \mathbb{E}_j\Big)\Big(\mathtt{I} - \mathbb{E}_k\Big) \prod_{l \neq j,k} \mathbb{E}_l(\phi) \tag{26}$$

where $\mathtt{I}$ is the identity operator, and $\mathbb{E}_i(\phi)$ is the conditional expectation of $\phi(X_1, \ldots, X_m)$ given all X_j except X_i, i.e. $j = 1, \ldots, m;\ j \neq i$.

2.1.2 Limiting distribution of U-statistics

When $\mathbb{E}(\phi^2) < \infty$, a normalized U-statistic

$$n^{\frac{r}{2}}\Big(U_n - \theta\Big) \tag{27}$$

has a limiting distribution, as $n \to \infty$, depending on the projections $\phi^{(1)}, \phi^{(2)}, \ldots$, which determine the value of r. The limits can be divided into a couple of cases as discussed below. A rigorous exposition on how to derive the limiting distribution is provided in Chapter 5 of Serfling (2009), along with examples and further references. Here, we just provide the asymptotic distributions associated with the cases:

Case 1 ($r = 1$):

If $\mathbb{E}(\phi^2) < \infty,\ \mathbb{E}[(\phi^{(1)})]^2 > 0$, then

$$n^{\frac{1}{2}}\Big(U_n - \theta\Big) \xrightarrow{d} N\Big(0,\ m^2\mathbb{E}[\phi^{(1)}(X_1)]^2\Big). \tag{28}$$

This is referred to as the *non-degenerate* case.

Case 2 ($r \geq 2$):

This case obtains when

$$\mathbb{E}[(\phi^{(j)})^2] = 0 \Leftrightarrow \mathbb{P}\{\phi^{(j)} = 0\} = 1,\ j = 1, 2, \ldots, r-1,\ \mathbb{E}[(\phi^{(r)})]^2 > 0,$$

and the limiting distribution is a little more complicated. As $n \longrightarrow \infty$,

$$\mathbb{E}\Big[n^{\frac{r}{2}}(U_n(\phi) - \theta - U_n(\phi^{(r)}))\Big]^2 = n^r\mathcal{O}\Big(\frac{1}{n^{r+1}}\Big) \longrightarrow 0, \tag{29}$$

and hence the limiting distribution of (27) is the same as that of $n^{\frac{r}{2}}U_n(\phi^{(r)})$. This is referred to as the *degenerate* case[1]. For all purposes alluding to our work, we will only require the first-order degenerate ($r = 2$) case.

A succinct explanation of how to derive the limiting distribution, as provided in Dynkin and

[1]Depending on the value of r, we call it $r-1$ order degenerate. For example, if $r = 2$, we call it *First-order degenerate*.

Mandelbaum (1983), is outlined next. When $r = 2$, limiting distribution of $nU_n(\phi)$ is given by a linear function of second-order Hermite polynomials[2] of independent standard Gaussian random variables.

A symmetric, zero-mean, square-integrable function $\phi^{(2)}(x_1, x_2)$ can be decomposed as

$$\phi^{(2)}(x_1, x_2) = \sum_{k=1}^{\infty} \lambda_k \varphi_k(x_1)\varphi_k(x_2) \tag{30}$$

with $\mathbb{E}(\varphi_k) = 0,\ \mathbb{E}(\varphi_k^2) = 1,$ and $\mathbb{E}(\varphi_k\varphi_l) = 0$ if $k \neq l$. Further,

$$\sum_{k=1}^{\infty} \lambda_k^2 < \infty.$$

Note that the Hermite polynomial of degree 2 is $\mathtt{H}_2(z) = z^2 - 1$, and the limiting distribution is given by

$$\frac{n}{\binom{n}{2}} \sum_{1 \leq i < j \leq n} \phi^{(2)}(X_i, X_j) \xrightarrow{d} \sum_{k=1}^{\infty} \lambda_k \left(Z_k^2 - 1 \right) \tag{31}$$

where $Z_1, Z_2, \ldots$ are identical and independent standard Gaussian random variables.

2.2 U-statistic Representation

First, we make some simplifying assumptions. We restrict ourselves to two dimensions[3] and assume the weight function is in product form,

$$w(\mathbf{y}) \coloneqq w_1(y_1)w_2(y_2), \text{ where } \mathbf{y} = (y_1, y_2)^\top,$$

and that, for $k = 1, 2$, the following is true:

$$\int_{-\infty}^{\infty} w_k(u)\, \mathrm{d}u < \infty.$$

[2] Further details on Hermite Polynomials, are discussed in Appendix A.3.

[3] Working with higher dimensions, the sums involved in the U-statistic as well as the projections become unwieldy, which is an inherent problem when dealing with U-statistics, especially one as complicated as ours.

Define the integral of the weight functions as

$$W_k(x) \coloneqq \int_x^\infty w_k(u)\,\mathrm{d}u\,, k = 1, 2.$$

Theorem 2.2.1. *Under the true unmixing matrix, the squared-error statistic ρ_w can be expressed as a U-Statistic. Further, the kernel of this statistic is first-order degenerate.*

Proof. Assume that $\mathbf{B}$ is the true unmixing matrix, as depicted in the relation (6). So we have the relation $\mathbf{X}_i = \mathbf{B}\mathbf{Y}_i \iff \begin{bmatrix} X_{1i} \\ X_{2i} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1^\top \\ \mathbf{b}_2^\top \end{bmatrix} \mathbf{Y}_i$. On expanding (20), in the 2D case, we obtain:

$$\begin{aligned}
\rho_w &= \int_{\mathbb{R}^2} \left[[F_n(\mathbf{y} \mid \mathbf{B})]^2 + \left[F_n^\perp(\mathbf{y} \mid \mathbf{B})\right]^2 - 2F_n(\mathbf{y} \mid \mathbf{B})F_n^\perp(\mathbf{y} \mid \mathbf{B}) \right] w(\mathbf{y})\,\mathrm{d}\mathbf{y} \\
&= \int_{\mathbb{R}^2} \Bigg[\frac{1}{n^2} \sum_i \sum_j \mathbb{1}\{\mathbf{b}_1^\top \mathbf{Y}_i \le y_1\}\mathbb{1}\{\mathbf{b}_2^\top \mathbf{Y}_i \le y_2\}\mathbb{1}\{\mathbf{b}_1^\top \mathbf{Y}_j \le y_1\}\mathbb{1}\{\mathbf{b}_2^\top \mathbf{Y}_j \le y_2\} \\
&\quad + \frac{1}{n^4} \sum_i \sum_j \sum_k \sum_l \mathbb{1}\{\mathbf{b}_1^\top \mathbf{Y}_i \le y_1\}\mathbb{1}\{\mathbf{b}_2^\top \mathbf{Y}_j \le y_2\}\mathbb{1}\{\mathbf{b}_1^\top \mathbf{Y}_k \le y_1\}\mathbb{1}\{\mathbf{b}_2^\top \mathbf{Y}_l \le y_2\} \\
&\quad - \frac{2}{n^3} \sum_i \sum_j \sum_k \mathbb{1}\{\mathbf{b}_1^\top \mathbf{Y}_i \le y_1\}\mathbb{1}\{\mathbf{b}_2^\top \mathbf{Y}_i \le y_2\}\mathbb{1}\{\mathbf{b}_1^\top \mathbf{Y}_j \le y_1\}\mathbb{1}\{\mathbf{b}_2^\top \mathbf{Y}_k \le y_2\} \Bigg] w(\mathbf{y})\,\mathrm{d}\mathbf{y} \\
&= \frac{1}{n^2} \sum_i \sum_j W_1\left(\mathbf{b}_1^\top \mathbf{Y}_i \vee \mathbf{b}_1^\top \mathbf{Y}_j\right) W_2\left(\mathbf{b}_2^\top \mathbf{Y}_i \vee \mathbf{b}_2^\top \mathbf{Y}_j\right) \\
&\quad + \frac{1}{n^4} \sum_i \sum_j \sum_k \sum_l W_1\left(\mathbf{b}_1^\top \mathbf{Y}_i \vee \mathbf{b}_1^\top \mathbf{Y}_k\right) W_2\left(\mathbf{b}_2^\top \mathbf{Y}_j \vee \mathbf{b}_2^\top \mathbf{Y}_l\right) \\
&\quad - \frac{2}{n^3} \sum_i \sum_j \sum_k W_1\left(\mathbf{b}_1^\top \mathbf{Y}_i \vee \mathbf{b}_1^\top \mathbf{Y}_j\right) W_2\left(\mathbf{b}_2^\top \mathbf{Y}_i \vee \mathbf{b}_2^\top \mathbf{Y}_k\right) \\
&= \frac{1}{n^2} \sum_i \sum_j W_1\left(X_{1i} \vee X_{1j}\right) W_2\left(X_{2i} \vee X_{2j}\right) \\
&\quad + \frac{1}{n^4} \sum_i \sum_j \sum_k \sum_l W_1\left(X_{1i} \vee X_{1k}\right) W_2\left(X_{2j} \vee X_{2l}\right) \\
&\quad - \frac{2}{n^3} \sum_i \sum_j \sum_k W_1\left(X_{1i} \vee X_{1j}\right) W_2\left(X_{2i} \vee X_{2k}\right) \\
&=: S_1 + S_2 - 2S_3, \text{ (say).}
\end{aligned}$$

The above is a linear combination of three V-statistics, S_1, S_2 and S_3, where,

$$W_k(x) = \int_x^\infty w_k(u)\, \mathrm{d}u,\ k = 1, 2.$$

For the next portion of the section, and the sake of simplicity[4], we rewrite X_{1i}, the i^{th} observation of X_1 as U_i, and X_{2i}, the i^{th} observation of X_2 as V_i. We also rewrite the PDF and CDF of $X_1 = U$ as f and F respectively; and that of $X_2 = V$ as g and G, respectively. Thus,

$$\begin{aligned}
S_1 =& \frac{1}{n^2} \sum_i \sum_j W_1(U_i \vee U_j) W_2(V_i \vee V_j), \\
S_2 =& \frac{1}{n^4} \sum_i \sum_j \sum_k \sum_l W_1(U_i \vee U_j) W_2(V_k \vee V_l), \\
S_3 =& \frac{1}{n^3} \sum_i \sum_j \sum_k W_1(U_i \vee U_j) W_2(V_i \vee V_k),
\end{aligned}$$

and the symmetric kernels corresponding to $S_1, S_2, \& S_3$ are as follows:

$$\begin{aligned}
\phi_1 \coloneqq& \phi_1\left(\begin{pmatrix} u_i \\ v_i \end{pmatrix}, \begin{pmatrix} u_j \\ v_j \end{pmatrix} \right) = W_1(u_i \vee u_j) W_2(v_i \vee v_j), \\
\phi_2 \coloneqq& \phi_2\left(\begin{pmatrix} u_i \\ v_i \end{pmatrix}, \begin{pmatrix} u_j \\ v_j \end{pmatrix}, \begin{pmatrix} u_k \\ v_k \end{pmatrix}, \begin{pmatrix} u_l \\ v_l \end{pmatrix} \right) = \frac{1}{6} \sum_\pi W_1(u_i \vee u_j) W_2(v_k \vee v_l), \\
\phi_3 \coloneqq& \phi_3\left(\begin{pmatrix} u_i \\ v_i \end{pmatrix}, \begin{pmatrix} u_j \\ v_j \end{pmatrix}, \begin{pmatrix} u_k \\ v_k \end{pmatrix} \right) = \frac{1}{6} \sum_\pi W_1(u_i \vee u_j) W_2(v_i \vee v_k),
\end{aligned}$$

the sums in ϕ_2 and ϕ_3 being taken over possible permutations to make them permutation invariant

[4]This change of notation, while not necessary, makes the following discussion material legible and easier to follow with the removal of an extra subscript.

(symmetric) kernels. Next, we define, for $i = 1, ..., n$, the following:

$$\begin{aligned} h_1(u_i) &\coloneqq \mathbb{E}[W_1(u_i \vee U)], \\ h_2(v_i) &\coloneqq \mathbb{E}[W_2(v_i \vee V)], \\ \mathbb{E}h_1 &\coloneqq \mathbb{E}[W_1(U_1 \vee U_2)], \\ \mathbb{E}h_2 &\coloneqq \mathbb{E}[W_2(V_1 \vee V_2)] \end{aligned}$$

It is a simple task to show that $\mathbb{E}\phi_1 = \mathbb{E}\phi_2 = \mathbb{E}\phi_3 = \mathbb{E}h_1 \cdot \mathbb{E}h_2$, which we will just denote as $\mathbb{E}\phi$. For S_1, with kernel ϕ_1 and expectation $\mathbb{E}\phi_1 = \mathbb{E}\phi$, the first-order and second-order projections are, respectively, given by:

$$\begin{aligned} \phi_1^{(1)}\left(\begin{pmatrix} u_i \\ v_i \end{pmatrix}\right) =& 2\left[\int\int W_1(u \vee u_i)W_2(v \vee v_i)\ f(u)g(v)\ \mathrm{d}u\mathrm{d}v - \mathbb{E}\phi\right] \\ =& 2h_1(u_i)h_2(v_i) - 2\mathbb{E}\phi', \end{aligned} \tag{32}$$

and

$$\begin{aligned} \phi_1^{(2)}\left(\begin{pmatrix} u_i \\ v_i \end{pmatrix}, \begin{pmatrix} u_j \\ v_j \end{pmatrix}\right) =& W_1(u_i \vee u_j)W_2(v_i \vee v_j) - \int\int W_1(u_i \vee u)W_2(v_i \vee v)\ f(u)g(v)\ \mathrm{d}u\mathrm{d}v \\ & - \int\int W_1(u \vee u_j)W_2(v \vee v_j)\ f(u)g(v)\ \mathrm{d}u\mathrm{d}v + \mathbb{E}\phi \\ =& W_1(u_i \vee u_j)W_2(v_i \vee v_j) - h_1(u_i)h_2(v_i) - h_1(u_j)h_2(v_j) + \mathbb{E}\phi. \end{aligned} \tag{33}$$

Similarly, for S_2, the kernel is ϕ_2 with expectation $\mathbb{E}\phi_2 = \mathbb{E}\phi$. The first-order projection is

$$\begin{aligned} \phi_2^{(1)}\left(\begin{pmatrix} u_i \\ v_i \end{pmatrix}\right) =& 4\left[\frac{1}{6}\left[3\int\int\int W_1(u_i \vee u)W_2(v \vee v')\ f(u)g(v)g(v')\ \mathrm{d}u\mathrm{d}v\mathrm{d}v'\right.\right. \\ & \left.\left. + 3\int\int\int W_1(u \vee u')W_2(u_i \vee v)\ f(u)f(u')g(v)\ \mathrm{d}u\mathrm{d}u'\mathrm{d}v\right] - \mathbb{E}\phi\right] \\ =& 2\left[h_1(u_i)\mathbb{E}h_2 + h_2(v_i)\mathbb{E}h_1\right] - 4\mathbb{E}\phi. \end{aligned} \tag{34}$$

while the second-order projection, by straightforward calculations, is

$$\begin{aligned}\phi_2^{(2)}\left(\begin{pmatrix}u_i\\v_i\end{pmatrix},\begin{pmatrix}u_j\\v_j\end{pmatrix}\right)=&\binom{4}{2}\left(\frac{1}{6}\Big[\mathbb{E}h_2\cdot W_1(u_i\vee u_j)+2h_1(u_i)h_2(v_j)+2h_1(u_j)h_2(v_i)\right.\\&+\mathbb{E}h_1\cdot W_2(v_i\vee v_j)\Big]-\frac{1}{6}\Big[3h_1(u_i)\mathbb{E}h_2+3h_2(v_i)\mathbb{E}h_1\Big]\\&\left.-\frac{1}{6}\Big[3h_1(u_j)\mathbb{E}h_2+3h_2(v_j)\mathbb{E}h_1\Big]+\mathbb{E}\phi\right).\end{aligned}\tag{35}$$

Lastly, for S_3 the kernel is ϕ_3, and expectation $\mathbb{E}\phi_3=\mathbb{E}\phi$. The first-order projection is

$$\begin{aligned}\phi_3^{(1)}\left(\begin{pmatrix}u_i\\v_i\end{pmatrix}\right)=&3\left[\frac{1}{6}\left[2\int\int W_1(u_i\vee u)\,W_2(v_i\vee v)\,f(u)g(v)\,\mathrm{d}u\mathrm{d}v\right.\right.\\&+2\int\int\int W_1(u_i\vee u)\,W_2(v\vee v')\,f(u)g(v)g(v')\,\mathrm{d}u\mathrm{d}v\mathrm{d}v'\\&\left.\left.+2\int\int\int W_1(u\vee u')\,W_2(v_i\vee v)\,f(u)f(u')g(v)\,\mathrm{d}u\mathrm{d}u'\mathrm{d}v\right]-\mathbb{E}\phi\right]\\=&h_1(u_i)h_2(v_i)+h_1(u_i)\mathbb{E}h_2+h_2(v_i)\mathbb{E}h_1-3\mathbb{E}\phi.\end{aligned}\tag{36}$$

and the second-order projection is obtained as

$$\begin{aligned}\phi_3^{(2)}\left(\begin{pmatrix}u_i\\v_i\end{pmatrix},\begin{pmatrix}u_j\\v_j\end{pmatrix}\right)=&\binom{3}{2}\left(\frac{1}{6}\Big[W_1(u_i\vee u_j)h_2(v_i)+W_1(u_i\vee u_j)h_2(v_j)+W_2(v_i\vee v_j)h_1(u_i)\right.\\&\left.+W_2(v_i\vee v_j)h_2(u_j)+h_1(u_i)h_2(v_j)+h_1(u_j)h_2(u_i)\Big]\right.\\&-\frac{1}{6}\Big[2h_1(u_i)h_2(v_i)+2h_1(u_i)\mathbb{E}h_2+2h_2(v_i)\mathbb{E}h_1\Big]\\&\left.-\frac{1}{6}\Big[2h_1(u_j)h_2(v_j)+2h_1(u_j)\mathbb{E}h_2+2h_2(v_j)\mathbb{E}h_1\Big]+\mathbb{E}\phi\right).\end{aligned}\tag{37}$$

The expectation of ρ_w is $\mathbb{E}\rho_w=\mathbb{E}\phi_1+\mathbb{E}\phi_2-2\mathbb{E}\phi_3=0$. Combining (32), (34) and (36), it

becomes clear that the first-order projection of the statistic is degenerate, i.e.,

$$\begin{aligned}
&\phi_1^{(1)}\begin{pmatrix}u_i\\ \\ v_i\end{pmatrix} + \phi_2^{(1)}\begin{pmatrix}u_i\\ \\ v_i\end{pmatrix} - 2\phi_3^{(1)}\begin{pmatrix}u_i\\ \\ v_i\end{pmatrix} \\
&= 2h_1(u_i)h_2(v_i) - 2\mathbb{E}\phi + 2\Big[h_1(u_i)\mathbb{E}h_2 + h_2(v_i)\mathbb{E}h_1\Big] - 4\mathbb{E}\phi \\
&- 2\Big[h_1(u_i)h_2(v_i) + h_1(u_i)\mathbb{E}h_2 + h_2(v_i)\mathbb{E}h_1 - 3\mathbb{E}\phi\Big] \\
&= 0.
\end{aligned}$$

■

While the first-order projection is degenerate, the second-order projection isn't. However, it has quite a simple form, as is revealed below:

$$\begin{aligned}
\phi_1^{(2)} + \phi_2^{(2)} - 2\phi_3^{(2)} =& \Big[W_1(u_i \vee u_j)W_2(v_i \vee v_j) - h_1(u_i)h_2(v_i) - h_1(u_j)h_2(v_j) + \mathbb{E}\phi\Big] \\
&+ \Big[\Big[\mathbb{E}h_2 \cdot W_1(u_i \vee u_j) + 2h_1(u_i)h_2(v_j) + 2h_1(u_j)h_2(v_i) + \mathbb{E}h_1 \cdot W_2(v_i \vee v_j)\Big] \\
&- \Big[3h_1(u_i)\mathbb{E}h_2 + 3h_2(v_i)\mathbb{E}h_1\Big] - \Big[3h_1(u_j)\mathbb{E}h_2 + 3h_2(v_j)\mathbb{E}h_1\Big] + 6\mathbb{E}\phi\Big] \\
&- \Big[\Big[W_1(u_i \vee u_j)h_2(v_i) + W_1(u_i \vee u_j)h_2(v_j) + W_2(v_i \vee v_j)h_1(u_i) \\
&+ W_2(v_i \vee v_j)h_2(u_j) + h_1(u_i)h_2(v_j) + h_1(u_j)h_2(u_i)\Big] \\
&- \Big[2h_1(u_i)h_2(v_i) + 2h_1(u_i)\mathbb{E}h_2 + 2h_2(v_i)\mathbb{E}h_1\Big] \\
&- \Big[2h_1(u_j)h_2(v_j) + 2h_1(u_j)\mathbb{E}h_2 + 2h_2(v_j)\mathbb{E}h_1\Big] + 6\mathbb{E}\phi\Big].
\end{aligned}$$

Rearranging the terms in the above expression, shows that the second-order projection, while not degenerate, is just a product of two terms — each term is dependent on one of the two independent components.

$$
\begin{aligned}
\phi_1^{(2)} + \phi_2^{(2)} - 2\phi_3^{(2)} =& \Big[W_1(u_i \vee u_j) W_2(v_i \vee v_j) + \mathbb{E}h_2 \cdot W_1(u_i \vee u_j) + \mathbb{E}h_1 \cdot W_2(v_i \vee v_j) \\
& + 2h_1(u_i)h_2(v_j) + 2h_1(u_j)h_2(v_i) - h_1(u_i)h_2(v_j) - h_1(u_j)h_2(u_i) \\
& - W_1(u_i \vee u_j)h_2(v_i) - W_1(u_i \vee u_j)h_2(v_j) \\
& - W_2(v_i \vee v_j)h_1(u_i) - W_2(v_i \vee v_j)h_2(u_j) \Big] \\
& - \Big[h_1(u_i)h_2(v_i) + 3h_1(u_i)\mathbb{E}h_2 + 3h_2(v_i)\mathbb{E}h_1 \\
& - 2h_1(u_i)h_2(v_i) - 2h_1(u_i)\mathbb{E}h_2 - 2h_2(v_i)\mathbb{E}h_1 \Big] \\
& - \Big[h_1(u_j)h_2(v_j) + 3h_1(u_j)\mathbb{E}h_2 + 3h_2(v_j)\mathbb{E}h_1 \\
& - 2h_1(u_j)h_2(v_j) - 2h_1(u_j)\mathbb{E}h_2 - 2h_2(v_j)\mathbb{E}h_1 \Big] + \mathbb{E}\phi \\
=& \Big[W_1(u_i \vee u_j) W_2(v_i \vee v_j) + \mathbb{E}h_2 \cdot W_1(u_i \vee u_j) + \mathbb{E}h_1 \cdot W_2(v_i \vee v_j) \\
& + h_1(u_i)h_2(v_j) + h_1(u_j)h_2(v_i) - W_1(u_i \vee u_j)h_2(v_i) \\
& - W_1(u_i \vee u_j)h_2(v_j) - W_2(v_i \vee v_j)h_1(u_i) - W_2(v_i \vee v_j)h_2(u_j) \Big] \\
& - \Big[- h_1(u_i)h_2(v_i) + h_1(u_i)\mathbb{E}h_2 + h_2(v_i)\mathbb{E}h_1 \Big] \\
& - \Big[- h_1(u_j)h_2(v_j) + h_1(u_j)\mathbb{E}h_2 + h_2(v_j)\mathbb{E}h_1 \Big] + \mathbb{E}h_1 \cdot \mathbb{E}h_2 \\
=& \Big[W_1(u_i \vee u_j) - h_1(u_i) - h_1(u_j) + \mathbb{E}h_1 \Big] \\
& \Big[W_2(v_i \vee v_j) - h_2(v_i) - h_2(v_j) + \mathbb{E}h_2 \Big] \qquad (38) \\
=& \alpha(u_i, u_j) \cdot \beta(v_i, v_j) =: \phi^{(2)} \left(\begin{pmatrix} u_i \\ v_i \end{pmatrix}, \begin{pmatrix} u_j \\ v_j \end{pmatrix} \right), \qquad (39)
\end{aligned}
$$

where $\alpha(u_i, u_j) = W_1(u_i \vee u_j) - h_1(u_i) - h_1(u_j) + \mathbb{E}h_1$ and $\beta(v_i, v_j) = W_2(v_i \vee v_j) - h_2(v_i) - h_2(v_j) + \mathbb{E}h_2$. As pointed out in Case 2 of Section 2.1.2, the limiting distribution should follow from the above.

2.3 Asymptotic Distribution of ρ_w

In the light of the discussion so far, the form of the limiting distribution is obtainable. First note that both $\alpha(u_i, u_j)$ and $\beta(v_i, v_j)$ can be represented as infinte series of the form (30):

$$\alpha(u_1, u_2) = \sum_{k=1}^{\infty} \lambda_{1k}\varphi_{1k}(u_1)\varphi_{1k}(u_2); \quad \beta(v_1, v_2) = \sum_{l=1}^{\infty} \lambda_{2l}\varphi_{2l}(v_1)\varphi_{2l}(v_2). \tag{40}$$

with both of the following series being finite:

$$\sum_{k=1}^{\infty} \lambda_{1k}^2 < \infty \text{ and } \sum_{l=1}^{\infty} \lambda_{2l}^2 < \infty.$$

Further, the statistic $n\rho_w$ is first-order degenerate and should have the same limiting distribution as

$$nU_n(\phi^{(2)}) = \frac{n}{\binom{n}{2}} \sum_{i<j} \phi^{(2)}\left(\begin{pmatrix} u_i \\ v_i \end{pmatrix}, \begin{pmatrix} u_j \\ v_j \end{pmatrix} \right). \tag{41}$$

It follows from (40), (41) and Section 2.1.2 that

$$\begin{aligned} nU_n(\phi^{(2)}) =& \frac{n}{\binom{n}{2}} \sum_{i<j} \alpha(u_i, u_j)\beta(v_i, v_j) \\ =& \frac{n}{\binom{n}{2}} \sum_{k=1}^{\infty}\sum_{l=1}^{\infty} \lambda_{1k}\lambda_{2l}\left[\sum_{i<j} \varphi_{1k}(u_i)\varphi_{1k}(u_j)\varphi_{2l}(v_i)\varphi_{2l}(v_j) \right] \end{aligned}$$

Hence, we have the following result.

Theorem 2.3.1. *The limiting distribution of ρ_w is given by*

$$n\rho_w \xrightarrow{d} \sum_{k=1}^{\infty}\sum_{l=1}^{\infty} \lambda_{1k}\lambda_{2l}\left[Z_{k,l}^2 - 1 \right], \tag{42}$$

where $Z_{k,l}$ are identical and independent standard Gaussian random variables.

2.4 Confidence Region for the Unmixing Matrix

In ICA where the ultimate goal is to recover independent components from observed data, the unmixing matrix plays a crucial role. Our method, as well as many others, rely on estimating this matrix first. It is thus critical to quantify the uncertainty surrounding this estimated parameter matrix. One such approach is constructing a confidence region for the parameter. However, due to inherent ambiguities (cf. Section 1.3) in ICA — where solutions are only determined up to scaling and rotation — coupled with the fact that the parameter is a matrix, constructing a meaningful confidence region for $\mathbf{B}$ presents a unique challenge.

2.4.1 Confidence Region Setup

Here, we discuss a novel way of constructing a confidence region for the unmixing matrix $\mathbf{B}$ within the ICA framework.

In the 2D case, given observed data $\mathbf{Y}$, modeled as $\mathbf{Y} = \mathbf{A}\mathbf{X} \iff \mathbf{X} = \mathbf{B}\mathbf{Y}$, where $\mathbf{A}$ is the invertible mixing matrix, $\mathbf{X}$ are the independent components, our objective is to construct a confidence region for $\mathbf{B} = \mathbf{A}^{-1}$. The following assumptions apply to the sources X_i:

(1) they are independent;

(2) they have unit variance.

The dispersion matrix of $\mathbf{Y}$, denoted by $\mathbf{\Sigma}_y$, satisfies

$$\mathbf{B}\mathbf{\Sigma}_y\mathbf{B}^\top = \mathbf{I}. \tag{43}$$

This equation shows that, under the above two assumptions, one needs to search for B only among matrices that diagonalize $\mathbf{\Sigma}_y$. An initial estimate of $\mathbf{B}$ is obtained through the EVD of $\mathbf{\Sigma}_y$:

$$\mathbf{\Sigma}_y = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^\top$$

where $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2)$ is the diagonal matrix of eigenvalues of $\mathbf{\Sigma}_y$ and $\mathbf{P} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 \end{bmatrix}$ consists

of the corresponding eigenvectors. Naturally, a candidate for $\mathbf{B}$ is then

$$\bar{\mathbf{B}} = \begin{bmatrix} \frac{1}{\sqrt{\lambda_1}}\mathbf{p}_1 & \frac{1}{\sqrt{\lambda_2}}\mathbf{p}_2 \end{bmatrix}. \tag{44}$$

Now, for an orthogonal matrix $\mathbf{C}$ satisfying $\mathbf{C}\mathbf{C}^\top = \mathbf{I}$, we see that $\mathbf{B} = \mathbf{C}\bar{\mathbf{B}}$ satisfies

$$\mathbf{B}\boldsymbol{\Sigma}_y\mathbf{B} = \mathbf{C}\bar{\mathbf{B}}\boldsymbol{\Sigma}_y\bar{\mathbf{B}}^\top\mathbf{C}^\top = \mathbf{I}. \tag{45}$$

This implies that we can restrict our search for $\mathbf{B}$ within the class of matrices that satisfies (45):

$$\mathfrak{B} = \{\mathbf{B} = \mathbf{C}\bar{\mathbf{B}} \mid \mathbf{C} \text{ is orthogonal}\}.$$

Of course, it's well known that all 2×2 orthogonal matrices with positive determinant can be represented as a rotation (or a reflection if the determinant is negative)[5], and so the matrix

$$\mathbf{C} = \mathbf{C}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \ \theta \in [0, 2\pi],$$

is a rotation (anticlockwise) matrix parameterized by the angle θ and thereby making all the choices for $\mathbf{B}$ dependent on the angle of rotation θ, i.e. $\mathbf{B}_\theta = \mathbf{C}_\theta\bar{\mathbf{B}}$. Our search is now constrained to rotations of the matrix $\bar{\mathbf{B}}$. Given data $\mathbf{Y}$, we can plot $\rho_w(\theta)$, treated as a function of θ, against θ for corresponding candidate $\mathbf{B}_\theta$, which in turn can be compared to the null distribution[6] of ρ_w to determine the suitability of candidates $\mathbf{B}_\theta \in \mathfrak{B}$.

The family $\mathfrak{B}$ of candidate unmixing matrices $\mathbf{B}_\theta$ is parameterized by the angle θ of matrix $\mathbf{C}_\theta$. Evaluating $\rho_w(\theta)$ over this family allows us to assess the plausibility of different unmixing matrices. It comes as no surprise that, due to the fundamental ambiguities inherent to ICA, $\rho_w(\theta)$ will experience a periodicity of $\frac{\pi}{2}$.

Note that the matrix $\mathbf{C}_{\theta+\frac{\pi}{2}}$ can be expressed as a composition of a fixed anticlockwise $\frac{\pi}{2}$-rotation

[5] cf. Result A.4.1 in Appendix A.4.

[6] The distribution of ρ_w under the true unmixing matrix.

on $\mathbf{C}_\theta$:

$$\mathbf{C}_{\theta+\frac{\pi}{2}} = \mathbf{C}_{\frac{\pi}{2}}\mathbf{C}_\theta, \text{ where } \mathbf{C}_{\frac{\pi}{2}} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

This implies that $\mathbf{B}_{\theta+\frac{\pi}{2}} = \mathbf{C}_{\frac{\pi}{2}}\mathbf{B}_\theta$, which represents a $\frac{\pi}{2}$-rotated version of $\mathbf{B}_\theta$. Geometrically, this corresponds to a cyclical reorientation of the corresponding vectors, effectively permuting the independent components and swapping one of their signs. Due to the well-known indeterminacies in ICA, namely:

(1) permutation of independent components; and

(2) sign problem (multiplying a source by -1 yields an equally valid source estimate).

It follows that $\mathbf{B}_{\theta+\frac{\pi}{2}}$ and $\mathbf{B}_\theta$ are equivalent in the ICA context. That is, if $\mathbf{B}_\theta\mathbf{Y}$ is valid in this context, then so is $\mathbf{B}_{\theta+\frac{\pi}{2}}\mathbf{Y}$, with the latter being a permuted and/or sign swapped version of the former. Since ρ_w, which is designed to evaluate the statistical independence of $\mathbf{X}_\theta = \mathbf{B}_\theta\mathbf{Y}$,[7] must be invariant under these ICA transformations. Consequently, $\rho_w(\theta)$ satisfies:

$$\rho_w(\theta) = \rho_w(\theta + \tfrac{\pi}{2}),$$

and thus, exhibits a $\frac{\pi}{2}$-periodicity. This property is empirically confirmed in the examples that follow and can be visualized in Figures 2.2 and 2.4.

2.4.2 Special Weights for the Statistic

While we can show convergence of $n\rho_w$ to a limiting distribution as done in Theorem 2.3.1, a confidence region for $\mathbf{B}$ is rather inconvenient to design around it. Fortunately, considering a simple change to the original formulation of ρ_w can ease the process. As shown later in this section, this alteration provides a massive benefit — it makes the null distribution of ρ_w independent of the underlying distributions of the independent components. This allows us to simulate the null distribution of ρ_w and obtain the 95% cut-off points.

[7] The independent component estimates from the observations $\mathbf{Y}$ *unmixed* by the candidate unmixing matrix $\mathbf{B}_\theta$, which of course is characterized here by the angle θ.

In the ICA scenario, the independent components and underlying densities are unknown. The only assumptions on these densities are that they are independent and that no more than one is Gaussian. The examples used throughout our work reflect this. Since the testing or estimation procedures are simulation studies, the source signals have to be generated in the first place. Established methods tend to treat these unknown densities as nuisance parameters, either estimating them empirically from the data or finding some clever way to bypass them (cf. Section 1.6).

The novelty of our method is its simplicity in making the null distribution invariant to the distributions of independent components (F_1 and F_2). To achieve this, we use a special form of the weight function in the expression of ρ_w (20). We assume that the weight function $w(\mathbf{x})$ is not only in product form $w(\mathbf{x}) = w_1(x_1) \cdot w_2(x_2)$, but also w_i is related to X_i through their distribution function F_i and the density function f_i, viz.,

$$w_i(F_i(x_i))\ f_i(x_i),\ i = 1, 2. \tag{46}$$

Then we have, following notations established in Section 1.7,

$$\rho_w = \int_{\mathbb{R}^2} \Big[F_n(\mathbf{y} \mid \mathbf{B}) - F_n^{\perp}(\mathbf{y} \mid \mathbf{B}) \Big]^2 w_1(F_1(y_1)) f_1(y_1)\ w_2(F_2(y_2)) f_2(y_2)\ \mathrm{d}y_1\ \mathrm{d}y_2 \tag{47}$$

Simply expanding the above expression leads to a U-statistic in the same way we had obtained it in Section 2.2, splitting ρ_w into 3 terms:

$$\begin{aligned}
\rho_w = \int_{\mathbb{R}^2} \Bigg[& \frac{1}{n^2} \sum_i \sum_j \mathbb{1}\{\mathbf{b}_1^\top \mathbf{y}_i \le y_1\} \mathbb{1}\{\mathbf{b}_1^\top \mathbf{y}_j \le y_1\} \mathbb{1}\{\mathbf{b}_2^\top \mathbf{y}_i \le y_2\} \mathbb{1}\{\mathbf{b}_2^\top \mathbf{y}_j \le y_2\} \\
& + \frac{1}{n^4} \sum_i \sum_j \sum_k \sum_l \mathbb{1}\{\mathbf{b}_1^\top \mathbf{y}_i \le y_1\} \mathbb{1}\{\mathbf{b}_1^\top \mathbf{y}_j \le y_1\} \mathbb{1}\{\mathbf{b}_2^\top \mathbf{y}_k \le y_2\} \mathbb{1}\{\mathbf{b}_2^\top \mathbf{y}_l \le y_2\} \\
& - \frac{1}{n^4} \sum_i \sum_j \sum_k \mathbb{1}\{\mathbf{b}_1^\top \mathbf{y}_i \le y_1\} \mathbb{1}\{\mathbf{b}_1^\top \mathbf{y}_j \le y_1\} \mathbb{1}\{\mathbf{b}_2^\top \mathbf{y}_i \le y_2\} \mathbb{1}\{\mathbf{b}_2^\top \mathbf{y}_k \le y_2\} \Bigg] \\
& w_1(F_1(y_1)) f_1(y_1)\ w_2(F_2(y_2)) f_1(y_2)\ \mathrm{d}y_1\ \mathrm{d}y_2 \\
= & \text{ term 1} + \text{term 2} + \text{term 3}.
\end{aligned}$$

Below we show the simplification of term 1:

$$\begin{aligned}
\text{term 1} = &\int_{\mathbb{R}^2} \left[\frac{1}{n^2} \sum_i \sum_j \mathbb{1}\{\mathbf{b}_1^\top \mathbf{y}_i \le y_1\} \mathbb{1}\{\mathbf{b}_1^\top \mathbf{y}_j \le y_1\} \mathbb{1}\{\mathbf{b}_2^\top \mathbf{y}_i \le y_2\} \mathbb{1}\{\mathbf{b}_2^\top \mathbf{y}_j \le y_2\} \right] \\
& w_1(F_1(y_1)) w_2(F_2(y_2)) \, \mathrm{d}F_1(y_1) \, \mathrm{d}F_2(y_2) \\
= &\frac{1}{n^2} \sum_i \sum_j \left[\int_{\mathbb{R}} \mathbb{1}\{\mathbf{b}_1^\top \mathbf{y}_i \le y_1\} \mathbb{1}\{\mathbf{b}_1^\top \mathbf{y}_j \le y_1\} w_1(F_1(y_1)) \mathrm{d}F_1(y_1) \right] \\
& \left[\int_{\mathbb{R}} \mathbb{1}\{\mathbf{b}_2^\top \mathbf{y}_i \le y_2\} \mathbb{1}\{\mathbf{b}_2^\top \mathbf{y}_j \le y_2\} w_2(F_2(y_2)) \mathrm{d}F_2(y_2) \right].
\end{aligned}$$

Let $F_1(y_1) = u$, $F_2(y_2) = v$. Then $\mathrm{d}F_1(y_1) = \mathrm{d}u$, $\mathrm{d}F_2(y_2) = \mathrm{d}v$, and $y_1 = F_1^{-1}(u)$, $y_2 = F_2^{-1}(v)$, while the supports of both the integrals change from $\mathbb{R}$ to the interval $[0, 1]$. Thus,

$$\begin{aligned}
\text{term 1} = &\frac{1}{n^2} \sum_i \sum_j \left[\int_0^1 \mathbb{1}\{\mathbf{b}_1^\top \mathbf{y}_i \le F_1^{-1}(u)\} \mathbb{1}\{\mathbf{b}_1^\top \mathbf{y}_j \le F_1^{-1}(u)\} w_1(u) \mathrm{d}u \right] \\
& \left[\int_0^1 \mathbb{1}\{\mathbf{b}_2^\top \mathbf{y}_i \le F_2^{-1}(v)\} \mathbb{1}\{\mathbf{b}_2^\top \mathbf{y}_j \le F_2^{-1}(v)\} w_2(v) \mathrm{d}v \right] \\
= &\frac{1}{n^2} \sum_i \sum_j \left[\int_0^1 \mathbb{1}\{u \ge F_1(\mathbf{b}_1^\top \mathbf{y}_i) \vee F_1(\mathbf{b}_1^\top \mathbf{y}_j)\} w_1(u) \, \mathrm{d}u \right] \\
& \left[\int_0^1 \mathbb{1}\{v \ge F_2(\mathbf{b}_2^\top \mathbf{y}_i) \vee F_2(\mathbf{b}_2^\top \mathbf{y}_j)\} w_2(v) \, \mathrm{d}v \right] \\
= &\frac{1}{n^2} \sum_i \sum_j \left[\int_{F_1(\mathbf{b}_1^\top \mathbf{y}_i) \vee F_1(\mathbf{b}_1^\top \mathbf{y}_j)}^1 w_1(u) \, \mathrm{d}u \right] \left[\int_{F_2(\mathbf{b}_2^\top \mathbf{y}_i) \vee F_2(\mathbf{b}_2^\top \mathbf{y}_j)}^1 w_2(v) \, \mathrm{d}v \right] \\
= &\frac{1}{n^2} \sum_i \sum_j W_1\Big(F_1(\mathbf{b}_1^\top \mathbf{y}_i) \vee F_1(\mathbf{b}_1^\top \mathbf{y}_j) \Big) \, W_2\Big(F_2(\mathbf{b}_2^\top \mathbf{y}_i) \vee F_2(\mathbf{b}_2^\top \mathbf{y}_j) \Big),
\end{aligned}$$

where $W_i(x) = \int_x^1 w_i(y) \mathrm{d}y;\ i = 1, 2$, and under the true mixing matrix, $\mathbf{b}_i^\top \mathbf{y}_j = X_{ij},\ i = 1, 2, j = 1, 2, \ldots n$,, thereby yielding

$$\text{term 1} = \frac{1}{n^2} \sum_i \sum_j W_1\Big(F_1(X_{1i}) \vee F_1(X_{1j}) \Big) \, W_2\Big(F_2(X_{2i}) \vee F_2(X_{2j}) \Big).$$

The other two terms, term 2 and term 3, can be analogously simplified, resulting in

$$\begin{aligned}\rho_w =& \frac{1}{n^2}\sum_i\sum_j W_1\Big(F_1(\mathbf{b}_1^\top\mathbf{y}_i)\vee F_1(\mathbf{b}_1^\top\mathbf{y}_j)\Big)\,W_2\Big(F_2(\mathbf{b}_2^\top\mathbf{y}_i)\vee F_1(\mathbf{b}_2^\top\mathbf{y}_j)\Big)\\ &+\frac{1}{n^4}\sum_i\sum_j\sum_k\sum_l W_1\Big(F_1(\mathbf{b}_1^\top\mathbf{y}_i)\vee F_1(\mathbf{b}_1^\top\mathbf{y}_j)\Big)\,W_2\Big(F_2(\mathbf{b}_2^\top\mathbf{y}_k)\vee F_1(\mathbf{b}_2^\top\mathbf{y}_l)\Big)\\ &-\frac{2}{n^3}\sum_i\sum_j\sum_k W_1\Big(F_1(\mathbf{b}_1^\top\mathbf{y}_i)\vee F_1(\mathbf{b}_1^\top\mathbf{y}_j)\Big)\,W_2\Big(F_2(\mathbf{b}_2^\top\mathbf{y}_i)\vee F_1(\mathbf{b}_2^\top\mathbf{y}_k)\Big).\end{aligned} \tag{48}$$

and under the true unmixing matrix,

$$\begin{aligned}\rho_w =& \frac{1}{n^2}\sum_i\sum_j W_1\Big(F_1(X_{1i})\vee F_1(X_{1j})\Big)\,W_2\Big(F_2(X_{2i})\vee F_2(X_{2j})\Big)\\ &+\frac{1}{n^4}\sum_i\sum_j\sum_k\sum_l W_1\Big(F_1(X_{1i})\vee F_1(X_{1j})\Big)\,W_2\Big(F_2(X_{2k})\vee F_2(X_{2l})\Big)\\ &-\frac{2}{n^3}\sum_i\sum_j\sum_k W_1\Big(F_1(X_{1i})\vee F_1(X_{1j})\Big)\,W_2\Big(F_2(X_{2i})\vee F_2(X_{2k})\Big).\end{aligned}$$

Regardless of the underlying distributions F_1 and F_2 of the components X_1 and X_2, the *probability integral transformation* (Casella & Berger, 2024) ensures that if $U = F_1(X_1),\ V = F_2(X_2)$, then U, V are identical and independent Uni$(0,1)$ random variables. Consequently, $F_i(X_{ij});\ i = 1,2; j = 1,2,...n$; are just observations from independent standard uniform random variables, and hence,

$$\begin{aligned}\rho_w =& \frac{1}{n^2}\sum_i\sum_j W_1\Big(U_i\vee U_j\Big)\,W_2\Big(V_i\vee V_j\Big)+\frac{1}{n^4}\sum_i\sum_j\sum_k\sum_l W_1\Big(U_i\vee U_j\Big)\,W_2\Big(V_k\vee V_l\Big)\\ &-\frac{2}{n^3}\sum_i\sum_j\sum_k W_1\Big(U_i\vee U_j\Big)\,W_2\Big(V_i\vee V_k\Big).\end{aligned} \tag{49}$$

The role of ρ_w is to provide an idea of how well a given candidate matrix $\mathbf{B}_\theta$ fits the observed data. Note that under the true $\mathbf{B}$ and our aforementioned special weight functions, the distribution of ρ_w can be simulated and 95% cut-off points can be computed. The (common) weight function used in this case is $w_i(s) = w(x) = \exp(-x^2)$ resulting in the simulated distribution of ρ_w with the special weights[8], as seen in Figure 2.1. Since this distribution is positive, right-tailed, and

[8]Note that the simulated distribution would depend on the exact form of the weights chosen.

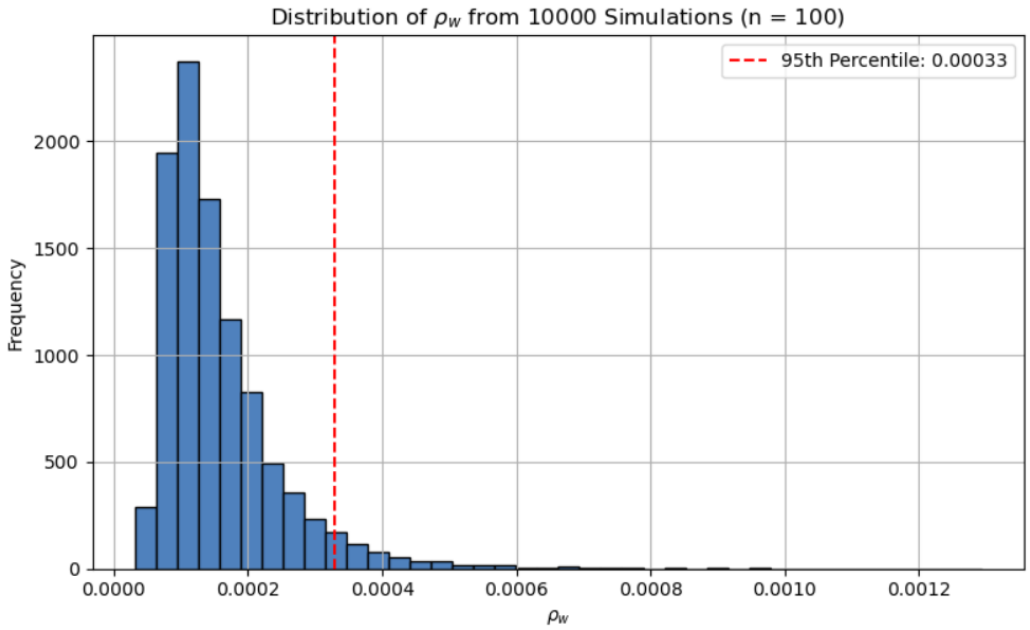

Figure 2.1: Histogram of ρ_w under true mixing matrix $\mathbf{B}$ ($n = 1000$). The 95^{th} percentile of the distribution is also labeled.

completely free from the observations $\mathbf{Y}$, the upper bound $\rho_{95} = 0.00033$ obtains for any dataset under these weight functions.

For the sample version, we use the empirical distribution $F_{ni}, i = 1, 2$, with the arguments swapped out for sample observations $\mathbf{Y}$. Once we generate the matrix $\mathbf{B}_\theta$ by a rotation of angle θ on the matrix $\bar{\mathbf{B}}$, we replace $\mathbf{b}_i^\top \mathbf{y}_j$ in (48) by $\mathbf{b}_{\theta i}^\top \mathbf{y}_j$, where $\mathbf{b}_{\theta i}$ are the i^{th} row of the candidate matrix $\mathbf{B}_\theta$, and $\mathbf{y}_j, j = 1, \ldots, n$, are the vector-valued synthetic observations. This yields the sample-variant $\hat{\rho}_w(\mathbf{B}_\theta, \mathbf{Y})$ given by:

$$\begin{aligned}\hat{\rho}_w(\mathbf{B}_\theta, \mathbf{Y}) =& \frac{1}{n^2} \sum_i \sum_j W_1\Big(F_{n1}(\mathbf{b}_{\theta 1}^\top \mathbf{y}_i) \vee F_{n1}(\mathbf{b}_{\theta 1}^\top \mathbf{y}_j)\Big)\, W_2\Big(F_{n2}(\mathbf{b}_{\theta 2}^\top \mathbf{y}_i) \vee F_{n2}(\mathbf{b}_{\theta 2}^\top \mathbf{y}_j)\Big) \\ &+ \frac{1}{n^4} \sum_i \sum_j \sum_k \sum_l W_1\Big(F_{n1}(\mathbf{b}_{\theta 1}^\top \mathbf{y}_i) \vee F_{n1}(\mathbf{b}_{\theta 1}^\top \mathbf{y}_j)\Big)\, W_2\Big(F_{n2}(\mathbf{b}_{\theta 2}^\top \mathbf{y}_k) \vee F_{n2}(\mathbf{b}_{\theta 2}^\top \mathbf{y}_l)\Big) \\ &- \frac{2}{n^3} \sum_i \sum_j \sum_k W_1\Big(F_{n1}(\mathbf{b}_{\theta 1}^\top \mathbf{y}_i) \vee F_{n1}(\mathbf{b}_{\theta 1}^\top \mathbf{y}_j)\Big)\, W_2\Big(F_{n2}(\mathbf{b}_{\theta 2}^\top \mathbf{y}_i) \vee F_{n2}(\mathbf{b}_{\theta 2}^\top \mathbf{y}_k)\Big).\end{aligned} \tag{50}$$

If a candidate $\mathbf{B}_\theta = \mathbf{C}_\theta \bar{\mathbf{B}}$ satisfies the condition $\hat{\rho}_w(\mathbf{B}_\theta, \mathbf{y}) \leq \rho_{95}$, then $\mathbf{B}_\theta \in \mathfrak{B}_\theta$, where

$\mathfrak{B}_\theta = \{\mathbf{B}_\theta = \mathbf{C}_\theta \bar{\mathbf{B}} \mid \mathbf{C} \text{ is orthogonal and } \hat{\rho}_w \leq \rho_{95}\}$. The algorithm to construct the confidence region is outlined in Algorithm 5 below. Since $\bar{\mathbf{B}}$ is fixed for a given dataset, all $\mathbf{B}_\theta \in \mathfrak{B}_\theta$ are characterized by the rotation angle θ.

Algorithm 5 Constructing Confidence Region for $\mathbf{B}$

1: **Input:** Observed data $\mathbf{y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$
2: Compute the sample dispersion matrix $\mathbf{S}_y$.
3: Perform EVD on $\mathbf{S}_y$ to obtain eigenpairs $(\lambda_i, \mathbf{p}_i)$ for $i = 1, 2$
4: Initialize:

$$\bar{\mathbf{B}} = \left[\frac{1}{\sqrt{\lambda_1}}\mathbf{p}_1 \quad \frac{1}{\sqrt{\lambda_2}}\mathbf{p}_2\right]$$

5: Generate candidate unmixing matrices:

$$\mathbf{B}_\theta = \mathbf{C}_\theta \bar{\mathbf{B}}, \quad \text{where } \mathbf{C}_\theta \text{ is a rotation matrix.}$$

6: **for all** $\theta \in [0, 2\pi]$ **do**
7: Compute $\hat{\rho}_w(\theta, \mathbf{y})$
8: **if** $\rho_{\theta w}(\mathbf{B}_\theta, \mathbf{y}) \leq \rho_{95}$ **then**
9: Include $\mathbf{B}_\theta$ in confidence region $\mathfrak{B}_\theta$
10: **end if**
11: **end for**
12: **Output:** Confidence region $\mathfrak{B}_\theta$ for $\mathbf{B}$

2.4.3 Simulated Examples

We illustrate how this procedure works with a couple of examples. The procedure is the same in both, though the sources $\mathbf{X}$ and the mixing matrix $\mathbf{A}$ differ. Using these we generate the observations $\mathbf{Y}$.

Example 1

In this comprehensive simulation study, we compare the confidence regions across four different setups that vary under two conditions:

(1) whether the sources are both Gaussian or not; and

(2) whether the mixing matrix $\mathbf{A}$ is the identity matrix or not.

The sample size is set to $n = 100$. The four cases are defined as follows:

- Case 1:

  X_1 is sampled from a Laplace distribution with mean 0 and a scale parameter of $\frac{1}{\sqrt{2}}$, and X_2 is sampled from a continuous uniform distribution defined over the interval $(-\sqrt{3}, \sqrt{3})$. More details about this particular choice for the sources are discussed in Chapter 4.

  The mixing matrix used to generate the synthetic observations is

$$\mathbf{A} = \begin{bmatrix} 8 & -12 \\ -2 & 15 \end{bmatrix},$$

- Case 2:

  Same sources as Case 1, but the mixing matrix is replaced by the identity matrix. In this case $\mathbf{Y} = \mathbf{X}$, which implies that the observations are already independent.

- Case 3:

  Both X_1 and X_2 are sampled from independent standard Gaussian distributions, and the mixing matrix is the same as Case 1.

- Case 4:

  The sources, like Case 3, are independent standard Gaussian, and the mixing matrix is replaced by the identity matrix (like Case 2).

It is expected that under Cases 2 and 4 setups, nearly all angles will be accepted in their respective confidence regions.

For cases where $\mathbf{A} \neq \mathbf{I}_2$, the unmixing matrix is given by

$$\mathbf{B} = \mathbf{A}^{-1} = \begin{bmatrix} 0.15625 & 0.12500 \\ 0.02083 & 0.08333 \end{bmatrix}.$$

To construct the set, we start with an initial choice of the unmixing matrix $\bar{\mathbf{B}}$ using EVD of the

sample dispersion matrix:

$$\bar{\mathbf{B}} = \begin{bmatrix} -0.1554 & -0.0336 \\ -0.1501 & 0.0347 \end{bmatrix} \tag{51}$$

We then define a family of candidate unmixing matrices $\mathbf{B}_\theta = \mathbf{C}_\theta \bar{\mathbf{B}}$, where $\mathbf{C}_\theta$ is the rotation matrix characterized by the angle $\theta \in [0, 2\pi]$. For $n = 100$, the 96^{th} percentile of the null distribution of $\hat{\rho}_w$ under the true unmixing matrix is simulated to be $\rho_{95} = 0.00033$. For each candidate, we compute the statistic $\hat{\rho}_w(\mathbf{B}_\theta, \mathbf{Y})$ and include $\mathbf{B}_\theta$ in the confidence region if

$$\hat{\rho}_w(\mathbf{B}_\theta, \mathbf{Y}) \leq \rho_{95}.$$

The results of the simulation are visualized in Figue 2.2, tabulated in Table 2.1 and detailed below:

Cases 1 and 3 reflect the standard ICA setup. Accepted regions are narrow and spaced (approximately) 90° apart, consistent with the $\frac{\pi}{2}$-periodicity discussed before. Cases 2 and 4 are the setups where the true sources are independent (as we replace $\mathbf{A} = \mathbf{I}_2$). The method successfully identifies very wide acceptance intervals, or full acceptance as in Case 4, which validates the approach. Under independence and Gaussian sources, every rotation results in a "valid" unmixing matrix due to rotational invariance of the joint distribution of the Gaussian densities (Cf. Section 1.3.3). The vertical dips in Figure 2.2 correspond to values of θ where $\hat{\rho}_w$ falls below the 95^{th} percentile. The $\frac{\pi}{2}$-periodicity is visible in both the non-degenerate cases. Correspondingly, the confidence region for $\mathbf{B}$ is the set $\mathcal{B}_\theta = \{\mathbf{B}_\theta = \mathbf{C}_\theta \bar{\mathbf{B}} \mid \theta \in \Theta, \mathbf{C} \text{ is orthogonal}\}$ where $\bar{\mathbf{B}}$ is as in (51).

In particular for Case 1, Θ is divided into 4 intervals, 90° apart. The small deviations from exact 90° separations are attributable to random fluctuations in the simulation and finite sample variability. For the first partition $\theta \in [32.5^\circ, 34^\circ]$, we have the matrices ranging (approximately)

$$\text{from } \mathbf{B}_{32.5^\circ} = \begin{bmatrix} -0.0504 & -0.0470 \\ -0.2101 & 0.0112 \end{bmatrix} \text{ to } \mathbf{B}_{34^\circ} = \begin{bmatrix} -0.0449 & -0.0473 \\ -0.2113 & 0.0100 \end{bmatrix},$$

with the entries varying continuously based on the angle θ. Specifically, the entries of the matrix $\mathbf{B}_\theta$ are trigonometric combinations of those of the matrix $\bar{\mathbf{B}}$, and so, are sinusoidal functions of θ

(graphed below in Figure 2.3).

Case	Mixing Matrix	Accepted Intervals for θ (in degrees)
1	$\mathbf{A}$	$[32.5, 34]$, $[122.5, 124]$, $[212.5, 213.5]$, $[302.5, 304]$
2	$\mathbf{I}_2$	$[0, 2.5]$, $[40.5, 80.5]$, $[131.5, 174.5]$, $[220.5, 275.5]$, $[308.5, 360]$
3	$\mathbf{A}$	$[33, 35]$, $[123, 125]$, $[213, 215]$, $[303, 305]$
4	$\mathbf{I}_2$	**All angles accepted**

Table 2.1: Accepted angle intervals (in degrees) for which $\hat{\rho}_w(\theta) \leq \rho_{95} = 0.00033$ under each case. Sources in Cases 1 and 2: Laplace and Uniform. Sources in Cases 3 and 4: Both Gaussian.

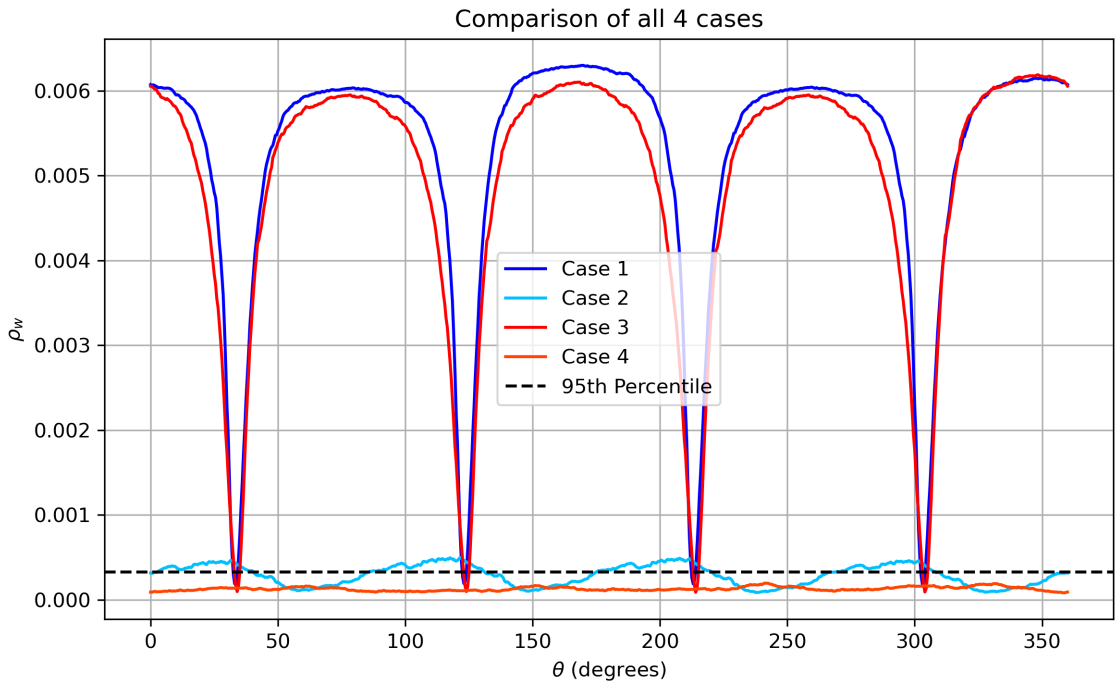

Figure 2.2: Comparison of $\hat{\rho}_w$ across Cases 1–4 over θ. The horizontal dashed line indicates the 95^{th} percentile of the null distribution. Note the $\frac{\pi}{2}$-periodicity exhibited by the function for Cases 1 and 3.

Example 2

In this example, we proceed in the same way to construct the confidence regions across four different setups. The sample size is set to $n = 100$. The four cases are defined as follows:

- Case 1:

  Source X_1 is sampled from standard Gaussian, while Source X_2 is sampled from a uniform

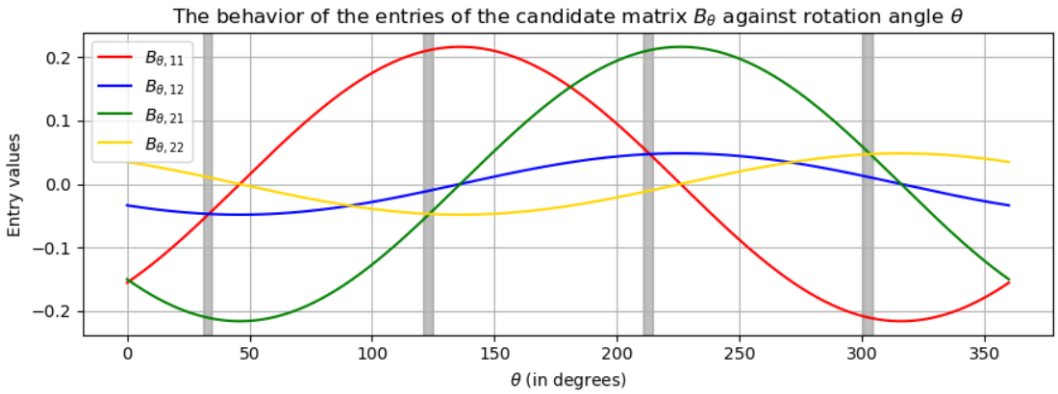

Figure 2.3: The plots shows the change in the values of the entries of $\mathbf{B}_\theta$ against θ. The shaded regions represent the angles θ corresponding the the confidence region $\mathcal{B}_\theta$ for Example 1.

distribution defined over the interval $(-\sqrt{3}, \sqrt{3})$. The mixing matrix used to generate the synthetic observations is

$$\mathbf{A} = \begin{bmatrix} 3 & -4 \\ -2 & 7 \end{bmatrix},$$

- Case 2:

  Same sources as Case 1, but the mixing matrix is replaced by the identity matrix. In this case $\mathbf{Y} = \mathbf{X}$, which implies that the observations are already independent.

- Case 3:

  Both X_1 and X_2 are sampled from independent standard Gaussian distributions, and the mixing matrix is the same as Case 1.

- Case 4:

  The sources, like Case 3, are independent standard Gaussian, and the mixing matrix is replaced by the identity matrix (like Case 2).

Again, it is expected that under Cases 2 and 4 setups, most, if not all, angles will be accepted in their confidence regions.

For cases where $\mathbf{A} \neq \mathbf{I}_2$, the unmixing matrix is given by

$$\mathbf{B} = \mathbf{A}^{-1} = \begin{bmatrix} 0.5385 & 0.1538 \\ 0.3077 & 0.2308 \end{bmatrix}.$$

The initial choice $\bar{\mathbf{B}} = \begin{bmatrix} -0.5555 & -0.0640 \\ -0.3714 & 0.0957 \end{bmatrix}$, and the statistic $\hat{\rho}_w$ is computed in each case and plotted (Figure 2.4). The accepted angles in each case is reported in Table 2.2. The intervals obtained are 90° apart, and for Case 1, we have:

$$\theta \in [41.5^\circ, 43.5^\circ] \cup [131.5^\circ, 133.5^\circ] \cup [222^\circ, 223.5^\circ] \cup [312^\circ, 313.5^\circ] =: \Theta.$$

and the confidence region for $\mathbf{B}$ is the set $\mathcal{B}_\theta = \{\mathbf{B}_\theta = \mathbf{C}_\theta \bar{\mathbf{B}} \mid \theta \in \Theta, \mathbf{C} \text{ is orthogonal}\}$.

Below, we show the matrices corresponding to the terminals of the first interval of $\mathcal{B}_\theta$ where $\theta \in [41.5^\circ, 43.5^\circ]$ given by

$$\mathbf{B}_{41.5^\circ} = \begin{bmatrix} -0.1699 & -0.1113 \\ -0.6462 & 0.0293 \end{bmatrix} \text{ to } \mathbf{B}_{43.5^\circ} = \begin{bmatrix} -0.1473 & -0.1123 \\ -0.6518 & 0.0254 \end{bmatrix},$$

with the entries varying continuously with change in θ, as shown in Figure 2.5.

Case	Mixing Matrix	Accepted Intervals for θ (in degrees)
1	$\mathbf{A}$	$[41.5, 43.5]$, $[131.5, 133.5]$, $[222, 223.5]$, $[312, 313.5]$
2	$\mathbf{I}_2$	**All angles accepted**
3	$\mathbf{A}$	$[42, 44]$, $[132, 133.5]$, $[222, 224]$, $[312, 314]$
4	$\mathbf{I}_2$	**All angles accepted**

Table 2.2: Accepted angle intervals (in degrees) for which $\hat{\rho}_w(\theta) \leq \rho_{95} = 0.00033$ under each case. Sources in Cases 1 and 2: Gaussian and Uniform. Sources in Cases 3 and 4: Both Gaussian.

As we will see in the following chapter, particularly in Section 3.2.4, a similar structure emerges when analyzing the principal components of the empirical process. The angles identified through this alternative approach closely match those found here, reinforcing the robustness of the confidence region for $\mathbf{B}$ constructed using this method.

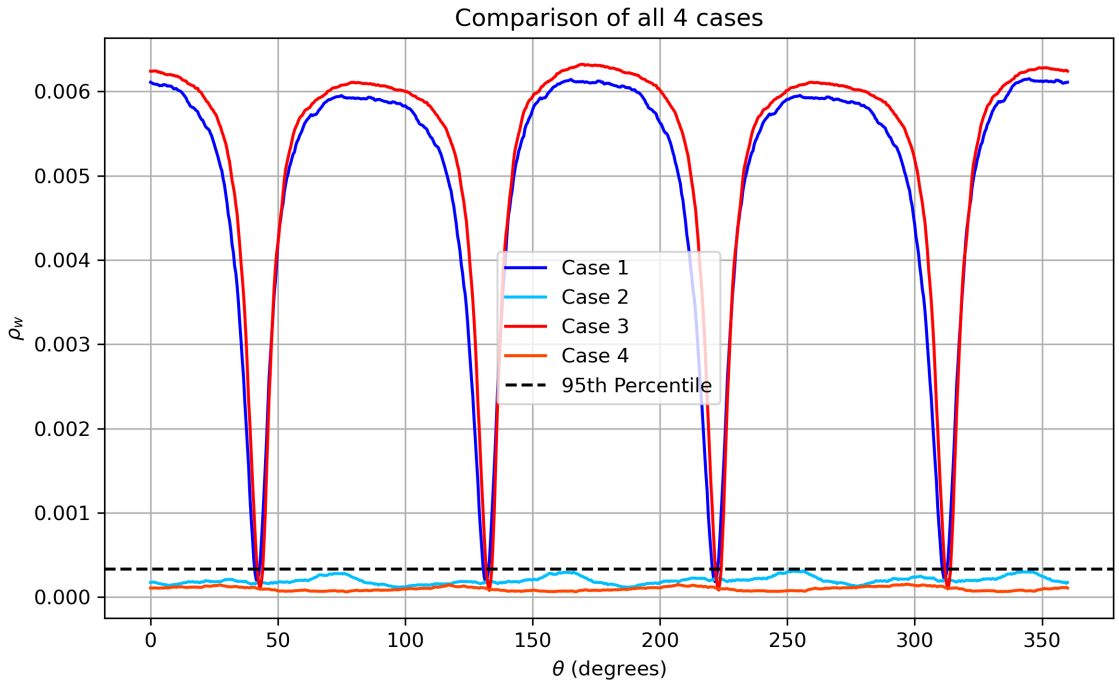

Figure 2.4: Comparison of $\hat{\rho}_w$ across Cases 1–4 over θ. The horizontal dashed line indicates the 95^{th} percentile of the null distribution. Note the $\frac{\pi}{2}$-periodicity exhibited by the function for Cases 1 and 3.

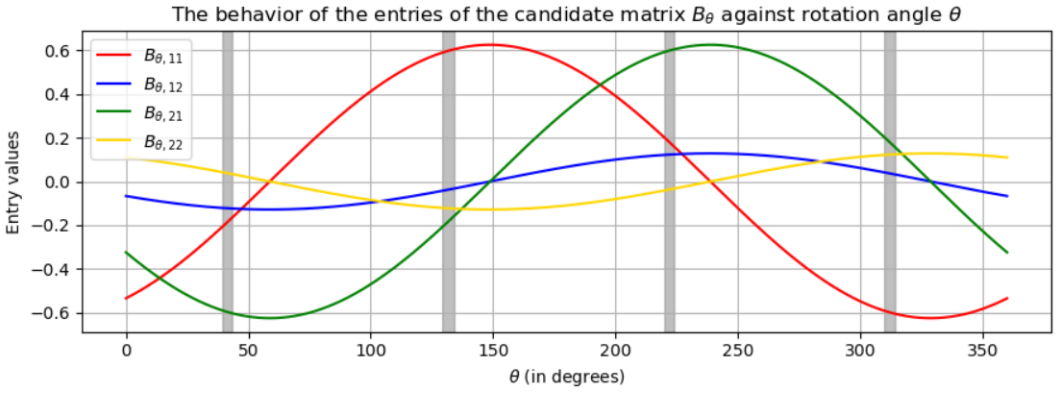

Figure 2.5: The plots shows the change in the values of the entries of $\mathbf{B}_\theta$ against θ. The shaded regions represent the angles θ corresponding the the confidence region $\mathcal{B}_\theta$ for Example 2.

Chapter 3

Asymptotics of the Empirical Process and Minimum Distance Estimator

Suppose $\mathbf{X} = (X_1, \ldots, X_d)$ is a d-variate random vector with joint distribution function $F(\mathbf{x}) = F(x_1, \ldots, x_d)$ and let the marginal distribution functions of X_j be $F_j(x_j);\ j = 1, 2, \ldots, d$. Let $\mathbf{X}_i = (X_{1i}, \ldots, X_{di})$ be identically and independently distributed observations from $\mathbf{X}$. Based on n observations, define the empirical joint and marginal distribution functions as follows:

$$\hat{F}(\mathbf{x}) \coloneqq \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{\mathbf{X}_i \leq \mathbf{x}\};\ \text{and}\ \hat{F}_j(x_j) \coloneqq \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{X_{ji} \leq x_j\}; j = 1, 2, \ldots, d. \tag{52}$$

Note: Here, we use a slightly modified notation for the empirical distribution function (compared to previous chapters), $\hat{F}$ instead of F_n, as the n in the subscript will be used to signify the relation between the sample size n and the empirical distribution function under reparameterization (which will be denoted by $\hat{F}_n$ later).

In the subsequent sections of this chapter, we will look at the asymptotics of the difference between the joint and product of marginal empirical distributions. Using a standard reparameterization of the parameter of interest, we will derive the limiting distribution of a minimizer estimate of the unmixing matrix $\mathbf{B}$.

3.1 Limit of the Empirical Process

We begin by stating and proving a general asymptotic result on the distribution of the *distance* defined by the empirical distribution functions (52).

Lemma 3.1.1. *Suppose the d random variables $X_1, X_2, \ldots, X_d$ are independent. Then for the empirical distribution functions as defined in (52), we have*

$$\sqrt{n}\left[\hat{F}(\mathbf{x}) - \prod_{j=1}^{d} \hat{F}_j(x_j)\right] \xrightarrow{\mathtt{d}} \mathcal{G}(\mathbf{x}), \text{ as } n \longrightarrow \infty, \tag{53}$$

where $\mathcal{G}(\mathbf{x})$[1] *has the covariance given by*

$$\begin{aligned}&\mathbb{C}ov(\mathcal{G}(\mathbf{x}), \mathcal{G}(\mathbf{y}))\\ &= F(\mathbf{x} \wedge \mathbf{y}) - F(\mathbf{x})F(\mathbf{y}) - \sum_{k=1}^{d} \pi(F_k(x_k))\pi(F_k(y_k))\Big[F_k(x_k \wedge y_k) - F_k(x_k)F_k(y_k)\Big].\end{aligned} \tag{54}$$

Proof. Start off by expanding the left side of (53) as follows:

$$\sqrt{n}\left[\hat{F}(\mathbf{x}) - \prod_{j=1}^{d} F_j(x_j) - \prod_{j=1}^{d} \hat{F}_j(x_j) + \prod_{j=1}^{d} F_j(x_j)\right]. \tag{55}$$

Note that

$$\begin{aligned}\prod_{j=1}^{d} \hat{F}_j(x_j) &= \prod_{j=1}^{d}\Big[\hat{F}_j(x_j) - F_j(x_j) + F_j(x_j)\Big]\\ &= \prod_{j=1}^{d} F_j(x_j) + \sum_{k=1}^{d}\Big(\hat{F}_k(x_k) - F_k(x_k)\Big)\prod_{\substack{j=1\\ j\neq k}}^{d} F_j(x_j) + \text{extra},\end{aligned}$$

the "extra" being terms containing a product of at least two sub-terms of the form $\hat{F}_j(x_j) - F_j(x_j)$.

[1] $\mathcal{G}(\mathbf{x})$ itself will be some function of Gaussian processes, though its exact nature is difficult to discern. However, as we show later that in the bivariate case, the left side of (53) would converge in distribution to a product of independent standard Brownian Bridges (a function of two Gaussian processes) and simpler to deal with.

Going forward, we will be using the following notation:

$$\pi(F_k(x_k)) \coloneqq \prod_{\substack{j=1 \\ j\neq k}}^{d} F_j(x_j), \text{ and so we can write,}$$

$$\sqrt{n}\left[\prod_{j=1}^{d}\hat{F}_j(x_j) - \prod_{j=1}^{d}F_j(x_j)\right] = \sqrt{n}\left[\sum_{k=1}^{d}\left[\left\{\frac{1}{n}\sum_i \mathbb{1}\{X_{ki}\leq x_k\} - F_k(x_k)\right\}\right]\pi(F_k(x_k)) + \text{extra}\right],$$

and thus, (55) can be expressed as:

$$\sqrt{n}\left[\frac{1}{n}\sum_i \mathbb{1}\{\mathbf{X}_i\leq \mathbf{x}\} - \prod_{j=1}^{d}F_j(x_j) - \sum_{k=1}^{d}\left[\left\{\frac{1}{n}\sum_i \mathbb{1}\{X_{ki}\leq x_k\} - F_k(x_k)\right\}\right]\pi(F_k(x_k)) - \text{extra}\right]$$

Here, as $n \longrightarrow \infty$, the term $\sqrt{n}\,[\text{extra}] = \mathcal{O}_p\left(\frac{1}{\sqrt{n}}\right)$, and the entire expression converges, in distribution, to some function of Gaussian Processes, say $\mathcal{G}(\mathbf{x}) \equiv \mathcal{G}(x_1, \ldots, x_d)$. The covariance of $\mathcal{G}$, after removing the constant terms and scalar multipliers, is given by

$$\begin{aligned}\mathbb{C}\text{ov}\Big(\mathcal{G}(\mathbf{x}), \mathcal{G}(\mathbf{y})\Big) =& \frac{1}{n}\mathbb{C}\text{ov}\Bigg(\sum_i \mathbb{1}\{\mathbf{X}_i\leq \mathbf{x}_i\} - \sum_{k=1}^{d}\Bigg[\sum_i \mathbb{1}\{X_{ki}\leq x_k\}\Bigg]\pi(F_k(x_k)),\\ &\sum_i \mathbb{1}\{\mathbf{X}_i\leq \mathbf{y}_i\} - \sum_{k=1}^{d}\Bigg[\sum_i \mathbb{1}\{X_{ki}\leq y_k\}\Bigg]\pi(F_k(y_k))\Bigg)\\ =& \frac{1}{n}\mathbb{C}\text{ov}\Big(C_1(\mathbf{x}) - C_2(\mathbf{x}), C_1(\mathbf{y}) - C_2(\mathbf{y})\Big), \end{aligned} \tag{56}$$

where $C_1(\mathbf{x}) = \sum_i \mathbb{1}\{\mathbf{X}_i\leq \mathbf{x}\}$ and $C_2(\mathbf{x}) = \sum_{k=1}^{d}\Bigg[\sum_i \mathbb{1}\{X_{ki}\leq x_k\}\Bigg]\pi(F_k(x_k))$.

The simplification of (56) to get a closed form is divided into four parts, labeled (i)-(iv) below.

For part (i),

$$\begin{aligned}
\mathbb{C}\text{ov}\left(C_1(\mathbf{x}), C_1(\mathbf{y})\right) =& \mathbb{C}\text{ov}\left(\sum_i \mathbb{1}\{\mathbf{X}_i \leq \mathbf{x}\}, \sum_i \mathbb{1}\{\mathbf{X}_i \leq \mathbf{y}\}\right) \\
=& \sum_{i=1}^{n} \mathbb{C}\text{ov}\Big(\mathbb{1}\{\mathbf{X}_i \leq \mathbf{x}\}, \mathbb{1}\{\mathbf{X}_i \leq \mathbf{y}\}\Big), \text{ since } \mathbf{X}_i \text{ are independent} \\
=& n \cdot \mathbb{C}\text{ov}\Big(\mathbb{1}\{\mathbf{X}_1 \leq \mathbf{x}\}, \mathbb{1}\{\mathbf{X}_1 \leq \mathbf{y}\}\Big), \text{ since } \mathbf{X}_i \text{ are identical} \\
=& n\left[\mathbb{E}\Big[\mathbb{1}\{\mathbf{X}_1 \leq \mathbf{x}\} \cdot \mathbb{1}\{\mathbf{X}_1 \leq \mathbf{y}\}\Big] - \mathbb{E}\Big[\mathbb{1}\{\mathbf{X}_1 \leq \mathbf{x}\}\Big]\mathbb{E}\Big[\mathbb{1}\{\mathbf{X}_1 \leq \mathbf{y}\}\Big]\right] \\
=& n\Big[F(\mathbf{x} \wedge \mathbf{y}) - F(\mathbf{x})F(\mathbf{y})\Big], \qquad (56.1)
\end{aligned}$$

where $F(\mathbf{x} \wedge \mathbf{y}) \coloneqq F(x_1 \wedge y_1, x_2 \wedge y_2, \ldots, x_d \wedge y_d)$, F being the joint distribution function of $\mathbf{X}$. For part (ii), due to independence of $X_k; k = 1, \ldots, d$, we can shift the sum over k in $C_2(\mathbf{x})$ outside

$$\begin{aligned}
\mathbb{C}\text{ov}\left(C_2(\mathbf{x}), C_1(\mathbf{y})\right) =& \mathbb{C}\text{ov}\left(\sum_{k=1}^{d}\Big[\sum_i \mathbb{1}\{X_{ki} \leq x_k\}\Big]\pi(F_k(x_k)), \sum_i \mathbb{1}\{\mathbf{X}_i \leq \mathbf{y}\})\right) \\
=& \sum_{k=1}^{d} \pi(F_k(x_k))\ \mathbb{C}\text{ov}\left(\sum_i \mathbb{1}\{X_{ki} \leq x_k\}, \sum_i \mathbb{1}\{\mathbf{X}_i \leq \mathbf{y}\})\right).
\end{aligned}$$

Now, consider the m^{th} term, i.e., $k = m,\ 1 \leq m \leq d$,

$$\begin{aligned}
\mathbb{C}\text{ov}\left(\sum_i \mathbb{1}\{X_{mi} \leq x_m\}, \sum_i \mathbb{1}\{\mathbf{X}_i \leq \mathbf{y}\})\right) =& \sum_i \mathbb{C}\text{ov}\Big(\mathbb{1}\{X_{mi} \leq x_m\}, \mathbb{1}\{\mathbf{X}_i \leq \mathbf{y}\}\Big) \\
=& n\Big[F(\mathbf{y}; x_m) - F_m(x_m)F(\mathbf{y})\Big],
\end{aligned}$$

where $F(\mathbf{y}; x_m) \coloneqq F(y_1, y_2, \ldots, y_{m-1}, y_m \wedge x_m, y_{m+1}, \ldots, y_d)$. Therefore,

$$\begin{aligned}
\mathbb{C}\text{ov}\left(C_2(\mathbf{x}), C_1(\mathbf{y})\right) =& n\sum_{k=1}^{d} \pi(F_k(x_k))\Big[F(\mathbf{y}; x_k) - F_k(x_k)F(\mathbf{y})\Big] \\
=& n\sum_{k=1}^{d} \pi(F_k(x_k))\pi(F_k(y_k))\Big[F_k(x_k \wedge y_k) - F_k(x_k)F_k(y_k)\Big]. \qquad (56.2)
\end{aligned}$$

By identical arguments, part (iii)

$$\mathbb{C}\text{ov}\left(C_1(\mathbf{x}), C_2(\mathbf{y})\right) = n\sum_{k=1}^{d}\pi(F_k(y_k))\pi(F_k(x_k))\Big[F_k(y_k \wedge x_k) - F_k(y_k)F_k(x_k)\Big]. \tag{56.3}$$

Lastly, part (iv) is

$$\begin{aligned}
\mathbb{C}\text{ov}\left(C_2(\mathbf{x}), C_2(\mathbf{y})\right) =& \mathbb{C}\text{ov}\left(\sum_{k=1}^{d}\Big[\sum_i \mathbb{1}\{X_{ki} \le x_k\}\Big]\pi(F_k(x_k)), \sum_{k=1}^{d}\Big[\sum_i \mathbb{1}\{X_{ki} \le y_k\}\Big]\pi(F_k(y_k))\right) \\
=& \sum_{k=1}^{d}\pi(F_k(x_k))\pi(F_k(y_k))\ \mathbb{C}\text{ov}\Big(\sum_i \mathbb{1}\{X_{ki} \le x_k\}, \sum_i \mathbb{1}\{X_{ki} \le y_k\}\Big) \\
=& \sum_{k=1}^{d}\pi(F_k(x_k))\pi(F_k(y_k))\Big[n \cdot \mathbb{C}\text{ov}\Big(\mathbb{1}\{X_{k1} \le x_k\}, \mathbb{1}\{X_{k1} \le y_k\}\Big)\Big] \\
=& n\sum_{k=1}^{d}\pi(F_k(x_k))\pi(F_k(y_k))\Big[F_k(x_k \wedge y_k) - F_k(x_k)F_k(y_k).\Big]
\end{aligned} \tag{56.4}$$

Combining parts (i)-(iv) (relations (56.1) to (56.4)), the covariance of the process is obtained as

$$\begin{aligned}
&\mathbb{C}\text{ov}(\mathcal{G}(\mathbf{x}), \mathcal{G}(\mathbf{y})) \\
&= F(\mathbf{x} \wedge \mathbf{y}) - F(\mathbf{x})F(\mathbf{y}) - \sum_{k=1}^{d}\pi(F_k(x_k))\pi(F_k(y_k))\Big[F_k(x_k \wedge y_k) - F_k(x_k)F_k(y_k)\Big].
\end{aligned}$$

■

Unfortunately, there is not a lot that can be achieved using this rather complicated covariance form, as it does not lead to a tractable Gaussian process, except for the 2-D scenario. As shown in the next corollary, the 2-D case yields an interesting form for the above process $\mathcal{G}(x)$, due to how manageable the covariance structure becomes.

Corollary 3.1.2. *In the two-dimensional case ($d = 2$) $\mathcal{G}(\cdot)$ on the right side of (53), is the product of two independent standard Brownian Bridges $\mathcal{B}_1$ and $\mathcal{B}_2$.*

Proof. When $d = 2$, the covariance in (54) simplifies greatly to

$$
\begin{aligned}
&F(x_1 \wedge y_1, x_2 \wedge y_2) - F(x_1, x_2)F(y_1, y_2) - F_2(x_2)F_2(y_2)\Big[F_1(x_1 \wedge y_1) - F_1(x_1)F_1(y_1)\Big] \\
&- F_1(x_1)F_1(y_1)\Big[F_2(x_2 \wedge y_2) - F_2(x_2)F_2(y_2)\Big] \\
&= \Big[F_1(x_1 \wedge y_1) - F_1(x_1)F_1(y_1)\Big]\Big[F_2(x_2 \wedge y_2) - F_2(x_2)F_2(y_2)\Big],
\end{aligned}
$$

which is just a product of the covariances of two standard Brownian Bridges. Therefore, from (53) we have

$$
\sqrt{n}\Big[\hat{F}(\mathbf{x}) - \hat{F}_1(x_1)\hat{F}_2(x_2)\Big] \xrightarrow{\mathtt{d}} \mathcal{B}_1(F_1(x_1)) \cdot \mathcal{B}_2(F_2(x_2)). \tag{57}
$$

■

Now we can finally divert our attention to its application in our Independent Components problem, specifically in the 2-D scenario, and the distance involved in ρ_w. For the ICA setup as stated in (3) and (6), define the rows of the matrix $\mathbf{B}$ as $\mathbf{b}_1^\top$ and $\mathbf{b}_2^\top$. So, for i^{th} observation, we have the following relation between the independent components, the rows of the unmixing matrix and the observations

$$
\mathbf{X}_i = \mathbf{B}\mathbf{Y}_i \implies \begin{bmatrix} X_{1i} \\ X_{2i} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1^\top \mathbf{Y}_i \\ \mathbf{b}_2^\top \mathbf{Y}_i \end{bmatrix}, \; i = 1, 2, \ldots, n. \tag{58}
$$

Result 3.1.3. *Under the knowledge of the true unmixing matrix, the difference* $F_n(\mathbf{y} \mid \mathbf{B}) - F_n^{\perp}(\mathbf{y} \mid \mathbf{B})$*, where* $F_n(\mathbf{y} \mid \mathbf{B}), F_n^{\perp}(\mathbf{y} \mid \mathbf{B})$ *are as defined in* (18) *and* (19)*, is asymptotically a product of standard Brownian Bridges* $\mathcal{B}_1()$ *and* $\mathcal{B}_2()$*, i.e.,*

$$
\sqrt{n}\Big[F_n(\mathbf{y} \mid \mathbf{B}) - F_n^{\perp}(\mathbf{y} \mid \mathbf{B})\Big] \xrightarrow{\mathtt{d}} \mathcal{B}_1(F_1(y_1)) \cdot \mathcal{B}_2(F_2(y_2)). \tag{59}
$$

Proof. The difference can be rewritten in an expanded form using (58) as follows:

$$
\begin{aligned}
&F_n(\mathbf{y} \mid \mathbf{B}) - F_n^{\perp}(\mathbf{y} \mid \mathbf{B}) \\
=&\frac{1}{n}\sum_i \mathbb{1}\left\{\mathbf{b}_1^\top \mathbf{Y}_i \le y_1,\ \mathbf{b}_2^\top \mathbf{Y}_i \le y_2\right\} - \frac{1}{n^2}\left[\sum_i \mathbb{1}\left\{\mathbf{b}_1^\top \mathbf{Y}_i \le y_1\right\}\right]\left[\sum_i \mathbb{1}\left\{\mathbf{b}_2^\top \mathbf{Y}_i \le y_2\right\}\right] \\
=&\frac{1}{n}\sum_i \mathbb{1}\left\{X_{1i} \le y_1,\ X_{2i} \le y_2\right\} - \frac{1}{n^2}\left[\sum_i \mathbb{1}\left\{X_{1i} \le y_1\right\}\right]\left[\sum_i \mathbb{1}\left\{X_{2i} \le y_2\right\}\right] \\
=&\hat{F}(\mathbf{y}) - \hat{F}_1(y_1)\hat{F}_2(y_2),
\end{aligned}
$$

where $\hat{F}, \hat{F}_1, \hat{F}_2$ are the joint and marginal empirical distribution functions (52) of $\mathbf{X}, X_1,$ and X_2, respectively. The RVs X_1 and X_2 are independent. If F is the joint distribution function of $\mathbf{X}$ and the corresponding marginal distribution functions of X_1 and X_2 are F_1 and F_2, then by Corollary 3.1.2,

$$
\sqrt{n}\left[F_n(\mathbf{y} \mid \mathbf{B}) - F_n^{\perp}(\mathbf{y} \mid \mathbf{B})\right] \xrightarrow{\mathtt{d}} \mathcal{B}_1(F_1(y_1)) \cdot \mathcal{B}_2(F_2(y_2)).
$$

■

3.2 Analysis of the Principal Components of the Empirical Process

A good portion of Chapter 5 in Shorack and Wellner (2009) is dedicated to the Principal Component Decomposition of processes and the derivation of corresponding eigenfunctions. The motivation behind the idea is simple, and it forms the backbone of data analysis and dimension reduction in Principal Component Analysis. However, the concept can be extended further, from the covariance matrix to the covariance function, as discussed below.

It is well known that an $n \times n$ covariance matrix Σ can be represented as countable (finite) series

$$
\Sigma = \sum_{i=1}^{n} \lambda_j \mathbf{g}_j \mathbf{g}_j^\top
$$

where λ_j are eigenvalues and $\mathbf{g}_j$ are orthonormal eigenvectors of the matrix Σ. If $Z_1, Z_2, \ldots$ are

independent and identically distributed standard Gaussian random variables, then the series

$$\sum_{j=1}^{n} \sqrt{\lambda_j} Z_j \mathbf{g}_j$$

is a $\mathtt{N}_n(\mathbf{0}, \Sigma)$, a multivariate Gaussian random variable with mean vector $\mathbf{0}$ and covariance matrix Σ.

Similarly, in Shorack and Wellner (2009) argue and show, the covariance function $K(s,t)$ of (many) processes can be represented as a countable (infinite) series

$$K(s,t) = \sum_{j=1}^{\infty} \lambda_j f_j(s) f_j(t) \tag{60}$$

for orthonormal functions $f_j \in \mathfrak{L}_2$. Thus, the process defined as

$$\mathcal{G}(t) = \sum_{j=1}^{\infty} \sqrt{\lambda_j} Z_j f_j(t) \tag{61}$$

is a Gaussian process with mean value function 0 and covariance function K. This section follows Anderson and Darling (1952) and Shorack and Wellner (2009) to determine the principal components of our process. Then we use the Principal Components construct a confidence region for the unmixing matrix $\mathbf{B}$.

3.2.1 Set of principal components

Suppose that the vector-valued random variables $\mathbf{x}_i = (x_{1i}, x_{2i}), i = 1, ..., n$ are drawn from identical and independent distributions. Let $F_n(\mathbf{x}),\ \mathbf{x} = (x_1, x_2)$; be the joint empirical distribution function, $F_{nj}(x_j),\ j = 1, 2$; be the independent marginal empirical distribution functions, and define the process

$$\mathcal{Y}_n(\mathbf{x}) = \sqrt{n}\Big[F_n(\mathbf{x}) - F_{n1}(x_1)F_{n2}(x_2)\Big]. \tag{62}$$

It has already been shown in Section 3.1 that

$$\mathcal{Y}_n(\mathbf{x}) \xrightarrow{\mathtt{d}} \mathcal{Y}(\mathbf{x}) = \mathcal{B}_1(F_1(x_1)) \cdot \mathcal{B}_2(F_2(x_2)),$$

where $\mathcal{B}_1$ and $\mathcal{B}_2$ are independent standard Brownian bridges. From the properties of Brownian bridges, we gather

$$\mathbb{E}(\mathcal{Y}(\mathbf{x})) = \mathbf{0}, \text{and}$$

$$\mathbb{C}\text{ov}(\mathcal{Y}(\mathbf{x}), \mathcal{Y}(\mathbf{x}')) = \prod_{j=1}^{2} K[F_j(x_j), F_j(x'_j)],$$

where $K(u, u') = u \wedge u' - uu'$.

The goal here is to obtain a sequence of uncorrelated random variables $\Gamma_n(i, j), i, j = 1, 2, ...;$ with mean 0 and unit variance, such that

$$\mathcal{Y}_n(\mathbf{x}) = \sum_i \sum_j \Gamma_n(i, j)\Psi_0(i, j)(\mathbf{x}), \tag{63}$$

where $\Gamma_n(i, j)$ are the unit variance *principal components* and $\Psi_0(i, j)(\mathbf{x})$ are the orthogonal basis functions.

The principal components of a vector $\mathbf{a} = (a_1, a_2, ..., a_N)$ with mean vector $\mathbf{0}$ and dispersion matrix Σ are given by $z_j = \mathbf{l}_j^\top \mathbf{a},\ j = 1, 2, ..., N$; where $\mathbf{l}_j$, are the normalized eigenvectors of Σ corresponding to the eigenvalues[2] $\lambda_1, ..., \lambda_N$, are obtainable by solving the following eigen-equation for $\lambda, \mathbf{l}$:

$$\Sigma\mathbf{l} = \lambda\mathbf{l} \tag{64}$$

Dividing z_j by their standard deviations $\sqrt{\lambda_j}$, the uncorrelated components, with zero mean and unit variance, are obtained:

$$z_j^* = \frac{1}{\sqrt{\lambda_j}} \mathbf{l}_j^\top \mathbf{a},\ j = 1, 2, ..., N. \tag{65}$$

[2] Assumed to be distinct and positive.

Similarly, to derive the principal components of $\mathcal{Y}_n(\mathbf{x})$ we can start with the eigen-equation:

$$\int_{\mathbb{R}^2}\left[\prod_{k=1}^{2} K(F_j(x_k), F_j(x_k'))\right]\Psi_{1i}(F_1(x_1'))\Psi_{2j}(F_2(x_2'))\,\mathrm{d}F_1(x_1')\mathrm{d}F_2(x_2')$$
$$= \lambda_{1i}\Psi_{1i}(F_1(x_1))\lambda_{2j}\Psi_{2j}(F_2(x_2)). \tag{66}$$

Let $u = F_1(x_1),\ v = F_2(x_2)$ and denote $u' = F_1(x_1'),\ v' = F_2(x_2')$. Then $\mathrm{d}u = \mathrm{d}F_1(x_1),\ \mathrm{d}v = \mathrm{d}F_2(x_2)$, and the support of the integral changes from $\mathbb{R}^2$ to $[0,1]\times[0,1]$. Thus, (66) becomes

$$\int_0^1\int_0^1 [u\wedge u' - uu'][v\wedge v' - vv']\ \Psi_{1i}(u')\Psi_{2j}(v')\,\mathrm{d}u'\mathrm{d}v' = \lambda_{1i}\Psi_{1i}(u)\lambda_{2j}\Psi_{2j}(v).$$

Since X_1 and X_2 are independent, so are u and v, and consequently, we can split the above into two identical integral equations as follows:

$$\begin{cases} \int_0^1 [u\wedge u' - uu']\ \Psi_{1i}(u')\,\mathrm{d}u' = \lambda_{1i}\Psi_{1i}(u) \\ \int_0^1 [v\wedge v' - vv']\ \Psi_{2j}(v')\,\mathrm{d}v' = \lambda_{2j}\Psi_{2j}(v) \end{cases} \tag{67}$$

The solution for the above is discussed in Anderson and Darling (1952). In short, an eigen-equation of the form

$$\int_0^1 [x\wedge x' - xx']\ \Psi(x')\,\mathrm{d}x' = \lambda\Psi(x) \tag{68}$$

can be considered to be the continuous equivalent of (64). Under the normalizing condition

$$\int_0^1 \Psi^2(x)\,\mathrm{d}x = 1,$$

the functions Ψ satisfy the condition $\Psi(0) = \Psi(1) = 0$, as the integrals on the left side of the equations in (68) vanish for $x = 0, 1$. Differentiating (68) twice, with respect to x, gives

$$\lambda\frac{\mathrm{d}^2\Psi(x)}{\mathrm{d}x^2} + \Psi(x) = 0.$$

The solutions of the above differential equation, satisfying the previously mentioned conditions, are

$$\Psi_j(x) = \sqrt{2}\sin(j\pi x),\ 0 \leq x \leq 1,\ j = 1, 2, ...; \tag{69}$$

corresponding to the eigenvalues

$$\lambda_j = \frac{1}{j^2\pi^2}. \tag{70}$$

Following the same string of arguments, we can translate it to our cases (67) and arrive at

$$\begin{cases} \text{Eigenfunctions:} & \Psi_{1k}(x) = \Psi_{2k}(x) =: \Psi_k(x) = \sqrt{2}\sin(k\pi x),\ 0 \leq x \leq 1; \\ \text{Eigenvalues:} & \lambda_{1k} = \lambda_{2k} =: \lambda_k = \dfrac{1}{k^2\pi^2}; \end{cases} \tag{71}$$

for $j = 1, 2, \ldots$. Here, we have introduced a common notation for the eigenfunctions Ψ_k and eigenvalues λ_k since their forms are identical in both the cases when solving (67).

Although a breach in notational integrity, we can express $\mathcal{Y}_n$ as a function of just $u, v \in [0, 1]$. The *principal components* are defined by projecting $\mathcal{Y}_n(u, v)$ onto $\Psi_0(i, j) = \Psi_i(u)\Psi_j(v)$, i.e.,

$$\begin{aligned} \Gamma_n^*(i, j) &:= \int_0^1 \int_0^1 \Psi_i(u)\Psi_j(v)\, \mathcal{Y}_n(u, v)\, \mathrm{d}u\mathrm{d}v,\ i, j = 1, 2, ... \\ &= \int_0^1 \int_0^1 \sqrt{2}\sin(i\pi u)\sqrt{2}\sin(j\pi v)\, \mathcal{Y}_n(u, v)\, \mathrm{d}u\mathrm{d}v \end{aligned} \tag{72}$$

where $\Gamma_n^*(i, j)$ are uncorrelated with zero-means and variances $(\frac{1}{ij\pi^2})^2$. We define the unit variance principal components as

$$\Gamma_n(i, j) := ij\pi^2 \cdot \Gamma_n^*(i, j) = Z_{ni} \cdot Z_{nj}, \tag{73}$$

where by construction, Z_{ni}, Z_{nj} are uncorrelated, with zero mean and unit variance, and as $n \longrightarrow \infty$,

$$Z_{ni}, Z_{nj} \xrightarrow[\mathrm{d}]{\text{IID}} N(0, 1),$$

Therefore, under the true unmixing matrix $\mathbf{B}$,

$$\Gamma_n(i, j) \xrightarrow{\mathrm{d}} Z_1 \cdot Z_2, \text{ where } Z_1, Z_2 \text{ are IID } N(0, 1). \tag{74}$$

The product of two independent standard Gaussian random variables (Gaunt (2018)) does not have a simple closed-form density function: If W is a product of two independent standard Gaussian distributions, then the density function of W is given by

$$f_W(w) = \frac{1}{\pi} K_0(|w|), \; w \in \mathbb{R},$$

where $K_0(\cdot)$ is the modified Bessel[3] function of the second kind. We have the benefit of simulation to generate the required density, and hence the 95% cut-off points that can be used for constructing confidence regions, discussed in the next section. The Figure 3.1 shows the distribution of the density of a product of independent standard Gaussian distributions.

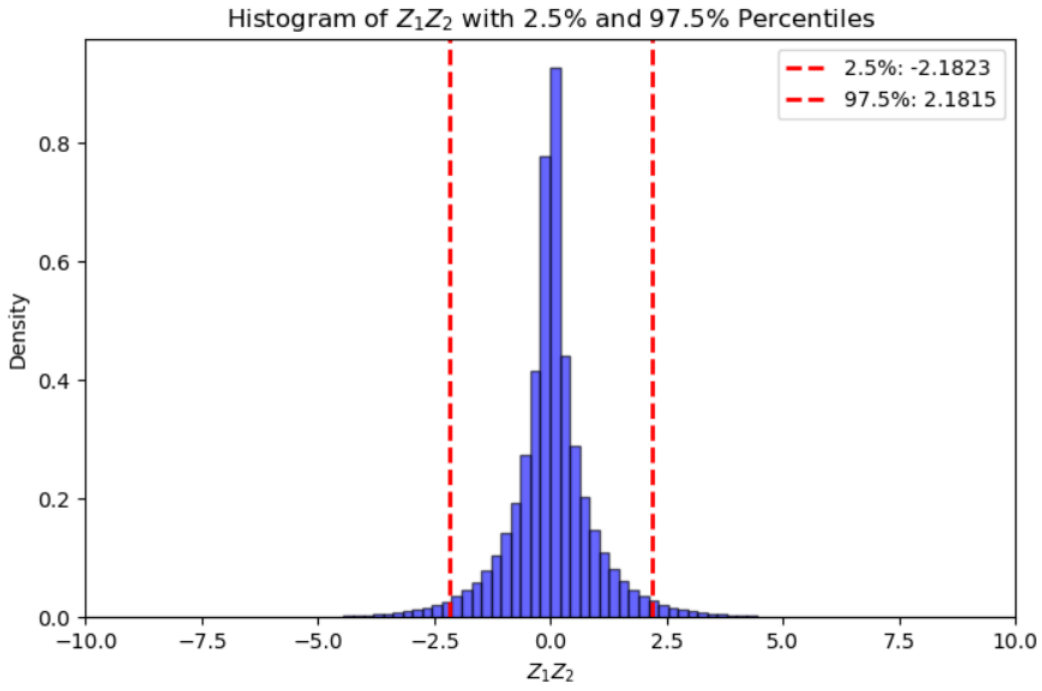

Figure 3.1: The (simulated) distribution of a product of standard Gaussian distributions. The 95% cutoff points are labeled.

[3] cf. Appendix A.6.

3.2.2 Confidence Set for $\mathbf{B}$ using Principal Components

While we have already used ρ_w to extract a confidence set for $\mathbf{B}$, there are certain limitations to it (cf. Section 2.4.2 and the example that follows). Additionally, it involves calculations of a U-statistic, which is taxing on computational resources and time. As such, applying the procedure for larger data becomes tedious. Instead, we present another approach using the empirical process (62), more specifically, its principal components. This is designed to bypass the hurdle posed by computation and time constraints, as it involves fewer computations. The procedure will compare each[4] of the sample principal components to the the theoretical values which, under the true unmixing matrix $\mathbf{B}_0$, follows the distribution (74). Note that the $(i,j)^{\text{th}}$ PC is given by:

$$\begin{aligned}\frac{\Gamma_n(i,j)}{ij\pi^2} =& \int_{\mathbb{R}^2} \sqrt{n}[F_n(x_1,x_2) - F_{n1}(x_1)F_{n2}(x_2)]\psi_i(F_1(x_1))\psi_j(F_2(x_2))\,\mathrm{d}F_1(x_1)\mathrm{d}F_2(x_2)\\ =&\sqrt{n}\int_{\mathbb{R}^2}\left[\frac{1}{n}\sum_k \mathbb{1}\{x_{1k}\le x_1\}\mathbb{1}\{x_{2k}\le x_2\} - \left(\frac{1}{n}\sum_r \mathbb{1}\{x_{1r}\le x_1\}\right)\right.\\ &\left.\left(\frac{1}{n}\sum_r \mathbb{1}\{x_{2r}\le x_2\}\right)\right]\psi_i(F_1(x_1))\psi_j(F_2(x_2))\,\mathrm{d}F_1(x_1)\mathrm{d}F_2(x_2)\\ =&\sqrt{n}\left[\frac{1}{n}\sum_k \Psi_i(F_1(x_{1k}))\Psi_j(F_2(x_{2k})) - \left(\frac{1}{n}\sum_r \Psi_i(F_1(x_{1r}))\right)\left(\frac{1}{n}\sum_r \Psi_i(F_2(x_{2r}))\right)\right].\end{aligned} \tag{75}$$

where

$$\Psi_i(F(x_k)) = \int_{\mathbb{R}} \mathbb{1}\{x_k \le x\}\psi_i(F(x))\,\mathrm{d}F(x) = \int_{x_k}^{\infty} \psi_i(F(x))\,\mathrm{d}F(x).$$

Let $F(x) = u \implies \mathrm{d}F(x) = \mathrm{d}u$ and the support of the integral changes from (x_k,∞) to $(F(x_k),1)$, i.e,,

$$\Psi_i(F(x_k)) = \int_{F(x_k)}^{1} \psi(u)\,\mathrm{d}u. \tag{76}$$

[4] As there are infinitely many principal components, it would be more prudent to say that we compare the first few principal components.

Further, $\psi_j(u) = \sqrt{2}\sin(j\pi u), j = 1, 2, ...,$ and so

$$\begin{aligned}\Psi_i(F(x_k)) &= \int_{F(x_k)}^{1} \sqrt{2}\sin(j\pi u)\,\mathrm{d}u \\ &= \sqrt{2}\left[-\frac{\cos(j\pi u)}{j\pi}\right]_{F(x_k)}^{1} \\ &= \frac{\sqrt{2}}{j\pi}\left[\cos(j\pi F(x_k)) - \cos j\pi\right]. \end{aligned} \tag{77}$$

For the sample principal components, replace F_1 and F_2 by their empirical counterparts F_{n1} and F_{n2} respectively, in (75). Then the expressions involved in the second term become

$$\begin{aligned}\sum_r \Psi_i(F_{n1}(x_{1r})) &= \Psi_i(F_{n1}(x_{11})) + \cdots + \Psi_i(F_{n1}(x_{1n})) \\ &= \Psi_i(F_{n1}(x_{1(1:n)})) + \cdots + \Psi_i(F_{n1}(x_{1(n:n)})) \\ &= \sum_r \Psi_i(F_{n1}(x_{1(r:n)})), \quad x_{1(i:n)} \text{ is the } i^{\text{th}} \text{ order statistic of } x_1, \\ &= \sum_r \Psi_i\left(\frac{r}{n}\right), \end{aligned}$$

and similarly,

$$\sum_r \Psi_j(F_{n2}(x_{2r})) = \sum_r \Psi_j\left(\frac{r}{n}\right).$$

The first term becomes

$$\sum_k \Psi_i(F_{n1}(x_{1k}))\Psi_j(F_{n2}(x_{2k})) = \sum_k \Psi_i\left(\frac{\mathrm{R}_{\#}(x_{1k})}{n}\right)\Psi_j\left(\frac{\mathrm{R}_{\#}(x_{2k})}{n}\right),$$

where $\mathrm{R}_{\#}(x_{ik})$ denotes the rank of k^{th} observation of $x_i, i = 1, 2$; among all n observations.

Consequently, the $(i, j)^{\text{th}}$ sample principal components $\gamma_n(i, j)$ are given by

$$\begin{aligned}\gamma_n(i,j) =& ij\pi^2\sqrt{n}\left[\frac{1}{n}\sum_k \Psi_i\left(\frac{\mathrm{R}_{\#}(x_{1k})}{n}\right)\Psi_j\left(\frac{\mathrm{R}_{\#}(x_{2k})}{n}\right)\right. \\ &\left. -\frac{1}{n^2}\left(\sum_r \Psi_i\left(\frac{r}{n}\right)\right)\left(\sum_r \Psi_j\left(\frac{r}{n}\right)\right)\right], \end{aligned} \tag{78}$$

where the explicit form of Ψ is obtained in (77).

3.2.3 Confidence Region

We adopt a route analogous to that used in Section 2.4 to extract angles satisfying a prescribed criterion. In the previous case, this involved comparing the values attained by the statistic ρ_w for a candidate unmixing matrix $\mathbf{B}_\theta$, with $\theta \in [0, 2\pi]$, against the 95% cutoff derived from the distribution of ρ_w under the true unmixing matrix. In this context, we instead evaluate a finite number of sample principal components, denoted by $\gamma_n(i, j)$, and compare them to their corresponding theoretical values. If a sample principal component lies within the interval $[-2.182, 2.182]$, defined by the 2.5^{th} and 97.5^{th} percentiles of the null distribution, the associated angle θ is accepted.

Based on the observations $\mathbf{Y}$ and an unmixing matrix candidate $\mathbf{B}_\theta$, we can generate the corresponding independent components from the relation

$$\mathbf{X} = \mathbf{B}_\theta \mathbf{Y}. \tag{79}$$

As such, to compute the sample principal components $\gamma_n(i, j)$, we replace the x_{1k} and x_{2k} by

$$\mathbf{x}_k = \begin{pmatrix} x_{1k} \\ x_{2k} \end{pmatrix} = \mathbf{B}_\theta \begin{pmatrix} y_{1k} \\ y_{2k} \end{pmatrix} = \mathbf{B}_\theta \mathbf{y}_k,$$

where $\mathbf{y}_k$ is the k^{th} observation.

Before proceeding to construct a confidence region for $\mathbf{B}$ based on the principal components, it is informative to visualize the behavior of these components under various choices of the unmixing matrix. Specifically, we consider four cases: the true unmixing matrix $\mathbf{B}$, the initial estimate $\bar{\mathbf{B}}$ (interpreted as a rotation of $0°$), a rotated version $\mathbf{B}_\theta$ of $\bar{\mathbf{B}}$ with $\theta = 33°$ (0.576 rads) chosen to align with the confidence region obtained in Section 2.4, and finally, the identity matrix $\mathbf{I}_2$, representing an extreme case far from the true unmixing matrix. We use the same synthetic dataset as was used in Example 1 of Section 2.4.3.

The resulting distributions of the sample principal components under each case are displayed in Figure 3.2. Under the true unmixing matrix $\mathbf{B}$, only two principal components fall outside the

specified bounds. The candidate $\bar{\mathbf{B}}$ leads to a noticeably large number of rejections along the diagonal entries. Notably, the candidate $\mathbf{B}_\theta$ with $\theta = 33^\circ$ results in no diagonal rejections, validating the angular intervals identified earlier. As anticipated, the identity matrix yields numerous rejections, particularly along the diagonal.

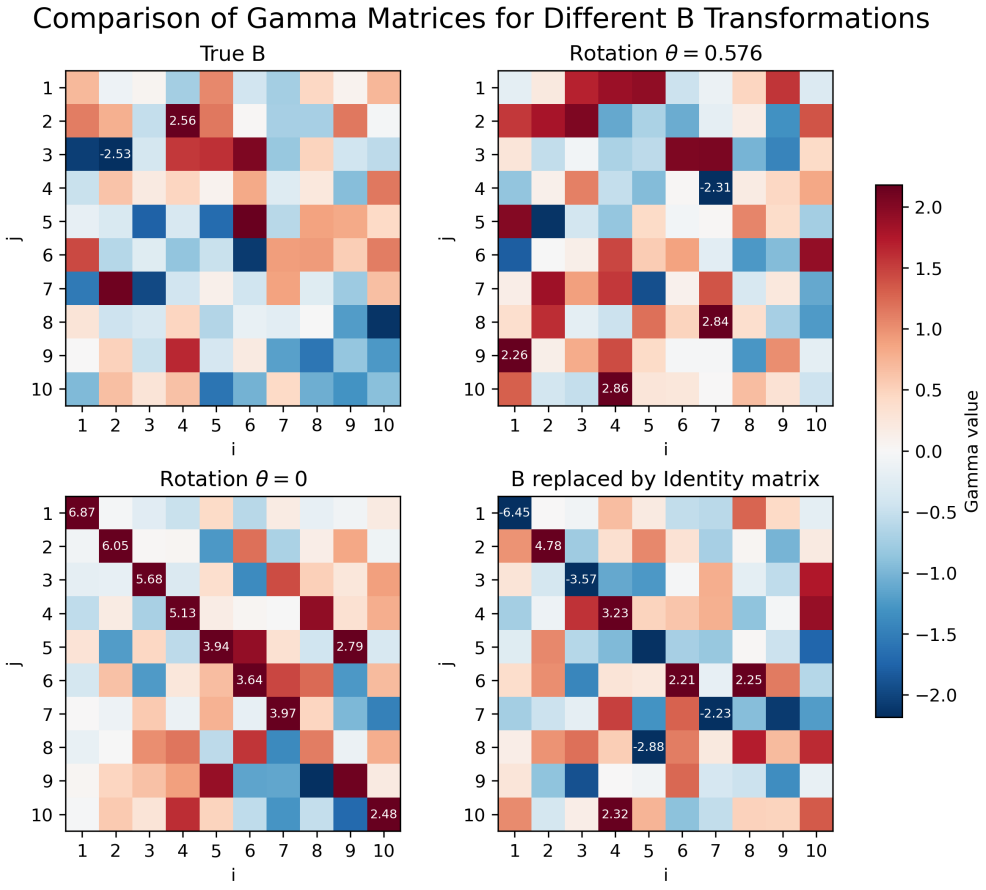

Figure 3.2: The figure shows the first 100 sample principal components computed for 4 different cases. The (i, j) cell in each heatmap corresponds to $\gamma_n(i, j)$ for the specific case. The values are color-coded from blue to red, in increasing order. Only values that are outside the 95% confidence interval are displayed.

3.2.4 Simulated Example

Example 1

We use the same synthetic dataset as previously used in the Example 1 of Section 2.4.3, examining the four cases, namely

Algorithm 6 Constructing Confidence Region Using Principal Components

Input: Observed data $\mathbf{y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$
Compute the sample dispersion matrix $\mathbf{S}_y$
Perform EVD on $\mathbf{S}_y$ to obtain eigenpairs $(\lambda_i, \mathbf{p}_i)$ for $i = 1, 2$
Initialize:

$$\bar{\mathbf{B}} = \left[\frac{1}{\sqrt{\lambda_1}}\mathbf{p}_1 \quad \frac{1}{\sqrt{\lambda_2}}\mathbf{p}_2\right]$$

Generate candidate unmixing matrices:

$$\mathbf{B}_\theta = \mathbf{C}_\theta \bar{\mathbf{B}}, \quad \text{where } \mathbf{C}_\theta \text{ is a rotation matrix}$$

for all $\theta \in [0, 2\pi]$ **do**
    Compute sample principal components $\gamma_n(i, j),\ i, j = 1, \ldots, 10$;
    if $\gamma_n(i, j) \in [-2.182, 2.182]$ **then**
        Accept the corresponding θ for $\mathbf{B}_\theta$
    end if
end for
Output: Accepted set of unmixing matrices $\mathbf{B}_\theta$

(1) Case 1: X_1 sampled from a Laplace distribution with mean 0 and a scale parameter of $\frac{1}{\sqrt{2}}$, and the variable X_2 sampled from a uniform distribution defined over the interval $(-\sqrt{3}, \sqrt{3})$. The mixing matrix used is

$$\mathbf{A} = \begin{bmatrix} 8 & -12 \\ -2 & 15 \end{bmatrix} \implies \mathbf{B} = \mathbf{A}^{-1} = \begin{bmatrix} 0.15625 & 0.12500 \\ 0.02083 & 0.08333 \end{bmatrix}.$$

(2) Case 2: Same sources as in Case 1, but the mixing matrix is replaced by $\mathbf{I}_2$.

(3) Case 3: Both sources are sampled from independent standard Gaussian distributions, and the mixing matrix is the same as in Case 1.

(4) Case 4: Both the sources are Gaussian and the mixing matrix is the identity matrix.

For each candidate unmixing matrix $\mathbf{B}_\theta$, we evaluate the first 100 principal components $\gamma_n(i, j)$, where both indices i and j range from 1 to 10. Due to the inherent variability in simulation-based studies, a small number of sample principal components are anticipated to fall outside the interval $[-2.182, 2.182]$. This behavior can occur even when the true source signals — those used to generate the synthetic observations — are employed in the transformation. This can be seen in the first PC heatmap (under true $\mathbf{B}$) in Figure 3.2. These deviations are often minor and should not be

interpreted as significant failures of the procedure. As such, we do not impose the strict condition

$$\gamma_n(i,j) \in [-2.182, 2.182] \quad \forall\, i,j = 1,\ldots,10.$$

instead opting for a more lenient acceptance criterion: an angle θ is deemed acceptable if at least p out of the 100 sample principal components fall within the specified interval. This p is a subjective choice. In the present example, we set $p = 95$. Based on our observations as expressed in Figure 3.2, we set the condition as the following: At least 9 of the diagonal principal components, and a total of 95 computed principal components lie within $[-2.182, 2.182]$, the angle θ is included in the confidence set. This is repeated for all the four cases, and applying this to the simulated data yields the following intervals of accepted angles per case:

- Case 1: $(32.5, 35.5)$, $(122.5, 124.5)$, $(213, 215.5)$, $(302.5, 305.5)$.
- Case 2: Many smaller intervals are formed, with about 56% of the total 2π being accepted.
- Case 3: $(31.5, 37)$, $(121.5, 127)$, $(211.5, 217)$, $(301.5, 307)$.
- Case 4: About 90% of the angles are accepted.

If we relax the condition a little, even more angles will be accepted, especially for Cases 2 and 4. For instance, in Case 4, if we lax the condition to 94 (just a change of 1) computed principal components lie within the interval, then nearly all angles are accepted. This suggests that while principal components can be used to find the confidence region (as Case 1) and it's computationally fast, it is also a crude method as it is too dependent on a subjective choice.

For Case 1, the accepted angles are

$$\theta \in [32.5^\circ, 35.5^\circ] \cup [122.5^\circ, 124.5^\circ] \cup [213^\circ, 215.5^\circ] \cup [302.5^\circ, 305.5^\circ] =: \Theta.$$

These intervals are closely aligned with those obtained in Example 1 of Section 2.4.3, in a sense validating both approaches. The confidence region for the unmixing matrix $\mathbf{B}$ is composed of all

matrices $\mathbf{B}_\theta$ obtained by rotating the matrix

$$\bar{\mathbf{B}} = \begin{bmatrix} -0.1554 & -0.0336 \\ -0.1501 & 0.0347 \end{bmatrix}$$

by the corresponding accepted angles θ. Formally, the confidence region is given by

$$\mathcal{B}_\theta = \left\{ \mathbf{B}_\theta = \mathbf{C}_\theta \bar{\mathbf{B}} \;\middle|\; \theta \in \Theta,\ \mathbf{C}_\theta \text{ orthogonal} \right\}.$$

Example 2

The data is generated from the same sources and mixing matrix as Example 2 of Section 2.4.3. Similar to the previous example, we divide it into 4 cases with

(1) Case 1: X_1 sampled from standard Gaussian distribution, and variable X_2 sampled from a uniform distribution defined over the interval $(-\sqrt{3}, \sqrt{3})$ and mixed using the matrix

$$\mathbf{A} = \begin{bmatrix} 3 & -4 \\ -2 & 7 \end{bmatrix} \implies \mathbf{B} = \mathbf{A}^{-1} = \begin{bmatrix} 0.53846 & 0.15385 \\ 0.30769 & 0.23077 \end{bmatrix}.$$

(2) Case 2: Same sources as in Case 1, but the mixing matrix is replaced by $\mathbf{I}_2$.

(3) Case 3: Both sources are sampled from independent standard Gaussian distributions, and the mixing matrix is the same as in Case 1.

(4) Case 4: Both the sources are Gaussian and the mixing matrix is the identity matrix.

Based on the computed principal components $\gamma_n(i, j)$, the following intervals of the angle θ (in degrees) are obtained:

- Case 1: $(40.5, 46.5)$, $(130, 137.5)$, $(220.5, 227.5)$, $(310.5, 317.5)$.
- Case 2: Many smaller intervals are formed, with about 68% of the total 2π being accepted.
- Case 3: $(40, 47.5)$, $(130.5, 136.5)$, $(220.5, 226.5)$, $(310.5, 316.5)$.

- Case 4: About 71% of the angles are accepted.

For case 1, the confidence region for the unmixing matrix $\mathbf{B}$ is composed of all matrices $\mathbf{B}_\theta$ obtained by rotating the matrix

$$\bar{\mathbf{B}} = \begin{bmatrix} -0.1554 & -0.0336 \\ -0.1501 & 0.0347 \end{bmatrix}$$

by the corresponding accepted angles θ. Formally, the confidence region is given by

$$\mathcal{B}_\theta = \left\{ \mathbf{B}_\theta = \mathbf{C}_\theta \bar{\mathbf{B}} \;\middle|\; \theta \in \Theta,\ \mathbf{C}_\theta \text{ orthogonal} \right\}.$$

3.3 Asymptotics of the Minimum Distance Estimator

As stated before, our main goal in this study is to devise an estimator of the unmixing matrix $\mathbf{B}$ from ρ_w. The theoretical value $\rho_w(\mathbf{B})$ is minimized when the unmixing matrix attains the "true" value $\mathbf{B}_0$, while the sample equivalent

$$\rho_w \coloneqq \rho_w\Big(F_n(\mathbf{y} \mid \mathbf{B}), F_n^{\perp}(\mathbf{y} \mid \mathbf{B}) \Big)$$

is minimized at some value of $\hat{\mathbf{B}}$, i.e.

$$\hat{\mathbf{B}} = \arg\min_{\mathbf{B}} \rho_w.$$

The estimator $\hat{\mathbf{B}}$ thus obtained should close the gap between itself and the true value $\mathbf{B}_0$, at least in some limiting sense, i.e., $(\hat{\mathbf{B}} - \mathbf{B}_0) \longrightarrow \mathbf{0}$ as $n \longrightarrow \infty$. More specifically, we can say that we wish to find the limiting distribution of the expression

$$\lim_{n \longrightarrow \infty} a_n(\hat{\mathbf{B}} - \mathbf{B}_0), \text{ where } a_n \text{ is a sequence dependent on } n. \tag{80}$$

We employ a standard technique called the *reparametrization* to expand on this and examine the limiting behaviors. This allows us to express the parameter of interest $\mathbf{B}$ in terms of another

parameter, which may be more convenient for inference. Let

$$a_n(\hat{\mathbf{B}} - \mathbf{B}_0) = \hat{\mathbf{C}} \Leftrightarrow \hat{\mathbf{B}} = \mathbf{B}_0 + \frac{1}{a_n}\hat{\mathbf{C}}. \tag{81}$$

Set $a_n = \sqrt{n}$, and define the reparametrization of $\mathbf{B}$ in terms of the new parameter matrix $\mathbf{C}$ as follows:

$$\mathbf{B} = \mathbf{B}_0 + \frac{1}{\sqrt{n}}\mathbf{C}. \tag{82}$$

Below we provide a few alternate ways to express (82) in terms of different matrices which are vital in handling the complicated expressions and equations that arise later on.

$$\mathbf{B} \;=\; \mathbf{B}_0 + \frac{1}{\sqrt{n}}\mathbf{C} \;=\; \left(\mathbf{I} + \frac{1}{\sqrt{n}}\mathbf{C}\mathbf{B}_0^{-1}\right)\mathbf{B}_0 \;=\; \left(\mathbf{I} + \frac{1}{\sqrt{n}}\mathbf{K}\right)\mathbf{B}_0 \;=\; \mathbf{D}_n\mathbf{B}_0. \tag{83}$$

In the above expressions, $\mathbf{B}_0$ is the true value of $\mathbf{B}$, and the observations and independent components are related through the usual $\mathbf{X}_i = \mathbf{B}_0\mathbf{Y}_i,\ i = 1, 2, \ldots, n$. The matrix $\mathbf{C}$ is an arbitrary matrix independent of n. So, as $n \longrightarrow \infty$, $\mathbf{B} \longrightarrow \mathbf{B}_0$, the true value. We simplify the larger notations by defining the matrices

$$\mathbf{K} \coloneqq \mathbf{C}\mathbf{B}_0^{-1}, \text{ and } \mathbf{D}_n \coloneqq \mathbf{I} + \frac{1}{\sqrt{n}}\mathbf{K}.$$

We shift our interest from the matrix $\mathbf{B}$ to the matrix $\mathbf{C}$ (or more specifically the matrix $\mathbf{K}$), which now acts as a new and more convenient parameter to work with. The function ρ_w can also be expressed in terms of the new parameter $\mathbf{C}$, i.e.,

$$\rho_w(\mathbf{B}) = \rho_w\left(\mathbf{B}_0 + \frac{1}{\sqrt{n}}\mathbf{C}\right)$$

and thus we may only consider the minimization of ρ_w with respect to the matrix $\mathbf{C}$ to find a minimizer.

The required estimate of the unmixing matrix $\hat{\mathbf{B}}$ can be obtained from the relation (81), with $a_n = \sqrt{n}$. Further, the limiting distribution of $\sqrt{n}(\hat{\mathbf{B}} - \mathbf{B}_0)$ is the same as that of $\hat{\mathbf{C}}$. So, instead

of dealing with ρ_w as a function of $\mathbf{B}$, we treat it as a function of $\mathbf{C}$ and minimize for the new parameter, yielding the minimizer $\hat{\mathbf{C}}$.

The study is conducted on the associated empirical process involved with ρ_w, viz. the *distance* gievn by

$$F_n(\mathbf{y} \mid \mathbf{B}) - F_n^{\perp}(\mathbf{y} \mid \mathbf{B}).$$

This can be expressed in terms of the new parameter $\mathbf{C}$ as follows:

$$\sqrt{n}\left[F_n\left(\mathbf{y}\middle|\mathbf{B}_0 + \frac{1}{\sqrt{n}}\mathbf{C}\right) - F_n^{\perp}\left(\mathbf{y}\middle|\mathbf{B}_0 + \frac{1}{\sqrt{n}}\mathbf{C}\right)\right] \xrightarrow{\mathtt{d}} \mathcal{G}(\mathbf{y}) + \mathcal{P}(\mathbf{y}, \mathbf{C}) \tag{84}$$

where $\mathcal{G}(y)$, a function of Gaussian processes, is the deterministic[5] part, while $\mathcal{P}(\mathbf{y}, \mathbf{C})$ is the part dependent on the parameter. Then,

$$n\rho_w\left(\mathbf{B}_0 + \frac{1}{\sqrt{n}}\mathbf{C}\right) \xrightarrow{\mathtt{d}} \int_{\mathbb{R}^d}\left[\mathcal{G}(\mathbf{y}) + \mathcal{P}(\mathbf{y}, \mathbf{C})\right]^2 w(\mathbf{y})\,\mathrm{d}\mathbf{y} \tag{85}$$

3.3.1 Two-Dimensional Case

In this subsection, we investigate what we have just explained in a more mathematical framework, constrained to the two-dimensional case. We introduce some notations as well as repeat a few important ones for clarity and self-containment.

Firstly, the matrices involved and their entries are related as follows:

$$\mathbf{K} \coloneqq \mathbf{C}\mathbf{B}_0^{-1} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \text{ and } \mathbf{D}_n \coloneqq \mathbf{I} + \frac{1}{\sqrt{n}}\mathbf{K} = \begin{bmatrix} 1 + \frac{1}{\sqrt{n}}k_{11} & \frac{1}{\sqrt{n}}k_{12} \\ \frac{1}{\sqrt{n}}k_{21} & 1 + \frac{1}{\sqrt{n}}k_{22} \end{bmatrix}. \tag{86}$$

Note that as $\mathbf{B}_0$ is the true value of the unmixing matrix the following expression experiences a

[5] As the dimension increases, this function of Gaussian processes becomes less tractable. At least for the 2-D case, its form and associated calculations are somewhat simple.

slight alteration to account for this. For $i = 1, 2, \ldots, n$,

$$\begin{aligned}
&\mathbb{1}\left\{\mathbf{b}_1^\top \mathbf{Y}_i \leq y_1, b_2^\top \mathbf{Y}_i \leq y_2\right\} \\
=&\mathbb{1}\left\{\mathbf{B}\mathbf{Y}_i \leq \mathbf{y}\right\} \\
=&\mathbb{1}\left\{\mathbf{D}_n \mathbf{B}_0 \mathbf{Y}_i \leq \mathbf{y}\right\}, \text{ from (83)} \\
=&\mathbb{1}\left\{\mathbf{D}_n \mathbf{X}_i \leq \mathbf{y}\right\} = \mathbb{1}\{\mathbf{d}_{n1}^\top \mathbf{X}_i \leq y_1, \mathbf{d}_{n2}^\top \mathbf{X}_i \leq y_2\},
\end{aligned}$$

where $\mathbf{D}_n \coloneqq \begin{bmatrix} \mathbf{d}_{n1}^\top \\ \mathbf{d}_{n2}^\top \end{bmatrix}$ (row-wise expression) with the determinant of $\mathbf{D}_n$ being

$$\det(\mathbf{D}_n) \coloneqq k_n = 1 + \frac{k_{11}}{\sqrt{n}} + \frac{k_{22}}{\sqrt{n}} + \frac{k_{11}k_{22}}{n} - \frac{k_{12}k_{21}}{n}.$$

At least asymptotically, the determinant k_n is non-zero, and as such, $\mathbf{D}_n$ is invertible. Further, let $\mathbf{H}_n = \mathbf{D}_n^{-1}$. Define the following empirical distribution functions and their corresponding cumulative distribution functions:

$$\hat{F}_n(y_1, y_2) \coloneqq \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{\mathbf{d}_{n1}^\top \mathbf{X}_i \leq y_1, \mathbf{d}_{n2}^\top \mathbf{X}_i \leq y_2\};\ \hat{F}_{nj}(y_j) \coloneqq \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{\mathbf{d}_{nj}^\top \mathbf{X}_i \leq y_j\},\ j = 1, 2.$$

$$F_n(y_1, y_2) \coloneqq \mathbb{P}\{\mathbf{d}_{n1}^\top \mathbf{X} \leq y_1, \mathbf{d}_{n2}^\top \mathbf{X} \leq y_2\};\ F_{nj}(y_j) \coloneqq \mathbb{P}\{\mathbf{d}_{nj}^\top \mathbf{X} \leq y_j\},\ j = 1, 2.$$

Under the reparametrization of the unmixing matrix, the left side of the expression in result 3.1.3 can be split into two parts — a deterministic part and another which involves the new parameter matrix $\mathbf{C}$ (or $\mathbf{K}$).

Result 3.3.1. *Consider the reparametrized form of* $\mathbf{B}$ *as in* (83)*. Then*

$$\begin{aligned}
\lim_{n\longrightarrow\infty} \sqrt{n}\Big[F_n(\mathbf{y} \mid \mathbf{B}) - F_n^{\perp}(\mathbf{y} \mid \mathbf{B})\Big] =& \mathcal{B}_1(F_1((y_1)) \cdot \mathcal{B}_2(F_2(y_2)) \\
&+ \lim_{n\longrightarrow\infty} \sqrt{n}\Big[F_n(y_1, y_2) - F_{n1}(y_1)F_{n2}(y_2)\Big].
\end{aligned}$$

Proof. If $\mathbf{B} = \left(\mathbf{I} + \frac{1}{\sqrt{n}}\mathbf{K}\right)\mathbf{B}_0 = \mathbf{D}_n\mathbf{B}_0$, $F_n(\mathbf{y} \mid \mathbf{B}) - F_n^{\perp}(\mathbf{y} \mid \mathbf{B})$ changes to $\hat{F}_n(y_1, y_2) - \hat{F}_{n1}(y_1)\hat{F}_{n2}(y_2)$. Now

$$
\begin{aligned}
&\sqrt{n}\left[\hat{F}_n(y_1, y_2) - \hat{F}_{n1}(y_1)\hat{F}_{n2}(y_2)\right] \qquad (87)\\
&= \sqrt{n}\Big[\hat{F}_n(y_1, y_2) - F_n(y_1, y_2) + F_n(y_1, y_2)\\
&- \hat{F}_{n1}(y_1)\hat{F}_{n2}(y_2) + F_{n1}(y_1)F_{n2}(y_2) - F_{n1}(y_1)F_{n2}(y_2)\Big]\\
&= \sqrt{n}\left[\hat{F}_n(y_1, y_2) - F_n(y_1, y_2) - \hat{F}_{n1}(y_1)\hat{F}_{n2}(y_2) + F_{n1}(y_1)F_{n2}(y_2)\right]\\
&+ \sqrt{n}\left[F_n(y_1, y_2) - F_{n1}(y_1)F_{n2}(y_2)\right],
\end{aligned}
$$

where the first part has an asymptotic covariance structure as in corollary 3.1.2. The calculations are identical to that in lemma 3.1.1. Thus, as $n \longrightarrow \infty$, (87) equals

$$\mathcal{B}_1(F_1((y_1)) \cdot \mathcal{B}_2(F_2(y_2)) + \lim_{n\longrightarrow\infty}\sqrt{n}\left[F_n(y_1, y_2) - F_{n1}(y_1)F_{n2}(y_2)\right]$$

■

Now, we make an adjustment, albeit minor, to the relations discussed thus far. Instead of working with the matrix $\mathbf{C}$, and the relation

$$\mathbf{B} = \mathbf{B}_0 + \frac{1}{\sqrt{n}}\mathbf{C},$$

we readjust the expression to work with matrix $\mathbf{K}$, i.e.

$$\mathbf{B} = \left(\mathbf{I} + \frac{1}{\sqrt{n}}\mathbf{K}\right)\mathbf{B}_0, \text{ where } \mathbf{K} = \mathbf{C}\mathbf{B}_0^{-1} \qquad (88)$$

We now attend to the part in the above result that is dependent on the parameter $\mathbf{K}$, viz, the last term in the above result:

$$\lim_{n\longrightarrow\infty}\sqrt{n}\left[F_n(y_1, y_2) - F_{n1}(y_1)F_{n2}(y_2)\right] \qquad (89)$$

To simplify this expression, we first need to discuss the transformation it entails. Owing to the

fact that the components of $\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$ are independent, the joint density function is the product $f_1(x_1)\ f_2(x_2)$.

Under the transformation $\mathbf{Z} \coloneqq \mathbf{D}_n\mathbf{X} = \begin{bmatrix} \mathbf{d}_{n1}^\top \\ \mathbf{d}_{n2}^\top \end{bmatrix} \mathbf{X}$, assuming $\mathbf{D}_n$ is invertible, we have the relation

$$\mathbf{X} = \mathbf{D}_n^{-1}\mathbf{Z} = \mathbf{H}_n\mathbf{Z} = \begin{bmatrix} \mathbf{h}_{n1}^\top \\ \mathbf{h}_{n2}^\top \end{bmatrix} \mathbf{Z}. \tag{90}$$

The joint PDF of $\mathbf{Z}$ would then be

$$f_1(\mathbf{h}_{n,1}^\top\mathbf{Z})f_2(\mathbf{h}_{n,1}^\top\mathbf{Z}) \cdot \mathcal{J}$$

with the Jacobian of transformation being $\mathcal{J} = |\det(\mathbf{D}_n^{-1})| = \dfrac{1}{|\det(\mathbf{D}_n)|} = \dfrac{1}{|k_n|}$.

Lemma 3.3.2. *The expression* (89) *equals to*

$$-k_{12}f_1(y_1)\mathbb{E}\Big[X_2\mathbb{1}\{X_2 < y_2\}\Big] - k_{21}f_2(y_2)\mathbb{E}\Big[X_1\mathbb{1}\{X_1 < y_1\}\Big].$$

Proof. In (89), the terms

$$F_n(y_1, y_2) = \int_{-\infty}^{y_1}\int_{-\infty}^{y_2} f_1(\mathbf{h}_{n1}^\top\mathbf{x})f_2(\mathbf{h}_{n2}^\top\mathbf{x})\ \frac{1}{k_n}\,\mathrm{d}\mathbf{x},$$

and $F_{n1}(y_1) = F_n(y_1, \infty); F_{n2}(y_2) = F_n(\infty, y_2)$, are joint and marginal cumulative distribution functions of Z_1 and Z_2. The joint density of the independent components $\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$ is $f(x_1, x_2) = f_1(x_1)f_2(x_2)$; the joint cumulative and marginal distribution functions are

$$F(y_1, y_2) = \int_{-\infty}^{y_1}\int_{-\infty}^{y_2} f_1(x_1)f_2(x_2)\mathrm{d}\mathbf{x}, \text{ and } F_1(y_1) = F(y_1, \infty); F_2(y_2) = F(\infty, y_2), \text{ respectively.}$$

The expression (89) is expanded into a sum of 2 new expressions as follows:

$$\lim_{n\longrightarrow\infty}\sqrt{n}\Big[F_n(y_1,y_2)-F_{n1}(y_1)F_{n2}(y_2)\Big]$$
$$=\lim_{n\longrightarrow\infty}\sqrt{n}\Big[F_n(y_1,y_2)-F_1(y_1)F_2(y_2)\Big]+\lim_{n\longrightarrow\infty}\sqrt{n}\Big[F_1(y_1)F_2(y_2)-F_{n1}(y_1)F_{n2}(y_2)\Big] \quad (91)$$

With the first limit in the above, we can proceed in the following manner:

$$F_n(y_1,y_2)-F_1(y_1)F_2(y_2)=\int_{-\infty}^{y_1}\int_{-\infty}^{y_2}\Big(f_1(\mathbf{h}_{n1}^{\top}\mathbf{x})f_2(\mathbf{h}_{n2}^{\top}\mathbf{x})\frac{1}{k_n}-f_1(x_1)f_2(x_2)\Big)\mathbf{dx}$$
$$=\frac{1}{k_n}\int_{-\infty}^{y_1}\int_{-\infty}^{y_2}\Big(f_1(\mathbf{h}_{n1}^{\top}\mathbf{x})f_2(\mathbf{h}_{n2}^{\top}\mathbf{x})-f_1(x_1)f_2(x_2)\Big)\mathbf{dx}+\Big(\frac{1}{k_n}-1\Big)\int_{-\infty}^{y_1}\int_{-\infty}^{y_2}f_1(x_1)f_2(x_2)\mathbf{dx}$$
$$:=\mathcal{A}_1+\mathcal{A}_2.$$

We can expand $\mathcal{A}_1$ as

$$\frac{1}{k_n}\int_{-\infty}^{y_1}\int_{-\infty}^{y_2}\Big(\Big[f_1(\mathbf{h}_{n1}^{\top}\mathbf{x})-f_1(x_1)\Big]f_2(h_{n2}^{\top}\mathbf{x})+\Big[f_2(\mathbf{h}_{n2}^{\top}\mathbf{x})-f_2(x_2)\Big]f_1(x_1)\Big)\,\mathbf{dx}. \quad (92)$$

Note the following relation among the entries of $\mathbf{D}_n$, $\mathbf{H}_n$ and $\mathbf{K}$. If we write the entries of $\mathbf{K}$ as $k_{ij},\ i,j=1,2;$, then $\mathbf{K}=\begin{pmatrix}k_{11} & k_{12}\\ k_{21} & k_{22}\end{pmatrix}$ and $\mathbf{D}_n=\mathbf{I}+\frac{1}{\sqrt{n}}\mathbf{K}$, with determinant of $\mathbf{D}_n$ being k_n (as defined earlier). Let the rows of the matrix $\mathbf{H_n}(=\mathbf{D}_n^{-1})$ be $\mathbf{h}_{n1}^{\top}$ and $\mathbf{h}_{n2}^{\top}$, and define constants $c_{ij},\ i,j=1,2;$ such that

$$\mathbf{H_n}=\frac{1}{k_n}\begin{pmatrix}1+\frac{k_{22}}{\sqrt{n}} & -\frac{k_{12}}{\sqrt{n}}\\ -\frac{k_{21}}{\sqrt{n}} & 1+\frac{k_{11}}{\sqrt{n}}\end{pmatrix}=\begin{pmatrix}\frac{1}{k_n}+\frac{c_{11}}{\sqrt{n}} & \frac{c_{12}}{\sqrt{n}}\\ \frac{c_{21}}{\sqrt{n}} & \frac{1}{k_n}+\frac{c_{22}}{\sqrt{n}}\end{pmatrix}.$$

Of course, as $n\longrightarrow\infty$, the constants $c_{i,j},i,j=1,2$, would converge as below:

$$c_{11}=\frac{k_{22}}{k_n}\longrightarrow k_{22};\ c_{12}=-\frac{k_{12}}{k_n}\longrightarrow -k_{12};\ c_{21}=-\frac{k_{21}}{k_n}\longrightarrow -k_{21};\ c_{22}=\frac{k_{11}}{k_n}\longrightarrow k_{11}.$$

Through Taylor's expansion (excluding the terms in the expansion which would eventually become negligible),

$$f_1(\mathbf{h}_{n1}^\top \mathbf{x}) = f_1(x_1) + (\mathbf{h}_{n1}^\top \mathbf{x} - x_1) f_1^{'}(x_1)$$
$$\implies f_1(\mathbf{h}_{n1}^\top \mathbf{x}) - f_1(x_1) = \left[\left(\frac{1}{k_n} - 1\right) x_1 + \frac{c_{11}}{\sqrt{n}} x_1 + \frac{c_{12}}{\sqrt{n}} x_2\right] f_1^{'}(x_1).$$

Similarly,

$$f_2(\mathbf{h}_{n2}^\top \mathbf{x}) - f_2(x_2) = \left[\left(\frac{1}{k_n} - 1\right) x_2 + \frac{c_{21}}{\sqrt{n}} x_1 + \frac{c_{22}}{\sqrt{n}} x_2\right] f_2^{'}(x_2).$$

As such (92) can be further expanded into

$$\frac{1}{k_n} \int_{-\infty}^{y_1} \int_{-\infty}^{y_2} \Bigg(\left[\left(\frac{1}{k_n} - 1\right) x_1 + \frac{1}{\sqrt{n}} \mathbf{c}_1^\top \mathbf{x}\right] f_1^{'}(x_1) f_2(\mathbf{h}_{n2}^\top \mathbf{x}) + \left[\left(\frac{1}{k_n} - 1\right) x_2 + \frac{1}{\sqrt{n}} \mathbf{c}_2^\top \mathbf{x}\right] f_2^{'}(x_2) f_1(x_1) \Bigg) \mathbf{dx},$$

and thus,

$$\begin{aligned}
&\lim_{n \longrightarrow \infty} \sqrt{n} \mathcal{A}_1 \\
&= \lim_{n \longrightarrow \infty} \int_{-\infty}^{y_1} \int_{-\infty}^{y_2} \sqrt{n} \left(\frac{1}{k_n} - 1\right) \left[x_1 f_1^{'}(x_1) f_2(\mathbf{h}_{n2}^\top \mathbf{x}) + x_2 f_2^{'}(x_2) f_1(x_1)\right] \mathbf{dx} \\
&+ \lim_{n \longrightarrow \infty} \int_{-\infty}^{y_1} \int_{-\infty}^{y_2} \left[\mathbf{c}_1^\top \mathbf{x}\ f_1^{'}(x_1) f_2(\mathbf{h}_{n2}^\top y) + \mathbf{c}_2^\top \mathbf{x}\ f_2^{'}(x_2) f_1(x_1)\right] \mathbf{dx} \\
&= \int_{-\infty}^{y_1} \int_{-\infty}^{y_2} -(k_{11} + k_{22}) \left[x_1\ f_1^{'}(x_1) f_2(x_2) + x_2\ f_2^{'}(x_2) f_1(x_1)\right] \mathbf{d}x_1 \mathbf{x} \\
&+ \int_{-\infty}^{y_1} \int_{-\infty}^{y_2} \Big[k_{22} x_1 f_1^{'}(x_1) f_2(x_2) - k_{12} x_2 f_1^{'}(x_1) f_2(x_2) \\
&- k_{21} x_1 f_2^{'}(x_2) f_1(x_1) + k_{11} x_2 f_2^{'}(x_2) f_1(x_1)\Big] \mathbf{dx} \\
&= \int_{-\infty}^{y_1} \int_{-\infty}^{y_2} \Big[- k_{11} x_1 f_1^{'}(x_1) f_2(x_2) - k_{12} x_2 f_1^{'}(x_1) f_2(x_2) \\
&- k_{21} x_1 f_2^{'}(x_2) f_1(x_1) - k_{22} x_2 f_2^{'}(x_2) f_1(x_1)\Big] \mathbf{dx}
\end{aligned}$$

$$= -k_{11}\Big[y_1 f_1(y_1) - F_1(y_1)\Big]F_2(y_2) - k_{12} f_1(y_1)\int_{-\infty}^{y_2} x_2 f_2(x_2)\,\mathrm{d}x_2$$
$$- k_{21} f_2(z_2)\int_{-\infty}^{y_1} x_1 f_1(x_1)\,\mathrm{d}x_1 - k_{22}\Big[y_2 f_2(y_2) - F_2(y_2)\Big]F_1(y_1).$$

The limit concerning the second term $\mathcal{A}_2$ can be simplified as:

$$\lim_{n\longrightarrow\infty}\sqrt{n}\mathcal{A}_2 = \lim_{n\longrightarrow\infty}\sqrt{n}\left(\frac{1}{k_n} - 1\right)\int_{-\infty}^{y_1}\int_{-\infty}^{y_2} f_1(x_1)f_2(x_2)\,\mathbf{dx} = -(k_{11}+k_{22})F_1(y_1)F_2(y_2).$$

Combining the last two limits above yields the following form of the first expression in (91), i.e.,

$$\lim_{n\longrightarrow\infty}\sqrt{n}\Big[F_n(y_1,y_2) - F_1(y_1)F_2(y_2)\Big] = -k_{11}y_1 f_1(y_1)F_2(y_2) - k_{12} f_1(y_1)\int_{-\infty}^{y_2} x_2 f_2(x_2)\,\mathrm{d}x_2$$
$$- k_{21} f_2(y_2)\int_{-\infty}^{y_1} x_1 f_1(x_1)\,\mathrm{d}x_1 - k_{22}y_2 f_2(y_2)F_1(y_1). \tag{91.1}$$

Following similar arguments for the second term in (91) and the terms involved are rewritten as follows:

$$F_{n1}(y_1)F_{n2}(y_2) - F_1(y_1)F_2(y_2)$$
$$=\Big[F_{n1}(y_1) - F_1(y_1)\Big]F_{n2}(y_2) + \Big[F_{n2}(y_2) - F_2(y_2)\Big]F_1(y_1)$$
$$=: \mathrm{J}_1\mathrm{J}_2 + \mathrm{J}_3\mathrm{J}_4,$$

where, $\mathrm{J}_4 = F_1(y_1)$; and as $n \longrightarrow \infty$,

$$\mathrm{J}_2 = \int_{-\infty}^{y_2}\int_{-\infty}^{\infty} f_1(\mathbf{h}_{n1}^{\top}\mathbf{x})f_2(\mathbf{h}_{n2}^{\top}\mathbf{x})\,\mathbf{dx} \longrightarrow \int_{-\infty}^{y_2}\int_{-\infty}^{\infty} f_1(x)f_2(x)\,\mathbf{dx} = F_2(y_2)$$

For J_1 and J_3, recall that, without loss of generality, the expectations of the independent components

are assumed to be zero, i.e.,

$$\mathbb{E}X_j = \int_{-\infty}^{\infty} x_j f_j(x_j)\ \mathrm{d}x_j\ = 0;\ j = 1, 2;\ \text{and thus,}$$

$$\lim_{n\longrightarrow\infty} \sqrt{n}\mathrm{J}_1 = (91.1)\Big|_{y_2=\infty} = -k_{11}y_1 f_1(y_1).$$

$$\lim_{n\longrightarrow\infty} \sqrt{n}\mathrm{J}_3 = (91.1)\Big|_{y_1=\infty} = -k_{22}y_2 f_2(y_2).$$

Therefore,

$$\lim_{n\longrightarrow\infty} \sqrt{n}\Big[F_{n1}(y_1)F_{n2}(y_2) - F_1(y_1)F_2(y_2)\Big] = -k_{11}y_1 f_1(y_1)F_2(y_2) - k_{22}y_2 f_2(y_2)F_1(y_1). \tag{91.2}$$

Combining (91.1) and (91.2), we get the following limit (91):

$$\begin{aligned}
&\lim_{n\longrightarrow\infty} \sqrt{n}\Big[F_n(y_1, y_2) - F_{n1}(y_1)F_{n2}(y_2)\Big] \\
&= -k_{12}f_1(y_1)\int_{-\infty}^{y_2} x_2 f_2(x_2)\mathrm{d}x_2 - k_{21}f_2(y_2)\int_{-\infty}^{y_1} x_1 f_1(x_1)\mathrm{d}x_1 \\
&= -k_{12}f_1(y_1)\mathbb{E}\Big[X_2\mathbb{1}\{X_2 < y_2\}\Big] - k_{21}f_2(y_2)\mathbb{E}\Big[X_1\mathbb{1}\{X_1 < y_1\}\Big].
\end{aligned} \tag{93}$$

■

The most notable observation from the above result is that the expression is free of the diagonal entries of the parameter matrix $\mathbf{K}$. So without any consequence, we may assume those entries to be 0, i.e,

$$\mathbf{K} = \begin{bmatrix} 0 & k_{12} \\ k_{21} & 0 \end{bmatrix}. \tag{94}$$

For notational simplification, we shall denote $\mathcal{B}_1(F_1(y_1)) \cdot \mathcal{B}_2(F_2(y_2)) = \mathfrak{B}(\mathbf{y})$, and (93) as

$$-k_{12} f_1(y_1) \mathbb{E}\Big[X_2 \mathbb{1}\{X_2 < y_2\}\Big] - k_{21} f_2(y_2) \mathbb{E}\Big[X_1 \mathbb{1}\{X_1 < y_1\}\Big] = -\tau(\mathbf{y})^\top \mathbf{k},$$

$$\text{where } \tau(\mathbf{y}) = \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} = \begin{pmatrix} f_2(y_2)\mathbb{E}\Big[X_1 \mathbb{1}\{X_1 < y_1\}\Big] \\ f_1(y_1)\mathbb{E}\Big[X_2 \mathbb{1}\{X_2 < y_2\}\Big] \end{pmatrix}, \mathbf{k} = \begin{pmatrix} k_{12} \\ k_{21} \end{pmatrix}, \text{ and } \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}.$$

To obtain estimates of the parameter $\mathbf{K}$, and hence, $\mathbf{B}$, we write[6] the metric (20) as

$$n\rho_w \cong \int_{\mathbb{R}^2} \Big[\mathfrak{B}(\mathbf{y}) - \tau(\mathbf{y})^\top \mathbf{k}\Big]^2 w(\mathbf{y})\, \mathrm{d}\mathbf{y}, \tag{95}$$

and aim to minimize it with respect to $\mathbf{k}$. The integrand in (95) is non-negative, which converts this minimization problem into one of *Least Squares*. The minimizer will always exist, be it a unique solution or infinitely many equivalent (to the minimizer) solutions. These solutions, i.e. the values of k_{12} and k_{21} can be obtained from the first-order partial derivatives.

In order to obtain the least-squares solution to (95), first we expand it into three simpler integrals, followed by taking the derivative with respect to k_{12} and k_{21} separately. Equating both to zero would return a system of two linear equations in two unknowns (the parameters k_{12} and k_{21}), which can be solved to get the minimizing values of the non-trivial entries of the matrix $\mathbf{K}$.

$$\begin{aligned}
&\int_{\mathbb{R}^2} \Big[\mathfrak{B}(\mathbf{y}) - \tau(\mathbf{y})^\top \mathbf{k}\Big]^2 w(\mathbf{y}) \mathrm{d}\mathbf{y} \\
&= \int_{\mathbb{R}^2} \Big[\mathfrak{B}^2(\mathbf{y}) + [\tau(\mathbf{y})^\top \mathbf{k}]^2 - 2\mathfrak{B}(\mathbf{y})\tau(\mathbf{y})^\top \mathbf{k}\Big] w(\mathbf{y}) \mathrm{d}\mathbf{y} \\
&= \int_{\mathbb{R}^2} \mathfrak{B}^2(\mathbf{y}) w(\mathbf{y}) \mathrm{d}\mathbf{y} + \int_{\mathbb{R}^2} \Big[k_{12}^2 \tau_1^2 + k_{21}^2 \tau_2^2 + 2k_{12}k_{21}\tau_1\tau_2\Big] w(\mathbf{y}) \mathrm{d}\mathbf{y} \\
&- 2\int_{\mathbb{R}^2} \mathfrak{B}(\mathbf{y}) \Big[k_{12}\tau_1 + k_{21}\tau_2\Big] w(\mathbf{y}) \mathrm{d}\mathbf{y} \\
&=: \mathfrak{A}(k_{12}, k_{21}).
\end{aligned}$$

[6] We write it as an approximation, with the symbol $\cong$, as the equality only holds for large n.

The first-order partial derivatives are:

$$\frac{\partial}{\partial k_{12}}\mathfrak{A} = 2k_{12}\int_{\mathbb{R}^2}\tau_1^2 w(\mathbf{y})\,\mathrm{d}\mathbf{y} + 2k_{21}\int_{\mathbb{R}^2}\tau_1\tau_2 w(\mathbf{y})\,\mathrm{d}\mathbf{y} - 2\int_{\mathbb{R}^2}\mathfrak{B}(\mathbf{y})\tau_1 w(\mathbf{y})\,\mathrm{d}\mathbf{y} \tag{96}$$

$$\frac{\partial}{\partial k_{21}}\mathfrak{A} = 2k_{21}\int_{\mathbb{R}^2}\tau_2^2 w(\mathbf{y})\,\mathrm{d}\mathbf{y} + 2k_{12}\int_{\mathbb{R}^2}\tau_1\tau_2 w(\mathbf{y})\,\mathrm{d}\mathbf{y} - 2\int_{\mathbb{R}^2}\mathfrak{B}(\mathbf{y})\tau_2 w(\mathbf{y})\,\mathrm{d}\mathbf{y} \tag{97}$$

and equating them to zero yields:

$$k_{12}\int_{\mathbb{R}^2}\tau_1^2 w(\mathbf{y})\,\mathrm{d}\mathbf{y} + k_{21}\int_{\mathbb{R}^2}\tau_1\tau_2 w(\mathbf{y})\,\mathrm{d}\mathbf{y} = \int_{\mathbb{R}^2}\mathfrak{B}(\mathbf{y})\tau_1 w(\mathbf{y})\,\mathrm{d}\mathbf{y} \tag{98}$$

$$k_{21}\int_{\mathbb{R}^2}\tau_2^2 w(\mathbf{y})\,\mathrm{d}\mathbf{y} + k_{12}\int_{\mathbb{R}^2}\tau_1\tau_2 w(\mathbf{y})\,\mathrm{d}\mathbf{y} = \int_{\mathbb{R}^2}\mathfrak{B}(\mathbf{y})\tau_2 w(\mathbf{y})\,\mathrm{d}\mathbf{y} \tag{99}$$

Solving (98) and (99) simultaneously will yield the required minimizing values for k_{12} and k_{21}. Let $\hat{\mathbf{K}}_{\mathrm{L}}$ denote the minimizer of ρ_w with respect to $\mathbf{K}$. Then we have

$$\begin{aligned}\hat{\mathbf{K}}_{\mathrm{L}} &= \hat{\mathbf{C}}\mathbf{B}_0^{-1}\\ &= \sqrt{n}(\hat{\mathbf{B}} - \mathbf{B}_0)\mathbf{B}_0^{-1}\\ &= \sqrt{n}(\hat{\mathbf{B}}\mathbf{B}_0^{-1} - \mathbf{I}).\end{aligned} \tag{100}$$

The limiting distribution of $\hat{\mathbf{K}}_{\mathrm{L}}$ is the same as the limiting distribution of $\sqrt{n}(\hat{\mathbf{B}}\mathbf{B}_0^{-1} - \mathbf{I})$, the latter being a convenient metric for estimating $\hat{\mathbf{B}}$. In a simulation study, using the computational algorithm[7] to estimate $\hat{\mathbf{B}}$, this form of the matrix $\hat{\mathbf{B}}\mathbf{B}_0^{-1} = \mathbf{I} + \mathbf{K}$ can be used to assess the accuracy of the estimate. In fact, $\hat{\mathbf{B}}\mathbf{B}_0^{-1} = \hat{\mathbf{B}}\mathbf{A}$ is the matrix product on which another similarity measure, called the *Amari Index* (cf. Section 4.3), is based on. With relation (100) in mind, we can measure the performance of an estimate $\hat{\mathbf{B}}$.

[7] Introduced in the next chapter and named MinDistICA.

3.3.2 Performance Measure for the Estimator

A very intuitive and effective approach, in light of the relations discussed above, is to examine the deviation of the matrix $\hat{\mathbf{B}}\mathbf{B}_0^{-1}$ from the identity matrix. The matrix product $\hat{\mathbf{B}}\mathbf{B}_0^{-1}$ should exhibit the following structure: its diagonal entries are approximately 1, and its off-diagonal entries are close to 0.

Define the matrix $\mathbf{P} \coloneqq \widehat{\mathbf{B}}\mathbf{B}_0^{-1}$, and a deviation matrix $\mathbf{D} \coloneqq \mathbf{P} - \mathbf{I}_2$. A natural performance measure is then the squared Frobenius norm of this deviation matrix:

$$d_{\mathcal{F}}(\widehat{\mathbf{B}}, \mathbf{B_0}) \coloneqq ||\mathbf{D}||_{\mathcal{F}}^2 = \text{tr}(\mathbf{D}^\top \mathbf{D}). \tag{101}$$

This quantity is non-negative and attains its minimum value of 0 when $\widehat{\mathbf{B}}\mathbf{B}_0^{-1} = \mathbf{I}_2$, indicating perfect source recovery. Of course, perfect recovery isn't possible, but the performance improves with an increase in sample size n as all associated results are asymptotic. In cases of successful source separation via the minimization algorithm (discussed in Chapter 4), $\mathbf{P}$ will approximate the identity matrix, and so tr$(\mathbf{D}^\top \mathbf{D})$ can be used for assessing the estimator $\hat{\mathbf{B}}$.

However, ICA inherently suffers from permutation and sign indeterminacies, meaning the order and signs of the recovered components are not uniquely identifiable. As a result, even when the ICA estimation is perfect, the matrix $\mathbf{P} = \widehat{\mathbf{B}}\mathbf{B}_0^{-1}$ may not be close to the identity matrix, and then the metric tr$(\mathbf{D}^\top \mathbf{D})$ may yield misleadingly large values. This, however, can be addressed by properly aligning the matrix $\mathbf{P}$ by exhaustively searching over all combinations of row permutations and sign flips. This corrected (properly aligned) matrix, say $\mathbf{P}_{\text{aligned}}$, will approximate the identity matrix, which can then be used to compute tr$(\mathbf{D}^\top \mathbf{D})$.

Example Consider the following unmixing matrix $\mathbf{B}$ and estimate $\hat{\mathbf{B}}$ in a 2-dimensional ICA problem obtained using the MinDistICA method:

$$\mathbf{B} = \begin{bmatrix} -0.25 & -0.375 \\ -0.75 & -0.625 \end{bmatrix}, \ \hat{\mathbf{B}} = \begin{bmatrix} 0.778 & 0.666 \\ 0.174 & 0.310 \end{bmatrix} \implies \mathbf{P} = \begin{bmatrix} -0.106 & -1.001 \\ -0.989 & 0.098 \end{bmatrix}.$$

Although $\mathbf{P}$ is far from the identity in raw form, it can be seen (by inspection) that it approximates the identity matrix under a sign flip in the first row, followed by a row swap. Once this optimal

choice of signed permutation is applied, the matrix thus obtained is

$$\mathbf{P}_{\text{aligned}} = \begin{bmatrix} 0.989 & -0.098 \\ 0.106 & 1.001 \end{bmatrix} \text{ and } \mathbf{D} = \mathbf{P}_{\text{aligned}} - \mathbf{I}_2$$

yields a much smaller value ($d_{\mathcal{F}} = 0.0209$), as compared to that obtained when using $\hat{\mathbf{B}}\mathbf{B}^{-1}$ directly ($d_{\mathcal{F}} = 4.0176$). In fact, across all 8 exhaustible row and sign permutations possible for $\mathbf{P}$, the *proper* alignment yields the lowest value of $d_{\mathcal{F}}$. Consequently, using the aligned matrix for $d_{\mathcal{F}}(\hat{\mathbf{B}}, \mathbf{B})$ provides a reliable indicator of the algorithm's recovery performance — the smaller the value obtained, the better is the estimation using our algorithm (discussed in Chapter 4).

3.3.3 Problem with Gaussian Components

Let X be a standard Gaussian random variable. Then,

$$\mathbb{E}\Big[X\mathbb{1}\{X < y\}\Big] = \int_{-\infty}^{y} \frac{x}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}\, \mathrm{d}x.$$

If $y \leq 0$, then, via the substitution $u = \dfrac{x^2}{2} \implies x\, \mathrm{d}x = \mathrm{d}u$, we have

$$\int_{-\infty}^{y} \frac{x}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}\, \mathrm{d}x = -\frac{1}{\sqrt{2\pi}} \int_{\frac{y^2}{2}}^{\infty} e^{-u}\, \mathrm{d}u = -\frac{1}{\sqrt{2\pi}} [e^{-u}]\Big|_{\frac{y^2}{2}}^{\infty} = \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}},$$

and if $y > 0$, we have, by the same substitution of variables,

$$\int_{-\infty}^{0} \frac{x}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}\, \mathrm{d}x + \int_{0}^{y} \frac{x}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}\, \mathrm{d}x = \frac{1}{\sqrt{2\pi}} + \frac{1}{\sqrt{2\pi}} [e^{-u}]\Big|_{0}^{\frac{y^2}{2}} = \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}.$$

So, if X is standard Gaussian with density function f, then

$$\mathbb{E}\Big[X\mathbb{1}\{X < y\}\Big] = f(y).$$

Suppose both X_1 and X_2 are independent standard Gaussian random variables with respective densitiy functions f_1 and f_2, then (93) becomes

$$-k_{12}f_1(y_1)f_2(y_2) - k_{21}f_2(y_2)f_1(y_1) = -(k_{12}+k_{21})f_1(y_1)f_2(y_2), \tag{102}$$

which makes separate estimation of the entries k_{12} and k_{21} impossible. In other words, only the sum of the entries, $k_{12}+k_{21}$, can be estimated.

3.3.4 General Case

The estimation procedure discussed for the 2-dimensional case can readily be used for the general case of d-dimensions, at least up to the point of obtaining the normal equations system for the parameters (reparametrized).

We follow a similar notation as was used in the 2-dimension case. Let $X_1, X_2, ..., X_d$ be independent random variables. Let the joint and marginal empirical distribution functions be

$$\hat{F}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{X_{1i} \leq x_1, ..., X_{di} \leq x_d\}, \text{ where } \mathbf{x} = (x_1, ..., x_d), \text{ and}$$
$$\hat{F}_j(x_j) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{X_{ji} \leq x_j\}, j = 1, 2, ..., d, \text{ respectively.}$$

The expression

$$\sqrt{n}\left[\hat{F}(\mathbf{x}) - \prod_{j=1}^{d}\hat{F}_j(x_j)\right] \xrightarrow{\text{d}} \mathcal{G}(\mathbf{x}),$$

with the covariance of the the process $\mathcal{G}$ being (54). This has already been shown in Lemma 3.1.1.

The reparametrization is the same. If we consider the true unmixing matrix[8] to be $\mathbf{B}_0$, we can introduce a new parameter matrix $\mathbf{K}$ by rewriting the parameter $\mathbf{B}$ as follows:

$$\mathbf{B} = \mathbf{B}_0 + \frac{1}{\sqrt{n}}\mathbf{C} = \left(\mathbf{I} + \frac{1}{\sqrt{n}}\mathbf{K}\right)\mathbf{B}_0 = \mathbf{D}_n\mathbf{B}_0,$$

[8] Recall that we have assumed the mixing matrix $\mathbf{A}$ to be invertible.

where

$\mathbf{K} := \mathbf{C}\mathbf{B}_0^{-1} = ((k_{ij}))_{i,j=1,...,d.}$, and

$$\mathbf{D}_n = \mathbf{I} + \frac{\mathbf{1}}{\sqrt{\mathbf{n}}}\mathbf{K} = ((d_{nij}))_{i,j=1,...,d.} \text{ with } d_{nij} = \begin{cases} 1 + \frac{k_{ij}}{\sqrt{n}}, & i = j, \\ \frac{k_{ij}}{\sqrt{n}}, & i \neq j. \end{cases}$$

For an arbitrary matrix $\mathbf{B}$, we have

$$\begin{aligned} \mathbb{1}\{\mathbf{b}_j^\top \mathbf{Y}_i \leq y_i,\ j = 1, ..., d\} &= \mathbb{1}\{\mathbf{B}\mathbf{Y}_i \leq \mathbf{y}\} \\ &= \mathbb{1}\{\mathbf{D}_n\mathbf{B}_0\mathbf{Y}_i \leq \mathbf{y}\} \\ &= \mathbb{1}\{\mathbf{D}_n\mathbf{X}_i \leq \mathbf{y}\} \\ &= \mathbb{1}\{\mathbf{d}_{nj}^\top\mathbf{X}_i \leq y_j, j = 1, ..., d\}, \end{aligned}$$

where $\mathbf{d}_{nj}^\top$ are the rows of the matrix $\mathbf{D}_n$.

Define the empirical distribution functions $\hat{F}_n(\mathbf{y})$ and $\hat{F}_{nj}(\mathbf{y}), j = 1, ..., d$, and the respective distribution functions $F_n(\mathbf{y})$ and $F_{nj}(\mathbf{y}), j = 1, ..., d$,

$$\hat{F}_n(\mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{\mathbf{d}_{n1}^\top\mathbf{X}_i \leq y_1, ..., \mathbf{d}_{nd}^\top\mathbf{X}_i \leq y_d\}, \quad \hat{F}_{nj}(\mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{\mathbf{d}_{nj}^\top\mathbf{X}_i \leq y_j\}, j = 1, ..., d,$$

$$F_n(\mathbf{y}) = \mathbb{P}\{\mathbf{d}_{n1}^\top\mathbf{X}_i \leq y_1, ..., \mathbf{d}_{nd}^\top\mathbf{X}_i \leq y_d\}, \quad F_{nj}(\mathbf{y}) = \mathbb{P}\{\mathbf{d}_{nj}^\top\mathbf{X}_i \leq y_j\}, j = 1, ..., d.$$

The process under discussion, written as a function of $\mathbf{y}$ and $\mathbf{B}$ can be then expressed as

$$\lim_{n\longrightarrow\infty} \sqrt{n}\Big[F_n(\mathbf{y} \mid \mathbf{B}) - F_n^{\perp}(\mathbf{y} \mid \mathbf{B})\Big] = \mathcal{G}(\mathbf{y}) + \lim_{n\longrightarrow\infty} \sqrt{n}\Big[F_n(\mathbf{y}) - \prod_{j=1}^{d} F_{nj}(y_j)\Big],$$

where only the second part depends on the parameter. Let

$$\mathbf{Z} = \mathbf{D}_n\mathbf{X} = \begin{bmatrix} \mathbf{d}_{n1}^\top \\ \vdots \\ \mathbf{d}_{nd}^\top \end{bmatrix} \mathbf{X} \implies \mathbf{X} = \mathbf{H}_n\mathbf{Z} = \begin{bmatrix} \mathbf{h}_{n1}^\top \\ \vdots \\ \mathbf{h}_{nd}^\top \end{bmatrix} \mathbf{Z}$$

where $\mathbf{D}_n^{-1} =: \mathbf{H}_n$.

The joint density function of $\mathbf{X} = (X_1, ..., X_d)$ is $\prod_{j=1}^{d} f_d(x_d)$, as $X_i, i = 1, ..., d$, are independent. Then, the density function of $\mathbf{Z}$ is given by $f_1(\mathbf{h}_{n1}^\top \mathbf{z}) \dots f_d(\mathbf{h}_{nd}^\top \mathbf{z}) \cdot \mathcal{J}$ with the Jacobian of transformation being $\mathcal{J} = |\det(\mathbf{D}_n^{-1})| = \dfrac{1}{|\det(\mathbf{D}_n)|} = \dfrac{1}{|k_n|} \longrightarrow 1$ as $n \longrightarrow \infty$. We proceed in the same way as in the earlier case of 2-dimensions, with focus on the second part which depends on the parameter (which via reparametrization is $\mathbf{K}$), and split it into two parts as follows:

$$\sqrt{n}\Big[F_n(\mathbf{y}) - \prod_{j=1}^{d} F_{nj}(y_j)\Big] = \sqrt{n}\Big[F_n(\mathbf{y}) - F(\mathbf{y})\Big] + \sqrt{n}\Big[\prod_{j=1}^{d} F_j(y_j) - \prod_{j=1}^{d} F_{nj}(y_j)\Big] =: \mathcal{P} + \mathcal{Q}, \tag{103}$$

where $F(\mathbf{y}) = \prod_{j=1}^{d} F_j(y_j)$ due to independence.

We will skim through most of the calculations here as they are nearly identical to what has already been discussed in the 2D case in Section 3.3.1. For the part $\mathcal{P}$, write

$$\begin{aligned}
F_n(\mathbf{y}) - F(\mathbf{y}) &= \int \cdots \int \Big[\frac{1}{k_n} f_1(\mathbf{h}_{n1}^\top \mathbf{x}) \dots f_d(\mathbf{h}_{nd}^\top \mathbf{x}) - \frac{1}{k_n} \prod_{j=1}^{d} f_j(x_j) + \frac{1}{k_n} \prod_{j=1}^{d} f_j(x_j) - \prod_{j=1}^{d} f_j(x_j)\Big] \mathbf{dx} \\
&= \int \cdots \int \frac{1}{k_n}\Big[f_1(\mathbf{h}_{n1}^\top \mathbf{x}) \dots f_d(\mathbf{h}_{nd}^\top \mathbf{x}) - \prod_{j=1}^{d} f_j(x_j)\Big] \mathbf{dx} \\
&+ \int \cdots \int \Big(\frac{1}{k_n} - 1\Big)\Big[\prod_{j=1}^{d} f_j(x_j)\Big] \mathbf{dx} \\
&=: \mathcal{P}_1 + \mathcal{P}_2.
\end{aligned}$$

The rows of the matrix $\mathbf{H}$ can be approximated using the Neumann series expansion for matrix inversion,

$$\begin{aligned}
\mathbf{D}_n^{-1} = \mathbf{H} &= \Big(\mathbf{I} + \frac{1}{\sqrt{n}}\mathbf{K}\Big)^{-1} \\
&= \mathbf{I} - \frac{1}{\sqrt{n}}\mathbf{K} + \frac{1}{n}\mathbf{K}^2 - \frac{1}{n^{3/2}}\mathbf{K}^3 + \cdots
\end{aligned}$$

provided the absolute value of the largest eigen value of $\frac{1}{\sqrt{n}}\mathbf{K} < 1$. In our case, this condition is

satisfied for large n. The entries of $\mathbf{H}$ are given by

$$h_{ij} = \delta_{ij} - \frac{1}{\sqrt{n}} k_{ij} + \frac{1}{n} \sum_{l} k_{il} k_{lj} - \cdots,$$

and thus for large n,

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_{n1}^{\top} \\ \vdots \\ \mathbf{h}_{nd}^{\top} \end{bmatrix} \approx \begin{bmatrix} 1 - \frac{1}{\sqrt{n}} k_{11} & -\frac{1}{\sqrt{n}} k_{12} & \cdots & -\frac{1}{\sqrt{n}} k_{1d} \\ -\frac{1}{\sqrt{n}} k_{21} & -1 - \frac{1}{\sqrt{n}} k_{22} & \cdots & -\frac{1}{\sqrt{n}} k_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{\sqrt{n}} k_{d1} & -\frac{1}{\sqrt{n}} k_{d2} & \cdots & 1 - \frac{1}{\sqrt{n}} k_{dd} \end{bmatrix}.$$

For notational convenience, denote $F_j(x_j)$ as F_j, $f_j(\mathbf{h}_{nj}^{\top}\mathbf{x})$ as g_j and $f_j(x_j)$ as f_j, $j = 1, ..., d$. Just like before, as $n \to \infty$, $g_j \to f_j, \forall j$. Then,

$$\mathcal{P}_1 = \int \cdots \int \frac{1}{k_n} \Big[g_1 \cdots g_d - f_1 \cdots f_d \Big] \mathbf{dx}$$

where $g_1 \cdots g_d - f_1 \cdots f_d = (g_1 - f_1) g_2 \cdots g_d + f_1 (g_2 - f_2) g_3 \cdots g_d + \cdots + f_1 \cdots f_{d-1} (g_d - f_d)$, i.e., a sum of d terms of the form $f_1 \cdots f_{j-1} (g_j - f_j) g_{j+1} \cdots g_d$, $j = 1, \cdots, d$. By Taylor's expansion, $g_j - f_j = (\mathbf{h}_{nj}^{\top}\mathbf{x}) f_j^{'} = -\frac{1}{\sqrt{n}} (\mathbf{k}_j^{\top}\mathbf{x}) f_j^{'}, j = 1, ..., d$, and hence, we get

$$\mathcal{P}_1 = -\frac{1}{k_n} \int \cdots \int \sum_{j=1}^{d} \Big[\frac{1}{\sqrt{n}} \mathbf{k}_j^{\top}\mathbf{x} f_1 \cdots f_{j-1} f_j^{'} g_{j+1} \cdots g_d \Big] \mathbf{dx}.$$

Therefore,

$$\begin{aligned}\lim_{n\to\infty}\sqrt{n}\mathcal{P}_1 &= \lim_{n\to\infty} -\frac{1}{k_n}\int\cdots\int\sum_{j=1}^{d}\Big[\mathbf{k}_j^\top\mathbf{x}f_1\cdots f_{j-1}f_j^{'}g_{j+1}\cdots g_d\Big]\mathbf{dx}\\ &= -1\Big[\sum_{j=1}^{d}k_{jj}[y_jf_j-F_j]F_1\cdots F_{j-1}F_{j+1}\cdots F_d+\sum_{\substack{i=1\\i\neq j}}^{d}\sum_{j=1}^{d}k_{ij}f_j[\int_{-\infty}^{y_i}x_if_i\,\mathrm{d}x_i]\prod_{\substack{l=1\\l\neq i,j}}^{d}F_l\Big]\\ &= -1\Big[\sum_{j=1}^{d}k_{jj}[y_jf_j]F_1\cdots F_{j-1}F_{j+1}\cdots F_d+\sum_{\substack{i=1\\i\neq j}}^{d}\sum_{j=1}^{d}k_{ij}f_j[\int_{-\infty}^{y_i}x_if_i\,\mathrm{d}x_i]\prod_{\substack{l=1\\l\neq i,j}}^{d}F_l\Big]\\ &\quad+F_1\cdots F_d\sum_{j=1}^{d}k_{jj}.\end{aligned}$$

The last term will cancel out with the other term, viz. $\lim_{n\to\infty}\mathcal{P}_2$, as we show next. Note that

$$\det(\mathbf{D}_n)=k_n\approx 1+\frac{1}{\sqrt{n}}\mathrm{tr}(\mathbf{K})=1+\frac{1}{\sqrt{n}}\sum_{i=1}^{d}k_{ii},$$

and hence,

$$\frac{1}{k_n}-1=-\frac{1}{\sqrt{n}}\frac{\mathrm{tr}(\mathbf{K})}{k_n}\implies\lim_{n\to\infty}\sqrt{n}\Big(\frac{1}{k_n}-1\Big)=-\mathrm{tr}(\mathbf{K})\cdot\lim_{n\to\infty}\frac{1}{k_n}=-\mathrm{tr}(\mathbf{K}).$$

Thus,

$$\lim_{n\to\infty}\sqrt{n}\mathcal{P}_2=\lim_{n\to\infty}\sqrt{n}\Big(\frac{1}{k_n}-1\Big)\int\cdots\int f_1\cdots f_d\,\mathbf{dx}=-\mathrm{tr}(\mathbf{K})F_1\cdots F_d=-F_1\cdots F_d\sum_{j=1}^{d}k_{jj}.$$

So we have

$$\begin{aligned}&\lim_{n\to\infty}\sqrt{n}\Big[F_n(\mathbf{y})-F(\mathbf{y})\Big]\\ &=-1\Big[\sum_{j=1}^{d}k_{jj}y_jf_jF_1\cdots F_{j-1}F_{j+1}\cdots F_d+\sum_{\substack{i=1\\i\neq j}}^{d}\sum_{j=1}^{d}k_{ij}f_j[\int_{-\infty}^{y_i}x_if_i\,\mathrm{d}x_i]\prod_{\substack{l=1\\l\neq i,j}}^{d}F_l\Big].\end{aligned}\tag{104}$$

For part $\mathcal{Q}$ of (103), we again use the following convenient change in notation: denote $F_{nj}(y_j)$

as G_j and $F_j(y_j)$ as just F_j. Then,

$$\begin{aligned}\prod_{j=1}^{d} F_{nj}(y_j) - \prod_{j=1}^{d} F_j(y_j) =& G_1 \cdots G_d - F_1 \cdots F_d \\ =& (G_1 - F_1)G_2 \cdots G_d + F_1(G_2 - F_2)G_3 \cdots G_d + \cdots \\ &+ F_1 \cdots F_{d-2}(G_{d-1} - F_{d-1})G_d + F_1 \cdots F_{d-1}(G_d - F_d).\end{aligned}$$

Now, for any of the terms of the form $G_j - F_j$, we have

$$\begin{aligned}G_j - F_j =& \int_{-\infty}^{y_j} \int \cdots \int \frac{1}{k_n} g_1 \cdots g_d - f_1 \cdots f_d \, \mathbf{dx} \\ =& \mathcal{P} \mid_{y_j \neq \infty, y_i = \infty \forall i \neq j = 1, \ldots, d.} =: \mathcal{P}_j^*.\end{aligned}$$

Therefore

$$\begin{aligned}\lim_{n \to \infty} \sqrt{n}\mathcal{Q} =& \lim_{n \to \infty} \sum_{j=1}^{d} \sqrt{n}\mathcal{P}_j^* \prod_{\substack{i=1 \\ i \neq j}}^{d} F_i \\ =& k_{11} y_1 f_1 F_2 \cdots F_d + k_{22} y_2 F_1 f_2 F_3 \cdots F_d + \cdots + k_{dd} y_d F_1 \cdots F_{d-1} f_d. \qquad (105)\end{aligned}$$

Combining (104) and (105), we obtain

$$\lim_{n \to \infty} \sqrt{n} \left[F_n(\mathbf{y}) - \prod_{j=1}^{d} F_{nj}(y_j) \right] = -1 \left[\sum_{\substack{i=1 \\ i \neq j}}^{d} \sum_{j=1}^{d} k_{ij} f_j \int_{-\infty}^{y_i} x_i f_i \, \mathrm{d}x_i \prod_{\substack{l=1 \\ l \neq i,j}}^{d} F_l \right]. \qquad (106)$$

Note that the above expression is devoid of the diagonal entries of the matrix $\mathbf{K}$, and hence, as in the 2D case, we can assume them to be 0. To express it in a more compact form, define:

$\text{vec}(\mathbf{K}) \coloneqq d^2 \times 1$ vector obtained by stacking columns of $\mathbf{K}$,

$$\mathbf{f} \coloneqq \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_d \end{pmatrix}, \quad \mathbf{E} \coloneqq \begin{pmatrix} \mathcal{E}_1(y_1) \\ \mathcal{E}_2(y_2) \\ \vdots \\ \mathcal{E}_1(y_1) \end{pmatrix}, \text{ where } \mathcal{E}_i(y_i) = \int_{-\infty}^{y_i} x_i f_i \, \mathrm{d}x_i,$$

$F^* \coloneqq d^2 - d$ dimension vector with entries $\prod_{\substack{l=1 \\ l \neq i,j}}^{d} F_l$ aligned with the same index order as $\text{vec}(\mathbf{K})$,

$\text{vec}(\mathbf{f}\mathbf{E}^\top) \coloneqq d^2$ dimension vector obtained by stacking columns of the $d \times d$ matrix $\mathbf{f}\mathbf{E}^\top$.

Now, without loss of generality, we can assume $k_{ii} = 0$ for all $i = 1, \ldots, d$. Let

$$\begin{aligned} \mathbf{k} &\coloneqq \text{ vector obtained from } \text{vec}(\mathbf{K}) \text{ by removing the } i = j \text{ terms}, \\ \mathbf{q} &\coloneqq \text{ vector obtained from } \text{vec}(\mathbf{f}\mathbf{E}^\top) \text{ by removing the } i = j \text{ terms}. \end{aligned}$$

Both $\mathbf{k}$ and $\mathbf{q}$ thus are $d^2 - d$ dimension vectors. Then (106) can be expressed as

$$\mathcal{L} \coloneqq -1 \left[\mathbf{k}^\top (\mathbf{q} \odot F^*) \right], \tag{107}$$

where $\odot$ denotes Kronecker product. One can now rewrite

$$n\rho_w \approx \int_{\mathbb{R}} \left[\mathcal{G}(\mathbf{y}) - \mathbf{k}^\top (\mathbf{q} \odot F^*) \right]^2 w(\mathbf{y}) \mathrm{d}y. \tag{108}$$

This expression is non-negative and the minimization problem boils down to least squares, as before in the 2D case (Section 3.3.1). Taking derivatives with respect to each $k_{ij},\ i, j = 1, \ldots, d;\ i \neq j$, and equating each to 0 will yield $d^2 - d$ square system of equations. Solving those would yield the minimizer values for the matrix $\mathbf{K}$.

Of course, this would not be analytically solvable, and one would have to utilize a numerical computation method. This is discussed in the next chapter.

Chapter 4

A Gradient Descent Estimator for the Unmixing Matrix

Any independent component analysis problem ultimately seeks to recover the sources from the mixtures. To do so, we proposed that we first estimate the unmixing matrix $\mathbf{B}$, and use it to "unmix" the observations, thereby obtaining an estimate of the sources. Since our approach is basically a "minimization of an objective function" problem, we use a simple yet very effective method to estimate unmixing matrix $\mathbf{B}$ called the *Gradient Descent* (GD) Algorithm. In this chapter, we develop a working algorithm that uses GD to find an estimate $\hat{\mathbf{B}}$ of the unmixing matrix $\mathbf{B}$, and use it to recover the sources from the observations. We compare the results with FastICA to gauge the performance and efficacy on the same data. Further, we make some observations about the estimate, based on the ideas presented at the end of Section 3.3.1. We begin this chapter with an introduction to Gradient Descent Algorithm.

4.1 Gradient Descent Algorithm

Gradient Descent is a first-order optimization algorithm (Polyak, 2010) widely used in statistics and machine learning. The algorithm minimizes an objective function by iteratively adjusting the model's parameters in the direction of steepest descent, as indicated by the negative gradient[1] of the

[1] In contrast, moving in the direction of the gradient leads to the maximization of a function and the procedure is then termed Gradient Ascent.

said objective function.

The origins of Gradient Descent can be traced back to the mid 19th century, and widely credited to Cauchy (1847). Since then, it has gained considerable traction with expansion in research in machine learning during the 20th century. Over the decades, it has undergone extensive development, leading to modern variants that address computational efficiency, convergence speed, and model generalization. It is now ubiquitous in optimization tasks because of its adaptability and efficiency in solving optimization problems.

4.1.1 Principle

For a minimization problem, gradient descent algorithms operates on a very simple premise: to iteratively update the model parameters in the direction that reduces the objective function. In other words, the parameters are updated in the opposite direction of the gradient of the objective function.

Given a differentiable objective function $C(\theta)$ of the parameter θ, the update rule for a single iteration is expressed as

$$\theta_{k+1} = \theta_k - \eta_k \nabla C(\theta_k). \tag{109}$$

The parameter value θ_k represents the model parameters at iteration k. The hyper-parameter η_k (> 0) involved in the iteration step is called the *learning rate*. It scales the gradient ∇C and controls the step size of each iteration. The learning rate η_k can be fixed as a constant, but in practice, it is usually taken to be dynamic, meaning that its value changes depending on the step. Intuitively, in the early steps, we would want to descend towards the minimum fast, so larger values of η_k are preferred, while in later steps, as the algorithm nears the (local) minimum, smaller values of η_k ensure the step does not overshoot, which may lead to oscillation around the minimum or even divergence. In our estimation algorithm, we use a specialized adaptive GD algorithm called *Adam* to dynamically change the parameter in each iteration.

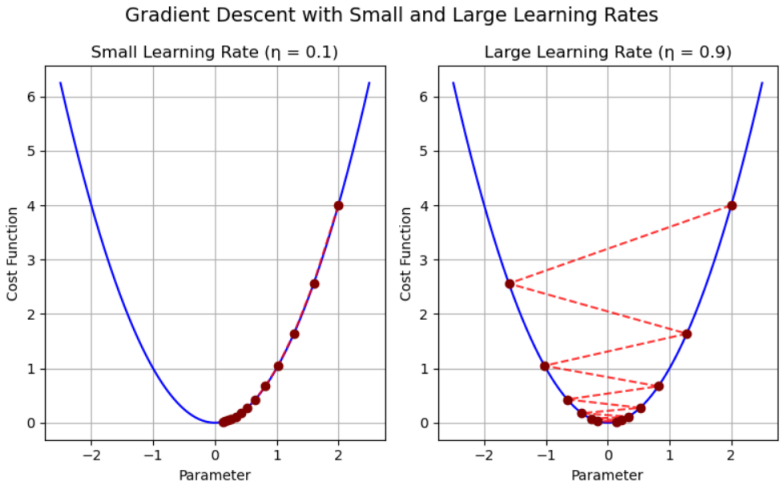

Figure 4.1: The difference in convergence based on the choice of learning rate.

4.1.2 Learning Rate and Convergence

The learning rate (also referred to as step size) is the size of the steps that are taken to reach the minimum. It is denoted by η in this work. It is usually taken as a small value (though the term small is subjective in a mathematical setting), and in most situations, it is evaluated and updated based on the behavior of the objective function. A low value of the learning rate results in smaller step sizes making it more precise at the cost of efficiency as it takes more time and steps to reach the minimum. Conversely, higher learning rates have larger steps but at the risk of overshooting the minimum. In extreme cases, it might even not converge. Figure 4.1 shows a simplistic illustrative comparison of the paths followed by GDA for small and large learning rates. There are no problems overshooting in this single (global) minimum example, and even if one jumps over the minimum, the gradient will eventually return toward it. So long as the learning rate is not too large to cause divergence, the algorithm will oscillate and eventually converge to the minimum. However, overshooting can be problematic in the case of a function with multiple local minima. It can lead to a jump from out of a local minimum's trough into a completely different region. It might bypass a good (lower) local minimum and land in a worse minimum or even on a flat or diverging path.

The choice of the learning rate greatly influences the convergence of Gradient Descent algorithms. They also include a limit to the maximum number of steps to iterate. A small learning

rate which converges slowly may cause the algorithm to reach this step limit before convergence. Whereas, if the learning rate is too large the algorithm may not converge to the optimal point (and jump around) or even cause it to diverge completely. In such cases, it would reach the iteration limit, and yield a result that may be far from any local minimum. A simple and intuitive workaround is to set the learning rate dynamically, allowing it to adapt as the iterations progress, as done in (109) with η_k changing with every iteration k.

Under the assumptions of convexity and continuity (and particular choices of the learning rate), the convergence of the objective function C can be guaranteed, at least to a local minimum. For convex functions, gradient descent can find the global minimum. However, for non-convex functions, finding the global minimum can be a struggle. The fact is, when the slope of the objective function is at (or very close to) zero, the model stops learning. Or simply put, the algorithm decides it has attained convergence. However, local minima and saddle points also yield this (close to) zero slope. The shape of a local minima is similar to the global minimum, where the slope of the objective function increases on either side of the point. While, saddle[2] points, are flat regions where the gradient is zero, neither maximizing nor minimizing the function. It can severely slow down convergence, or worse, trap the algorithm. These issues hinder efficient and reliable optimization and require adaptive methods to overcome them.

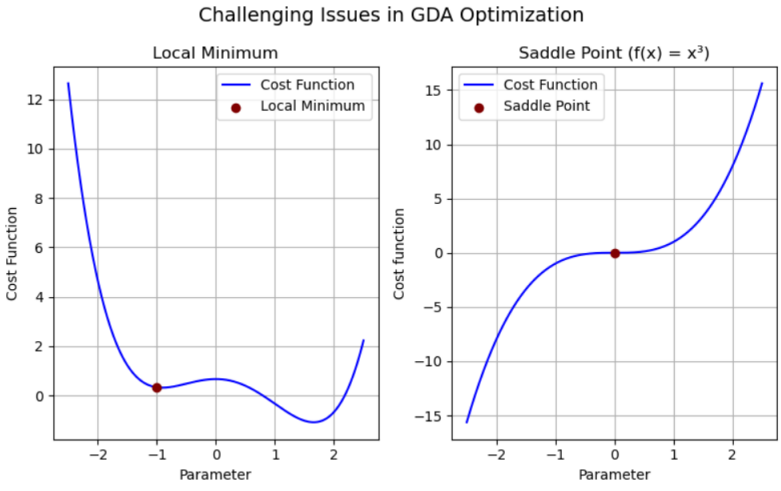

Figure 4.2: Scenarios where GDA might converge at a suboptimal value.

[2]It's shaped like a horse's saddle, earning it the name.

4.1.3 General Algorithm

The general form of the Gradient Descent Algorithm is straightforward and the basic framework is outlined below in Algorithm 7.

The objective function $C(\theta)$ is assumed to be convex (or locally convex near the parameter) and continuous. To start the iterative procedure, the parameter θ is assigned an initial value. It is initialized randomly or by using heuristic methods. The learning rate is usually advised to be set as dynamic, allowing it to change at each iteration. As explained before, earlier steps can have larger step sizes to accelerate the algorithm towards the solution, but as it nears the optimizer, the step sizes need to be smaller. This dynamic nature of the learning rate can be set quite simply by multiplying the learning rate choice η_k by a fraction (e.g. $0.9,\ 0.99$, etc.) at the end of each iteration. This also reduces the number of steps required for the convergence more often than not, allowing the algorithm to attain the convergence criterion instead of stopping at the maximum allowed iterations. Of course, the iteration limit can be set at an abnormally high value to ensure convergence, but that is very inefficient, especially when the suggested idea (or other means) can ensure convergence more effectively. There are also other optimization techniques one can look into to enhance the performance of GDA, one such being the Adam optimizer which we will be implementing. All the initial requirements are discussed further in the nextSectionwhere we apply GDA to our setting.

Algorithm 7 Gradient Descent Algorithm (GDA)

Require: Objective function $C(\theta)$, initial parameter θ_0, learning rate η, stop criterion (max iterations M and the convergence criterion).

1: Initialize $\theta = \theta_0$
2: **while** stop criterion not met **do**
3: Compute gradient: $\nabla_\theta C(\theta_k)$
4: Update parameter: $\theta_{k+1} = \theta_k - \eta_k \cdot \nabla_\theta C(\theta_k)$
5: **end while**
6: **return** θ_k

4.1.4 Enhancing GDA — Adam Optimizer

Adam (Adaptive Moment Estimation), introduced by Kingma and Ba (2015), can be applied as an optimization algorithm that builds on the standard Gradient Descent technique. When introducing the algorithm, the authors list certain benefits of using Adam on optimization (even non-convex) problems, like easy implementation, computation and memory resource efficiency, invariance under diagonal rescaling of the gradients, etc. It is well-suited for problems with large data (in terms of observations or parameters or both).

Adam combines the strengths of two popular methods — Adaptive Gradient (AdaGrad) and Root Mean Square Propagation (RMSProp), reaping the benefits of both. AdaGrad adjusts the learning rate for each parameter based on the history of past gradients. Adam implements this via first moment estimates. This introduces a momentum term m_k that smooths the update direction and accelerates convergence speed. RMSProp modifies the learning rate based on the magnitude of recent gradients to normalize updates, preventing them from becoming too large or too small. This is included in Adam through the second moment estimate v_k.

In other words, Adam uses an exponentially decaying average of the past gradients and squared gradients to smooth the update direction, reducing oscillations and improving convergence. The parameters β_1 and β_2 control the decay rates of these moving averages. The initial value of the moving averages and the tuning parameters β_1 and β_2 values are usually set close to 1.0 (recommended) resulting in a bias (towards 0) of moment estimates. Adam includes bias-correction terms to ensure that the first and second-moment estimates are unbiased during the initial iterations.

Tuning paremeters:

i. η: the learning rate is the value by which the weights are updated.

ii. β_1: the exponential decay rate for the first moment estimates.

iii. β_2: the exponential decay rate for the second-moment estimates.

iv. ϵ: it is a failsafe to prevent any division by zero in the implementation.

Algorithm 8 Gradient Descent with Adam Optimizer

Require: Objective function $C(\theta)$, learning rate η, initial parameters θ_0, stopping criterion, hyper-parameters $\beta_1, \beta_2, \epsilon$.

Initialize $\theta = \theta_0$, $m_0 = 0$, $v_0 = 0$, $k = 0$

while stopping criterion not met **do** ▷ All operations are element-wise for vectors.

  $k = k + 1$

  Compute gradient: $g_k = \nabla_\theta C(\theta_{k-1})$

  Update biased first moment estimate: $m_k = \beta_1 m_{k-1} + (1 - \beta_1) g_k$

  Update biased second moment estimate: $v_k = \beta_2 v_{k-1} + (1 - \beta_2) g_k^2$

  Compute bias-corrected first moment: $\hat{m}_k = \frac{m_k}{1-\beta_1^k}$

  Compute bias-corrected second moment: $\hat{v}_k = \frac{v_k}{1-\beta_2^k}$

  Update parameters: $\theta_k = \theta_{k-1} - \eta \cdot \frac{\hat{m}_k}{\sqrt{\hat{v}_k} + \epsilon}$

end while

return θ_k (resulting parameter)

4.2 Implementation

In this section we discuss how the Gradient Descent Algorithm is implemented in our setup, with further adaptive optimization using Adam for computational efficiency and accuracy. Consider the ICA problem with d independent sources $\mathbf{X} = (X_1, \ldots, X_d)$ and d observations $\mathbf{Y} = (Y_1, \ldots, Y_d)$; with the sources mixed by a matrix $\mathbf{A}$. Assuming $\mathbf{A}$ is invertible, with the inverse (the *unmixing matrix*) being $\mathbf{B}$, we have the relation (6).

4.2.1 Objective function

To implement a Gradient Descent search for the estimate as per our proposal (cf. Section 1.7), we rewrite the metric defined in (20) as a function of the unmixing matrix $\mathbf{B}$ to serve as the objective function:

$$\rho_w := \rho_w(\mathbf{B}) = \int_{\mathbb{R}^d} \Big[F_n(\mathbf{y} \mid \mathbf{B}) - F_n^{\perp}(\mathbf{y} \mid \mathbf{B})\Big]^2 w(\mathbf{y})\, \mathrm{d}\mathbf{y}. \tag{110}$$

4.2.2 Update Rule

The gradient descent update rule at iteration[3] $k,\ k = 0, 1, \ldots, M$, is given

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \eta_k \nabla \rho_w(\mathbf{B}_k) \tag{111}$$

with gradient $\nabla \rho_w(\mathbf{B}_k)$ and learning rate η_k. $\mathbf{B}_0$ is an initial choice of the matrix $\mathbf{B}$.

The initial choice for the parameter $\mathbf{B}_0$ is intuitive and sample-based. Note that in the ICA setup, we have

$$\mathbf{y} = \mathbf{A}\mathbf{x} \Longleftrightarrow \mathbf{x} = \mathbf{B}\mathbf{y},$$

with $\mathbb{V}(\mathbf{X}) = \mathbf{I}$, by assumption. As such, $\mathbb{V}(\mathbf{Y}) = \mathbf{A}\mathbf{A}^\top$, and therefore, $\mathbf{I} = \mathbf{B}\mathbb{V}(\mathbf{Y})\mathbf{B}^\top = \mathbf{B}\mathbf{A}\mathbf{A}^\top\mathbf{B}^\top$. Pre- and post-multiplying both sides by $\mathbf{B}^{-1}$ and $\mathbf{B}^{-\top}$, respectively, we have

$$\begin{aligned} & \mathbf{B}^{-1}\mathbf{B}^{-\top} = \mathbf{B}^{-1}\mathbf{B}\mathbf{A}\mathbf{A}^\top\mathbf{B}^\top\mathbf{B}^{-\top} \\ \Longrightarrow\ & (\mathbf{B}^\top\mathbf{B})^{-1} = \mathbf{A}\mathbf{A}^\top \\ \Longrightarrow\ & \mathbf{B}^\top\mathbf{B} = (\mathbf{A}\mathbf{A}^\top)^{-1}. \end{aligned} \tag{112}$$

where $\mathbf{B}^{-\top} := (\mathbf{B}^{-1})^\top$.

Since, $\mathbf{B}$ is non-singular, $\mathbf{B}^\top\mathbf{B}$ is symmetric and positive definite, and there exists[4] a matrix $\mathbf{R}$, called the square root of $\mathbf{B}^\top\mathbf{B}$, such that $\mathbf{R}^2 = \mathbf{B}^\top\mathbf{B}$. Therefore, a logical choice for the initial value of the parameter is to take $\mathbf{B}_0 = (\mathbf{A}\mathbf{A}^\top)^{-\frac{1}{2}}$. In practice, $\mathbf{A}$ is unknown, and so, to initialize the matrix $\mathbf{B}$ for the GDA, we choose the data-based analogue of $\mathbf{A}\mathbf{A}^\top$, viz,

$$\mathbf{B}_0 = \mathbf{V}^{-\frac{1}{2}},\ \text{where } \mathbf{V} \text{ is the dispersion matrix of the observations } \mathbf{Y}.$$

[3] M being the maximum number of iterations that the algorithm will run.

[4] This is a standard result in Linear Algebra. For a detailed proof, refer to Tsumura (2020).

4.2.3 The MinDistICA Algorithm

The MinDistICA algorithm follows a structured process to estimate the unmixing matrix $\mathbf{B}$ and recover the independent components $\mathbf{X}$ from the observed mixtures $\mathbf{Y}$. The Adam optimized Gradient Descent Algorithm applied to our minimum distance approach to ICA (MinDistICA) is provided in Algorithm 9. Of course, these are simulation studies, and so the data $\mathbf{Y}$ has to be generated synthetically. We describe the major steps below.

Generating the Data

The observed data $\mathbf{Y}$ is synthesized by mixing two independent sources X_1 and X_2 using a predefined invertible mixing matrix $\mathbf{A}$ such that $\mathbf{Y} = \mathbf{AA}$, $\mathbf{X} = (X_1, X_2)$. The sources are drawn from different probability distributions (varies across examples used in this study) and are normalized to have zero mean and unit variance. It goes without saying that there is no loss of generality with standardizing the sources. These data mixtures $\mathbf{Y}$ serve as input for the algorithm. We stress again that in practical cases, only the observations are known; both the sources $\mathbf{X}$ and $\mathbf{A}$ are unknowns. The only assumption on the sources $\mathbf{X}$ are that they are independent and at most one of them can be Gaussian. For the mixing matrix $\mathbf{A}$, we only assume it is invertible, with the inverse being $\mathbf{B}$.

Computing the Metric ρ_w as a U-statistic

The objective of the algorithm is to minimize the metric ρ_w with respect to the parameter $\mathbf{B}$. It is, as shown before, a U-statistic comprised of three parts, each part involving integrals of the weight functions w_1 and w_2. These integrals are computed numerically for efficiency and stored using memoization[5] to avoid redundant calculations, thereby speeding consequent calculations.

[5] We use memoization to enhance efficiency during the computation process, cf. Abelson and Sussman (1996).

Gradient Computation

To minimize ρ_w, the gradient with respect to the unmixing matrix $\mathbf{B}$ has to be computed. The gradient can be calculated using finite-difference derivative approximations (cf.Chapter 8 of Nocedal and Wright (1999)), where each element of $\mathbf{B}$ is perturbed by a small value δ. This allows for the estimation of the partial derivatives of ρ_w with respect to $\mathbf{B}$, without the need of the explicit form of the derivative. Mathematically, this is performed element-wise, and the gradient of ρ_w with respect to $\mathbf{B}$ is approximated using central-differences: when computing the gradient with respect to the entry b_{ij} of the matrix $\mathbf{B}$, perturb b_{ij} by a small value δ (in all the simulated examples conducted in this work $\delta = 10^{-7}$ unless specified otherwise) compute ρ_w for $\mathbf{B} + \delta\mathbf{E}_{ij}$ and $\mathbf{B} - \delta\mathbf{E}_{ij}$, and calculate the approximate partial derivatives as:

$$\frac{\partial \rho_w}{\partial b_{ij}} \approx \frac{\rho_w(\mathbf{B} + \delta\mathbf{E}_{ij}) - \rho_w(\mathbf{B} - \delta\mathbf{E}_{ij})}{2\delta},$$

where $\mathbf{E}$ is a matrix with the $(i,j)^{\text{th}}$-entry as 1 and 0 elsewhere. This simple method provides an efficient route to computing the gradient.

Iterative Steps and Convergence

The iterative algorithm uses the Adam optimizer for the gradient descent, allowing the learning rate to dynamically change as the steps progress. The initial unmixing matrix $\mathbf{B}_0$, as discussed earlier, is chosen as the inverse square root of the dispersion matrix of the observations $\mathbf{y}$. The algorithm iteratively updates $\mathbf{B}$ over a maximum of M iterations, aiming to minimize ρ_w following an adaptive update rule. Adam has already been discussed in Section 4.1.4. The main steps are the same, as explained in the aforementioned section(cf. Steps 2–10 in Algorithm 8). Since we are working with a matrix, some clarifications need to be made. The initial moment estimates $\mathbf{M}_1$ and $\mathbf{M}_2$ are taken as zero matrices $\mathbf{0}$ of the same size as the parameter matrix $\mathbf{B}$. All operations followed in the iterative steps of the algorithm are done element-wise.

Apart from reaching the maximum number of permissible iterations M, the algorithm ideally

stops upon satisfying a predefined convergence criterion. The convergence criterion can be implemented in two ways. One would be to check the gradient at the k^{th} step,

$$\mathbf{G}_k = \nabla \rho_w(\mathbf{B}_{k-1}),$$

and stop when its magnitude is below a predefined threshold. The other (which we use) would be based on the value of the statistic ρ_w and its change over an iteration, comparing the effects of old and new unmixing matrix estimates. Mathematically, at the k^{th} step, the convergence criterion is defined as

$$|\rho_w(\mathbf{B}_{k+1}) - \rho_w(\mathbf{B}_k)| < \zeta, \tag{113}$$

where ζ is a small preset tolerance value. Once the stopping criterion is met, we obtain a matrix $\mathbf{B}_{\text{a}}$ that minimizes[6] ρ_w with respect to $\mathbf{B}$. However, since we have not made any provision to contain the dispersion of the estimated independent components obtained, they are always scaled. This is not a big issue and can be corrected with just one post-processing step.

Algorithm 9 Adam Optimized Gradient Descent applied to MinDistICA approach. $\mathbf{G}_k^{\odot 2} = \mathbf{G}_k \odot \mathbf{G}_k$ indicates the elementwise square.

Require: Observed mixtures $\mathbf{y}$, weight functions w_1 and w_2, learning rate η, stopping criterion.

1: Initialize parameter: $\mathbf{B} = \mathbf{B}_0$, weight functions: $w_1(u)$ and $w_2(u)$.
2: Set Adam parameters $\beta_1, \beta_2, \epsilon$.
3: Initialize moment estimates $\mathbf{M}_1 = \mathbf{0}$, $\mathbf{M}_2 = \mathbf{0}$.
4: **while** stop criterion not met **do** ▷ All operations on matrices are element-wise.
5: Compute gradient: $\mathbf{G}_k = \nabla \rho_w(\mathbf{B}_{k-1})$.
6: Update biased moments: $\mathbf{M}_1 = \beta_1 \mathbf{M}_1 + (1-\beta_1)\mathbf{G}_k, \quad \mathbf{M}_2 = \beta_2 \mathbf{M}_2 + (1-\beta_2)\mathbf{G}_k^{\odot 2}$.
7: Correct bias: $\hat{\mathbf{M}}_1 = \frac{1}{1-\beta_1^t}\mathbf{M}_1, \quad \hat{\mathbf{M}}_2 = \frac{1}{1-\beta_2^t}\mathbf{M}_2$.
8: Update $\mathbf{B}$: $\mathbf{B}_{k+1} = \mathbf{B}_k - \eta \frac{\hat{\mathbf{M}}_1}{\sqrt{\hat{\mathbf{M}}_2 + \epsilon}}$.
9: **end while**
10: **Return**: unmixing matrix $\mathbf{B}_k$.

[6] Since this is a numerical method, it would be more accurate to say approximately minimizes.

4.2.4 Post-processing

After the GDA returns $\mathbf{B}_{\text{a}}$, we can get a (scaled) estimate of the sources, i.e., $\hat{\mathbf{X}} = \mathbf{B}_{\text{a}}\mathbf{Y}$. This happens since we do not make any arrangements to contain the estimates' variance to unity. In most, if not all, established methods discussed in Section 1.6, or otherwise, the data $\mathbf{Y}$ is preprocessed to zero mean and unit variance, converting the search for the mixing matrix to the space of orthogonal matrices. Refer to Section 1.5 for details. As such, the MinDistGDA we have discussed thus far retains the "shape" but not the "scale" of the estimated Independent components, compared to the sources we generate to have unit variance in the simulation examples[7].

However, there is a very logical step we can apply at the end to solve this issue. Since the estimates obtained have non-unit variances but are independent, $\mathbb{V}(\hat{X}_1) = \nu_1$ and $\mathbb{V}(\hat{X}_2) = \nu_2$, where $\nu_1, \nu_2 \neq 1$, and $\mathbb{C}\text{ov}(\hat{X}_1, \hat{X}_2) = 0$. Intuitively, it makes sense just to scale the estimates by their standard deviations to convert them to unit variance. Mathematically, this can be illustrated as shown below:

Let the dispersion matrix of the estimated $\hat{\mathbf{x}}$ independent components be

$$\mathbb{V}(\hat{\mathbf{X}}) := \mathbf{V} = \begin{bmatrix} \nu_1 & 0 \\ 0 & \nu_2 \end{bmatrix}.$$

Consider the transformation from $\hat{\mathbf{Z}} = \mathbf{S}\hat{\mathbf{X}}$, such that

$$\mathbf{S} = \begin{bmatrix} \frac{1}{\sqrt{\nu_1}} & 0 \\ 0 & \frac{1}{\sqrt{\nu_2}} \end{bmatrix}.$$

Then, $\mathbb{V}(\hat{\mathbf{Z}}) = \mathbf{S}\mathbf{V}\mathbf{S}^\top = \mathbf{I}$. So, all that is required is to scale each estimated independent component by their standard deviations, or, in the matrix notation in the ICA setup, pre-multiply $\mathbf{B}_{\text{a}}$ by the scaling matrix $\mathbf{S}$ to get the final estimate of the unmixing matrix, i.e, $\hat{\mathbf{B}} = \mathbf{S}\mathbf{B}_{\text{a}}$. The independent components, thus obtained, using $\hat{\mathbf{B}}$ are the final estimates with unit variance.

[7] We provide the graphs for the first example showing the unscaled estimates for comparison.

4.3 Comparative Evaluation of MinDistICA and FastICA

To evaluate the performance of the proposed MinDistICA approach, we compare it against FastICA as a benchmark. For the purpose of quantifying and comparing the separation quality of our algorithm with that of FastICA, we employ 3 evaluation metrics: Amari Index, Signal-to-Interference Ratio (SIR) and the Average Correlation Coefficient. These metrics capture distinct aspects of the ICA performance — accuracy of the estimated unmixing matrix, signal separation fidelity and alignment with the "ground-truth" (true) components. Independent Components are not ordered or sign consistent across different ICA runs or even for the same data for two different methods. The metrics mentioned below are used as they are all permutation- and sign-invariant.

Amari Index

Originally introduced in Amari, Cichocki, and Yang (1995) as a performance metric in Blind Source Separation problems, the *Amari Index* (also called *Amari Distance*) is a similarity index[8] between two invertible matrices. It is now a widely used measure that quantifies how close an estimated unmixing matrix $\hat{\mathbf{B}}$ is to the true unmixing matrix $\mathbf{B}$. For two invertible matrices $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{d\times d}$, the Amari Index is defined as

$$d_{\mathcal{A}}(\mathbf{U},\mathbf{V}) := \frac{1}{2d(d-1)}\left[\sum_{i=1}^{d}\left(\frac{\sum_{j=1}^{d}p_{ij}}{\max(p_{ij})}-1\right)+\sum_{j=1}^{d}\left(\frac{\sum_{i=1}^{d}p_{ij}}{\max(p_{ij})}-1\right)\right], \tag{114}$$

where $\mathbf{P} = ((p_{ij}))_{i,j=1,\ldots,d} = \mathbf{U}^{-1}\mathbf{V}$. $d_{\mathcal{A}}$ is non-negative and attains 0 if and only if $\mathbf{U}^{-1}\mathbf{V}$ is a scale and permutation matrix. It penalizes deviation from the ideal permutation and scaling matrix structure. Values closer to 0 indicate better recovery, while values greater than 1 typically indicate substantial misalignment.

Signal-to-Interference Ratio

Signal-to-Interference Ratio (SIR) (Ma, Zheng, Wu, Yu, and Xiang (2022)) is another measure used to evaluate the performance of source separation techniques such as ICA. SIR compares the power of the desired signal component to the power of the interference component in the estimated signal, measured in decibels (dB). Mathematically, for a single estimated source $\hat{X}_i$ corresponding

[8] A similarity index is a real-valued function that quantifies similarity between two objects.

to true source X_i,

$$\text{SIR}_i := 10 \log_{10} \left(\frac{||\hat{X}_i||^2}{||X_i - \hat{X}_i||^2} \right), i = 1, \ldots, d, \tag{115}$$

and the mean SIR is given by

$$\text{SIR} = \frac{1}{d} \sum_{i=1}^{d} \text{SIR}_i. \tag{116}$$

It is evident from the definition that the larger the value, the better the separation effect — the target signal dominates and there is minimal contamination from other sources. Values below 0 dB suggest interference dominates the recovered source.

Average Absolute Correlation Coefficient

Average Absolute Correlation Coefficient (AACC) is a simple yet effective measure used to evaluate how well components extracted by one method correlate with components extracted by another method (or with ground truth sources, if available). Computation of AACC is pretty straightforward, and involves the following:

(1) Pairwise correlation matrix: Compute the absolute correlation coefficient between each pair of estimated components from the two methods (or correlation between the estimated components and the true sources), given by

$$R_{ij} = |\text{corr}(\hat{X}_i, \hat{Z}_j)|, i, j = 1, \ldots, d,$$

where $\hat{X}_i$ is the i^{th} component estimate obtained from MinDistICA, and $\hat{Z}_j$ is the j^{th} component estimate obtained from FastICA.

(2) Optimal pairing: Use an assignment algorithm, like the Hungarian algorithm (cf. Kuhn (1955)), to find the best one-to-one matching between components from $\hat{X}_i$ and $\hat{Z}_i$ that maximizes total correlation.

(3) Average correlation: Take the mean of the absolute correlation values for the matched pairs.

4.4 Simulation Studies

To assess the empirical performance of the proposed MinDistICA algorithm, we conduct a series of simulation studies. For every scenario, we compute the performance measure $d_{\mathcal{F}}$ (cf. Section 3.3.2) to express the validity of the estimation method in terms of a single quantity. In addition, we compare it against the well-established FastICA method, based on three widely used performance metrics in the Independent Component Analysis literature: the Amari Index (AI), the Signal-to-Interference Ratio (SIR), and the Average Absolute Correlation Coefficient (AACC). These metrics offer complementary perspectives on the quality of source recovery, with AI capturing the deviation from exact source separation, SIR quantifying the separation fidelity in terms of interference suppression, and AACC providing a measure of linear correspondence between estimated and true sources. After the general comparison, we provide two particular simulation studies in Example 4.4.1 and Example 4.4.2.

The simulations are designed to evaluate the performance across different types of source distributions. Three distinct settings are considered, each involving two components drawn from the following distributional pairs:

- Case 1: Uniform and Gaussian (Figure 4.4);
- Case 2: Gaussian and Exponential (Figure 4.5);
- Case 3: Laplace and Uniform (Figure 4.6).

In all cases, the source signals are generated with zero mean and unit variance to ensure comparability across settings. For each scenario, we simulate $N = 100$ independent datasets, each consisting of $n = 50$ observations. A random mixing matrix is generated for each dataset, subject to a check for invertibility to ensure adherence to the invertibility assumption for the ICA model. Both FastICA and MinDistICA are then applied to the mixed observations, and their performance is evaluated using the three aforementioned metrics. The simulation results are summarized using boxplots for each performance measure, allowing for a clear visualization of the distribution of outcomes across the datasets.

Across all three simulation scenarios — Uniform vs Gaussian, Gaussian vs Exponential, and

Laplace vs Uniform — MinDistICA demonstrates strong source separation capabilities. This is reflected in the values of $d_{\mathcal{F}}$, which is summarized below and visualized together in Figure 4.3.

Table 4.1: Summary statistics for $d_{\mathcal{F}}$ under the different scenarios.

Scenario	**Mean**	**Std**	**Min**	**Max**	**Median**
Case 1	0.259	0.179	0.006	0.648	0.233
Case 2	0.284	0.196	0.0155	0.729	0.250
Case 3	0.262	0.179	0.008	0.677	0.225

Table 4.1 reports summary statistics for the performance metric $d_{\mathcal{F}}$, which quantifies the accuracy of source separation achieved by the MinDistICA algorithm. Each scenario corresponds to a distinct simulation setup, with varying source distributions, over $N = 100$ simulations each (300 in total) with sample sizes of $n = 50$. Lower values of $d_{\mathcal{F}}$ indicate better alignment between the estimated unmixing matrix and the true mixing matrix. Across all three cases, the mean and median values remain low (approximately 0.25), suggesting that MinDistICA consistently delivers excellent source recovery. The similarity of the statistics across cases implies stable and robust performance across different simulation scenarios. However, the maximum values in each case range between 0.64 and 0.73, indicating that in a small number of runs the algorithm produced less accurate estimates — potentially due to local minima, sensitivity to initial choice, or data-specific challenges. This highlights that while MinDistICA is generally robust and accurate, it can occasionally yield suboptimal results. This warrants further investigation, in future follow-up studies, into the conditions under which its performance deteriorates.

In terms of its performance with that of FastICA, it is comparable. With respect to the mixing matrix recovery, as assessed by the Amari Index, both algorithms perform similarly across the 3 cases. However, MinDistICA often demonstrates greater robustness, particularly in the exponential and Laplace configurations, where it exhibits less variability and more stable recovery. The Signal-to-Interference Ratio (SIR) results reveal a more pronounced difference: MinDistICA consistently achieves higher median SIR values and upper bounds are higher.. This indicates that MinDistICA is more effective at separating underlying source signals. Finally, the Average Absolute Correlation Coefficient (AACC) shows that both methods produce source estimates that are highly correlated

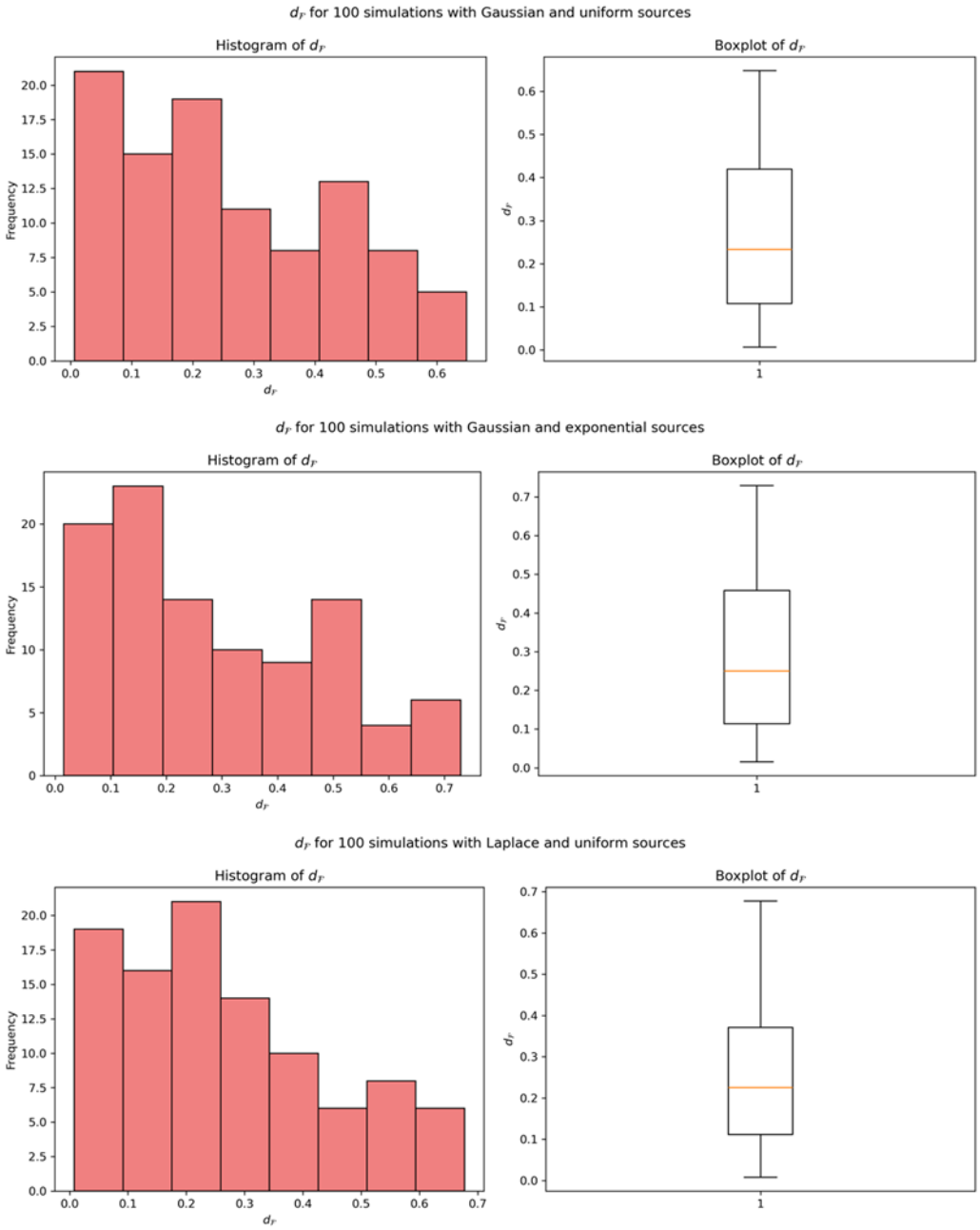

Figure 4.3: Histogram and Scatterplots of $d_{\mathcal{F}}$ for the 3 scenarios of simulations; Case 1: Uniform and Gaussian, Case 2: Gaussian and Exponential, Case 3: Laplace and Uniform.

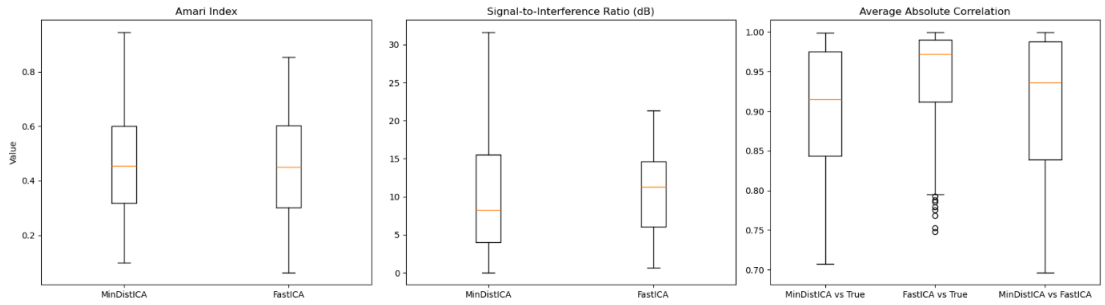

Figure 4.4: Boxplots for the performance metrics — Uniform & Gaussian sources.

with the ground truth, with FastICA displaying slightly higher median values and tighter consistency in most cases. Nonetheless, MinDistICA is competitive even in scenarios with sample sizes of $n = 50$ (as was done for the simulations). For larger sample sizes ($n = 75, 100$, etc), the algorithm's accuracies improved, however, due to the technical difficulty in computing the U-statistic $N = 100$ times in each case, we settled for $n = 50$ for the simulations.

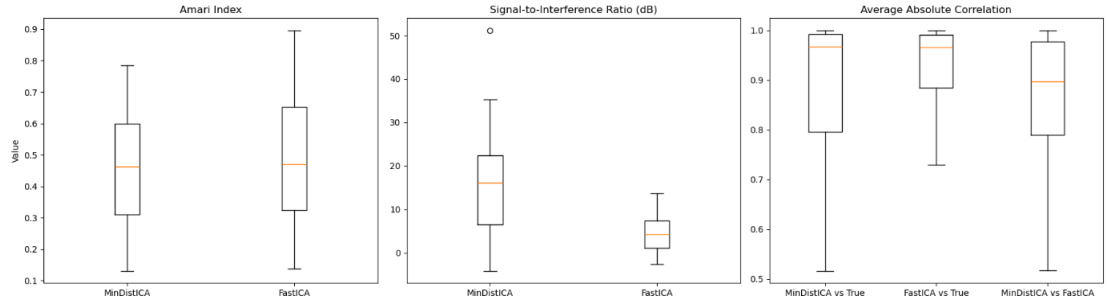

Figure 4.5: Boxplots for the performance metrics — Exponential & Gaussian sources.

4.4.1 Example 1

For the first example, the steps and results are explained in greater detail. We also consider a smaller sample size for ease of visual comparison between the sources (used to generate the data) and the estimated independent components obtained at the end of the MinDistICA approach.

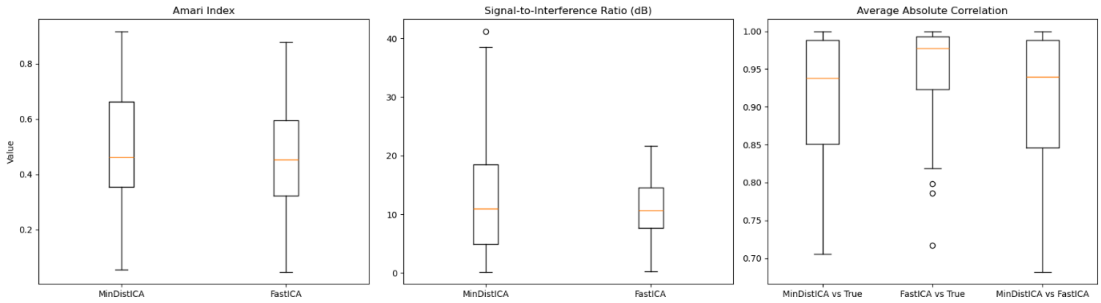

Figure 4.6: Boxplots for the performance metrics — Laplace & Uniform sources.

Independent Components

The independent components X_1 and X_2 are generated separately from distinct distributions to satisfy the assumption of *statistical independence*.

i. The variable X_1 is sampled from a Laplace distribution with mean 0 and a scale parameter of $\frac{1}{\sqrt{2}}$.

ii. The variable X_2 is sampled from a uniform distribution defined over the interval $(-\sqrt{3}, \sqrt{3})$. This interval is specifically chosen to ensure that the distribution has a zero mean and unit variance..

The parameters for both sampling distributions are adjusted to ensure zero mean and unit variance of the sources. Additionally, we choose the sources to have contrasting kurtoses[9] — the Laplace distribution is *leptokurtic*, while the uniform distribution in *platykurtic*. This is chosen for the first example to compare the performance against a kurtosis-based FastICA algorithm.

Mixing Matrix and Observed Data

The observed data $\mathbf{y}$ is obtained by linearly mixing the independent components $\mathbf{x} = (X_1, X_2)^\top$ using the mixing matrix

$$\mathbf{A} = \begin{bmatrix} 8 & -12 \\ -2 & 15 \end{bmatrix}.$$

[9]Leptokurtic distributions, such as the Laplace distribution, exhibit sharp peaks and heavy tails, resulting in a higher kurtosis value compared to a Gaussian distribution. Platykurtic distributions, such as the uniform distribution, have flat peaks and light tails, with a smaller kurtosis value than a Gaussian distribution.

with the unmixing matrix being

$$\mathbf{B} = \mathbf{A}^{-1} = \begin{bmatrix} 0.15625 & 0.12500 \\ 0.02083 & 0.08333 \end{bmatrix}.$$

The observed data is computed as $\mathbf{Y} = \mathbf{AX}$, where $\mathbf{Y} = (Y_1, Y_2)^\top$, represents the observations

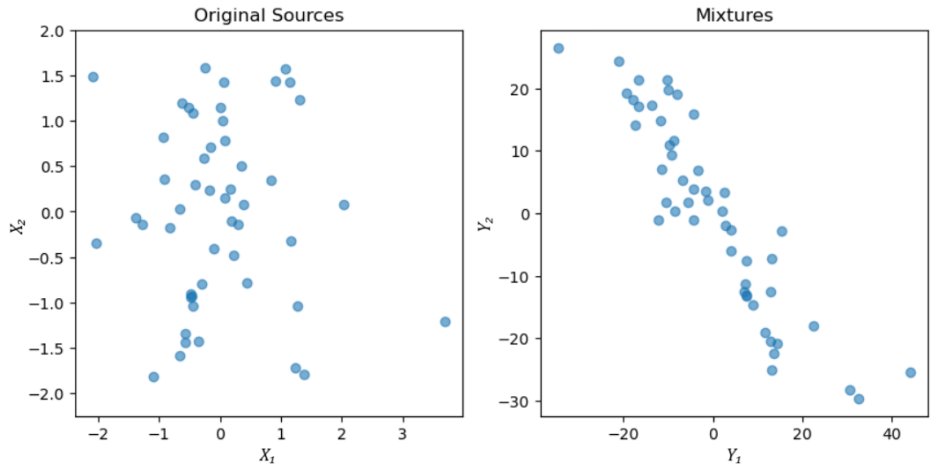

Figure 4.7: Scatterplots of the sources $\mathbf{X}$ and the mixtures $\mathbf{Y}$.

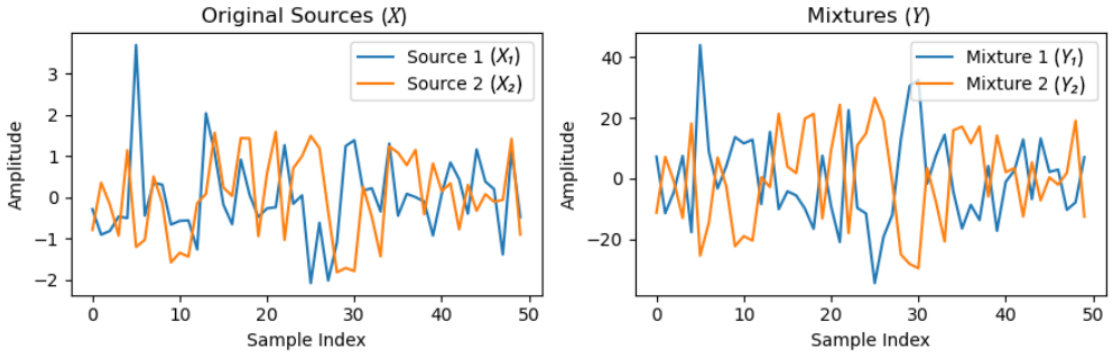

Figure 4.8: Line plots of sources $\mathbf{X}$ and the mixtures $\mathbf{Y}$.

after mixing. As a reminder, only the data $\mathbf{Y}$ is known in practical situations. They may or may not have zero mean and identity dispersion matrix, though as explained in Section 1.5, we can always center the data. A crucial point is that we will skip the pre-processing part of whitening the data. It is unnecessary for our implementation, and scaling of the output is handled using a scaling matrix

at the end, as a post-processing step.

Weight Functions

The weight functions have minor constraints — they have to be positive and integrable. The weight functions chosen for the example satisfy both these conditions: $w_i(x) = e^{-x^2}, i = 1, 2$. We choose the same function for both w_1 and w_2, but any function that satisfies the constraints would suffice and in fact, they do not necessarily have to be the same.

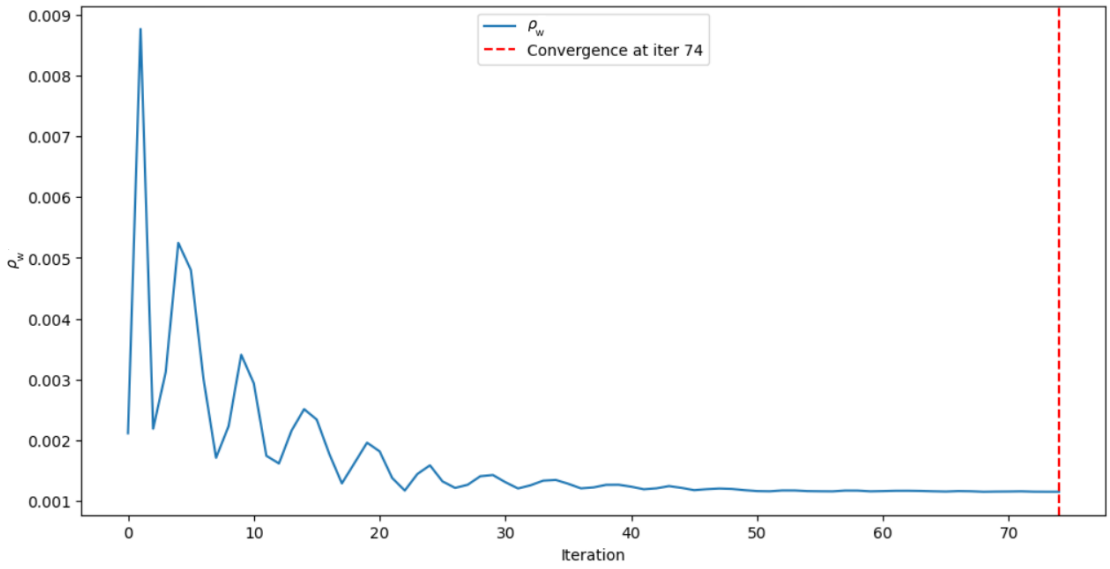

Figure 4.9: Convergence of ρ_w for Example 1 (Section 4.4.1).

Results

The algorithm converges quite fast at only 74 iterations. This is due to the Adam optimizer which is much faster compared to a direct GDA (MinDist) application which takes about 10 times as many iterations to converge (for the same data). Figure 4.16 shows the values of the objective function ρ_w during the iterative minimization process.

The unmixing matrix generated by the GDA part is

$$\mathbf{B}_{\mathrm{a}} = \begin{bmatrix} 0.11855 & 0.08875 \\ 0.05967 & 0.16492 \end{bmatrix}.$$

This is enough to extract the estimates of the sources from the observations. Figure 4.10 shows that the estimates can distinguish the sources very competently, however, the scaling of the independent components has to be managed.

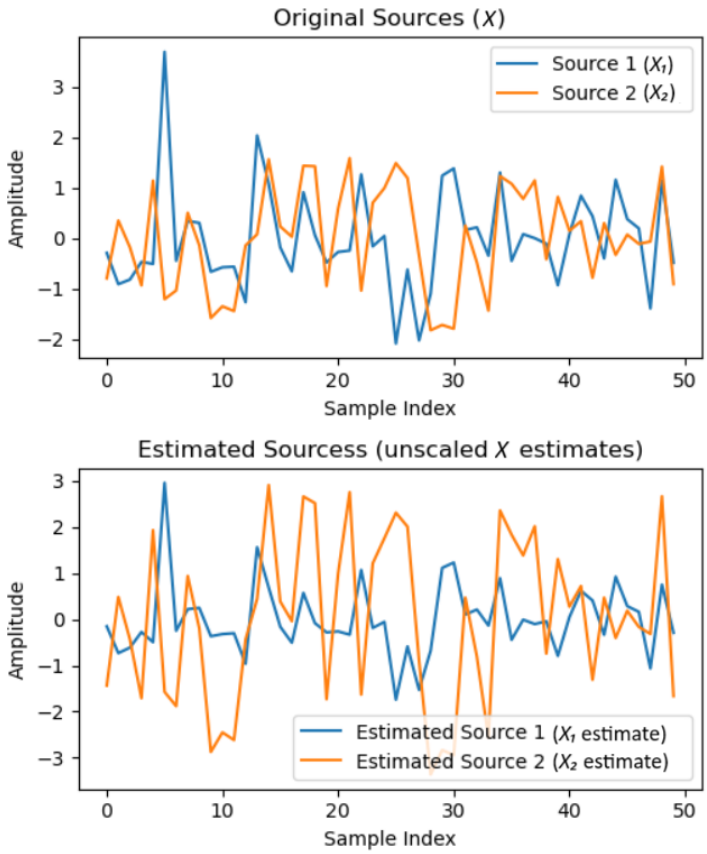

Figure 4.10: Comparison of the sources and their estimates using the matrix $\mathbf{B}_{\mathrm{a}}$.

As explained in Section 4.2.4, this is performed by the post-processing step of normalizing the variances by dividing each by their respective standard deviations. Of course, this is reflected, in matrix terms, by a final estimate of the unmixing matrix given by

$$\hat{\mathbf{B}}_{\mathrm{MinDistICA}} = \begin{bmatrix} 0.1521 & 0.1144 \\ 0.0328 & 0.0929 \end{bmatrix}.$$

The "final" estimates (scaled to unit variance) of the sources are shown in Figure 4.11. The unmixing

matrix obtained via FastICA is

$$\hat{\mathbf{B}}_{\text{FastICA}} = \begin{bmatrix} -0.0763 & -0.1245 \\ -0.1395 & -0.0908 \end{bmatrix}.$$

Though the MinDistICA approach is slightly slower[10] compared to FastICA, it performs better for

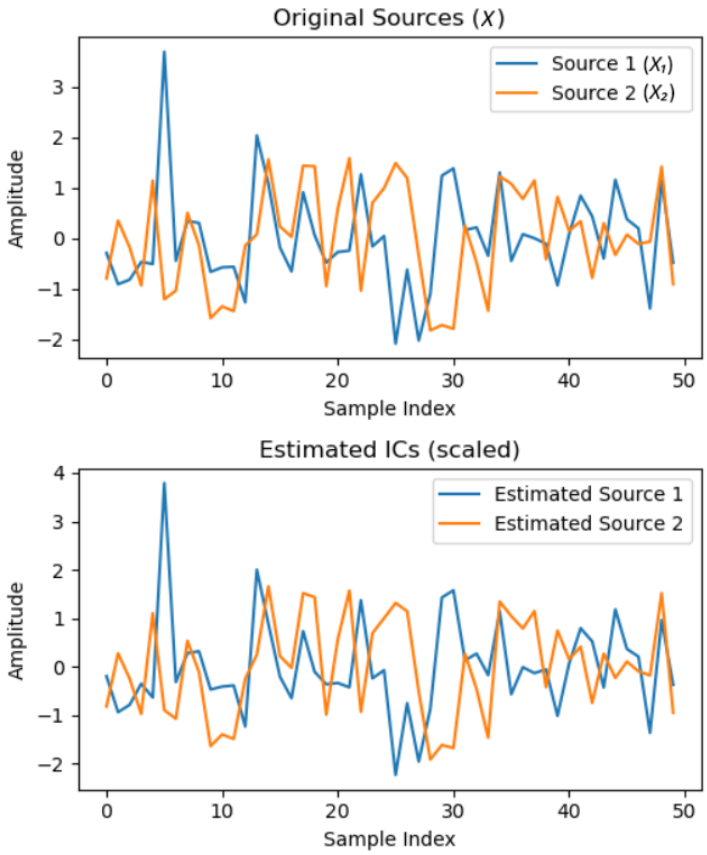

Figure 4.11: Comparison of the sources and their estimates using the matrix $\hat{\mathbf{B}}$.

this example (cf. Figure 4.12, Table 4.2). The sources were generated from distributions with very different kurtoses, and the FastICA implementation used the classical kurtosis-based approximation for the functional form for negentropy approximation[11] `fun = 'cube'`. The signs and estimates have been matched for better visual understanding.

Quantitative Comparison of MinDistICA and FastICA

[10] Since the algorithm involves computation of U-statistics.

[11] The other in-built functions in python for FastICA, `'logcosh'` and `'exp'`, perform worse.

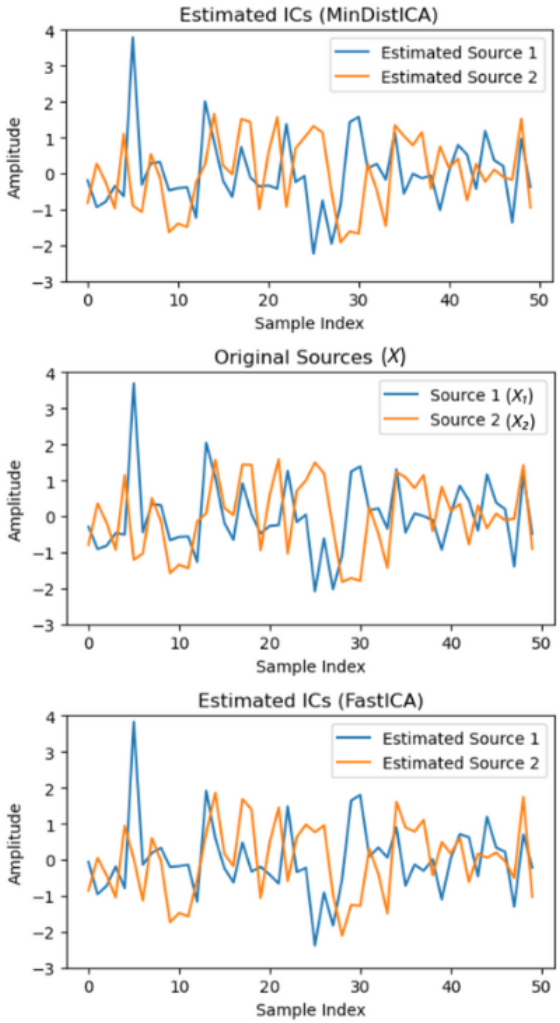

Figure 4.12: The source estimates for Example 1 as obtained from FastICA (bottom) and MinDistICA (top), compared to the sources (mid).

The separation performance of the proposed MinDistICA method is quantitatively compared with FastICA using a synthetic mixture model with a known mixing matrix $\mathbf{A} \in \mathbb{R}^{2\times 2}$ that was used to construct the observed mixtures. The mixing matrix used is

$$\mathbf{A} = \begin{bmatrix} 8 & -12 \\ -2 & 15 \end{bmatrix}$$

The goal of both algorithms was to recover the source signals and estimate the corresponding unmixing matrix $\hat{\mathbf{B}}$. The estimated unmixing matrices were:

$$\hat{\mathbf{B}}_{\text{MinDistICA}} = \begin{bmatrix} 0.1521 & 0.1144 \\ 0.0328 & 0.0929 \end{bmatrix}, \quad \hat{\mathbf{B}}_{\text{FastICA}} = \begin{bmatrix} -0.0763 & -0.1245 \\ -0.1395 & -0.0908 \end{bmatrix}.$$

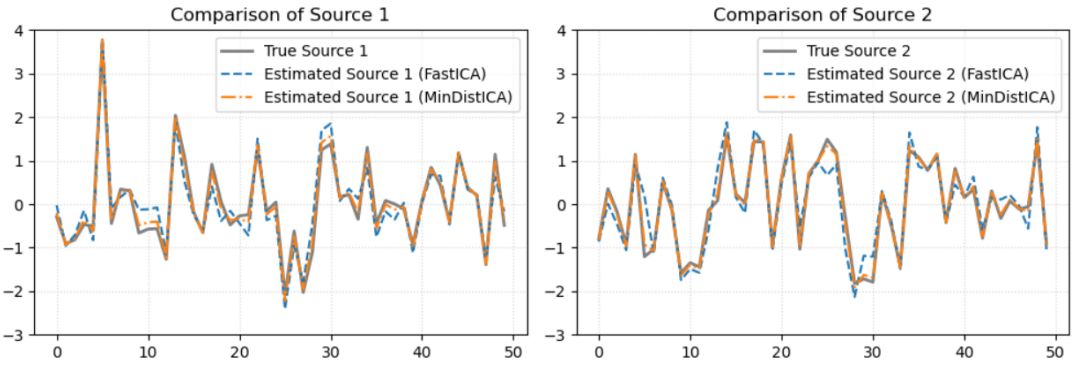

Figure 4.13: A comparison of each of the sources and their estimates.

First, we check the performance of the MinDistICA estimator using the trace-based metric developed in Section 3.3.2. Note that

$$\hat{\mathbf{B}}_{\text{MinDistICA}}\mathbf{B}^{-1} = \begin{bmatrix} 0.9883 & -0.1091 \\ 0.0761 & 1.0012 \end{bmatrix}$$

is already aligned, and so, $d_{\mathcal{F}} = \text{tr}(\mathbf{D}^\top \mathbf{D}) = 0.0178$, where $\mathbf{D} = \hat{\mathbf{B}}_{\text{MinDistICA}}\mathbf{B}^{-1} - \mathbf{I}_2$.

The quality of separation was assessed using three complementary metrics: the *Amari Index*, *Signal-to-Interference Ratio (SIR)*, and *Average Correlation Coefficient*

Table 4.2: Quantitative performance comparison between FastICA and MinDistICA (Example 1)

Metric	**FastICA**	**MinDistICA**
Amari Index	0.3564	0.0931
Mean SIR (dB)	8.9855	20.7799
$\text{SIR}_i; i = 1, 2.$	9.7186, 8.2524	19.2042, 22.3555
Avg. Corr. w/ True Sources	0.9416	0.9956
Corr. (MinDist vs Fast)	0.9681	

Interpretation:

Amari Index: MinDistICA achieved a significantly lower Amari index (0.0931), indicating that its estimated unmixing matrix more closely approximates the true unmixing matrix than FastICA (0.3564).

Signal-to-Interference Ratio (SIR): MinDistICA recovered the source signals with substantially higher fidelity, achieving a mean SIR of 20.7799 dB compared to 8.9855 dB for FastICA. This reflects a clear improvement in signal separation. The per-source SIR values reveal that the MinDistICA algorithm demonstrates not only overall improvement but also consistently better separation across individual sources. The large gap in SIR values — particularly the second component (22.3555 dB vs. 8.2524 dB) — indicates that MinDistICA better preserves the identity of individual signals.

Average Correlation: The correlation between MinDistICA recovered sources and the true sources was 0.9956, a nearly perfect alignment. FastICA yielded a correlation of 0.9416, which is also good but still less. The two estimated solutions themselves (MinDistICA vs FastICA) had a correlation of 0.968, indicating substantial but not complete agreement.

These results provide evidence that MinDistICA offers superior source recovery performance over FastICA, in both matrix-level estimation accuracy and signal-level separation quality.

4.4.2 Example 2

The independent components X_1 and X_2, as with the last example, are generated separately from distinct distributions.

i. The variable X_1 is sampled from Gaussian distribution with mean 0 and and variance 1.

ii. The variable X_2 is sampled from an exponential distribution with scale parameter 1.

Here we choose a larger sample size ($n = 100$). The sampled sources are then standardized to ensure zero mean and unit variance. The synthetic observations $\mathbf{Y}$ are generated using the mixing matrix

$$\mathbf{A} = \begin{bmatrix} 5 & -3 \\ -6 & 2 \end{bmatrix}, \text{ with the unmixing matrix being } \mathbf{B} = \begin{bmatrix} -0.250 & -0.375 \\ -0.750 & -0.625 \end{bmatrix}.$$

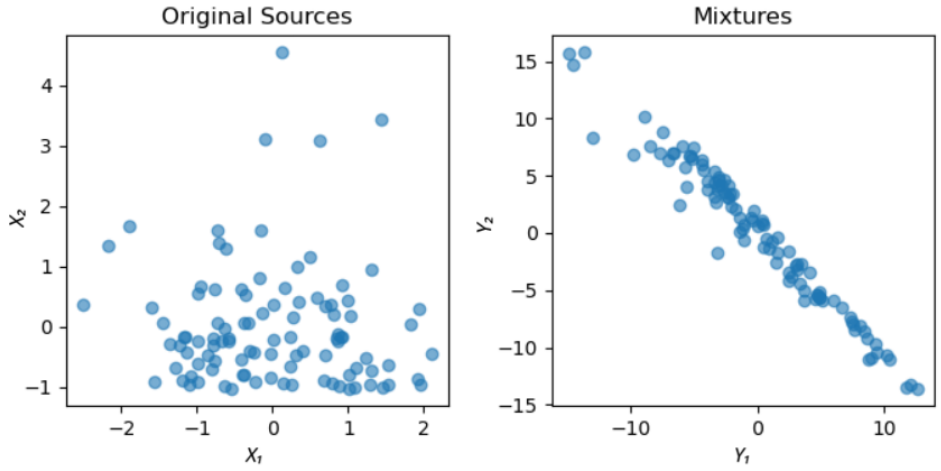

Figure 4.14: Scatterplots of the sources $\mathbf{X}$ and the mixtures $\mathbf{Y}$.

The MinDistICA algorithm converges in 131 iterations, and the following unmixing matrix is obtained:

$$\hat{\mathbf{B}}_{\text{MinDistICA}} = \begin{bmatrix} 0.1521 & 0.1144 \\ 0.0328 & 0.0929 \end{bmatrix},$$

while the unmixing matrix from FastICA is

$$\hat{\mathbf{B}}_{\text{FastICA}} = \begin{bmatrix} 0.1830 & 0.3176 \\ 0.7865 & 0.6805 \end{bmatrix}.$$

Figure 4.15 provides a visual comparison of the estimates[12] and the sources. Both appear to perform very well, and it is hard to distinguish which is better. The quantitative measures provide a better understanding and are summarized in Table 4.3.

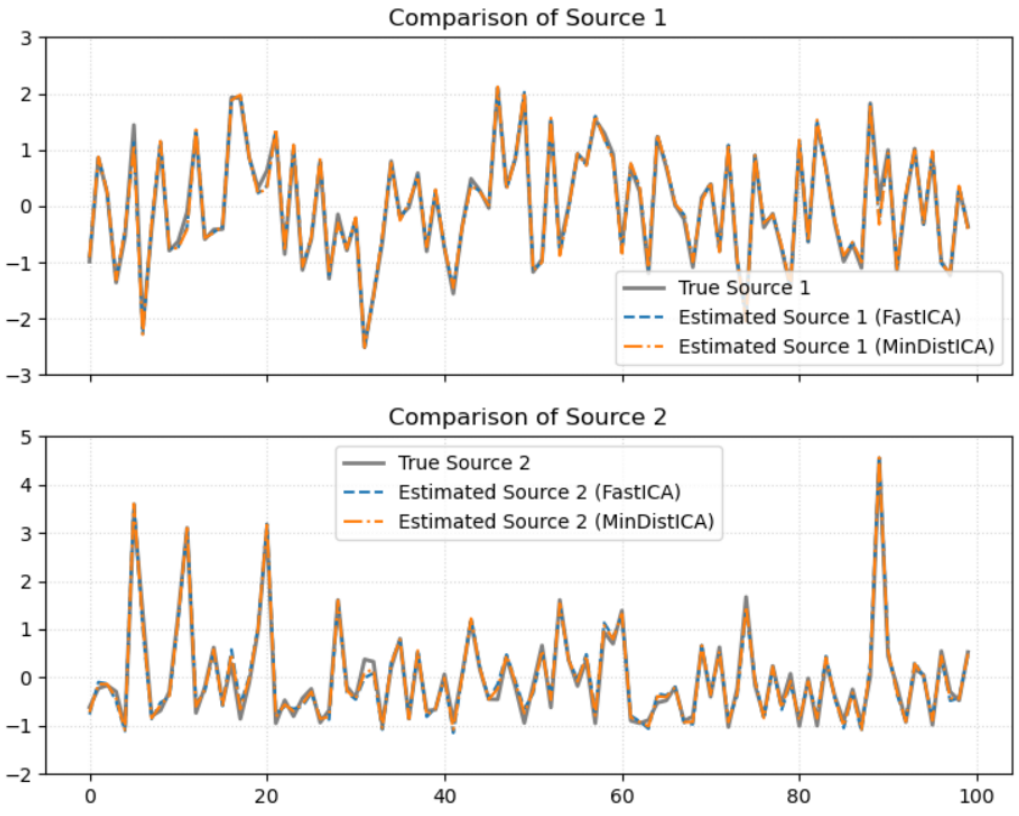

Figure 4.15: A comparison of each of the sources and their estimates for Example 2.

The MinDistICA method gives a very good estimate, and this is evident from the small value attained by $d_{\mathcal{F}}$. Since

$$\hat{\mathbf{B}}_{\text{MinDistICA}}\mathbf{B}^{-1} = \begin{bmatrix} -0.1061 & -1.0013 \\ -0.9889 & 0.0976 \end{bmatrix},$$

the estimate isn't aligned. After proper alignment (cf. Example in Section 3.3.2), the metric yields $d_{\mathcal{F}} = 0.0209$.

The comparative results demonstrate that both FastICA and MinDistICA exhibit high-quality

[12]The sources and estimates are "order" matched for comparison.

separation performance, though MinDistICA again slightly outperforming FastICA across the metrics. MinDistICA yields a marginally lower Amari index (0.1025) compared to FastICA (0.1191). Both methods achieve high mean SIR values, with MinDistICA obtaining a slightly higher average (19.8119 dB vs. 18.8151 dB) implying that both algorithms are effective at suppressing interference between sources. Both methods achieve excellent alignment with the true sources, with MinDistICA (0.9948) again slightly outperforming FastICA (0.9925). Additionally, the correlation between the source estimates from both methods is extremely high (0.9995), showing that they produce highly similar results overall.

Table 4.3: Quantitative performance comparison between FastICA and MinDistICA (Example 2)

Metric	**FastICA**	**MinDistICA**
Amari Index	0.1191	0.1025
Mean SIR (dB)	18.8151	19.8119
$\text{SIR}_i; i = 1, 2.$	21.2759, 16.3543	20.1647, 19.4591
Avg. Corr. w/ True Sources	0.9925	0.9948
Corr. (MinDist vs Fast)	0.9995	

4.4.3 Example 3: Gaussian Sources

For a simulated example, regardless of the source distributions, any ICA algorithm will yield an output for the unmixing matrix estimate, and hence, the source estimates. In fact, even when using Gaussian distribution (for both sources) one might even find exceptional accuracy in the estimation. However, in practical situations, we can't compare against the true sources or the true unmixing matrix, ergo there is no guarantee for the estimates obtained.

Here, the sources X_1 and X_2 are sampled independently from standard Gaussian distributions, i.e., $N(0, 1)$, to adhere to the zero-mean and unit variance assumptions. These sources are then combined using the mixing matrix

$$\mathbf{A} = \begin{bmatrix} 3 & 4 \\ -2 & -6 \end{bmatrix}.$$

to generate the observations Y_1 and Y_2. The true unmixing matrix is given by

$$\mathbf{B} = \mathbf{A}^{-1} = \begin{bmatrix} 0.6 & 0.4 \\ -0.2 & -0.3 \end{bmatrix}.$$

As expected, both the MinDistICA and FastICA algorithms give an estimate. However, the convergence isn't as smooth as the other examples, as seen in the following: Further, the function

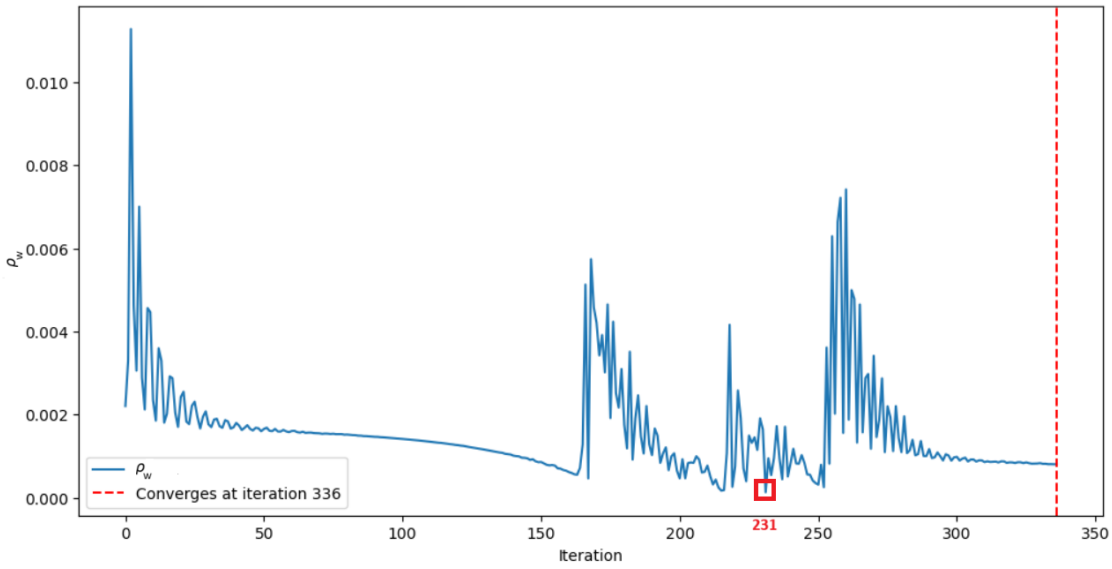

Figure 4.16: Convergence of ρ_w for Example 3 (Section 4.4.3). Note that the function ρ_w is less at iteration 231 (highlighted in red) than where it converged.

fluctuates wildly instead of stabilizing, and even jumps over a local minimum (or at least a smaller value) before converging. The estimated unmixing matrices are obtained as follows:

$$\hat{\mathbf{B}}_{\text{MinDistICA}} = \begin{bmatrix} 0.5226 & 0.3773 \\ 0.0285 & -0.1296 \end{bmatrix} \text{ and } \hat{\mathbf{B}}_{\text{FastICA}} = \begin{bmatrix} 0.0923 & 0.2114 \\ 0.5910 & 0.4481 \end{bmatrix}.$$

A representation of the source estimates is provided in Figure 4.17. As seen here, both methods perform well, though MinDistICA is less accurate, especially for the second source.

The performance measure $d_{\mathcal{F}} = 0.0655$ would suggest "good" estimation. The other measures, summarized in Table 4.4 show that MinDistICA performs worse across all the comparative metrics;

however, it should be pointed out that there is at least one point (corresponding to iteration 231) that is a lower value than where the MinDistICA algorithm converged. This also highlights the importance of fine-tuning the hyper-parameters of algorithms, so over-jumps like this do not occur.

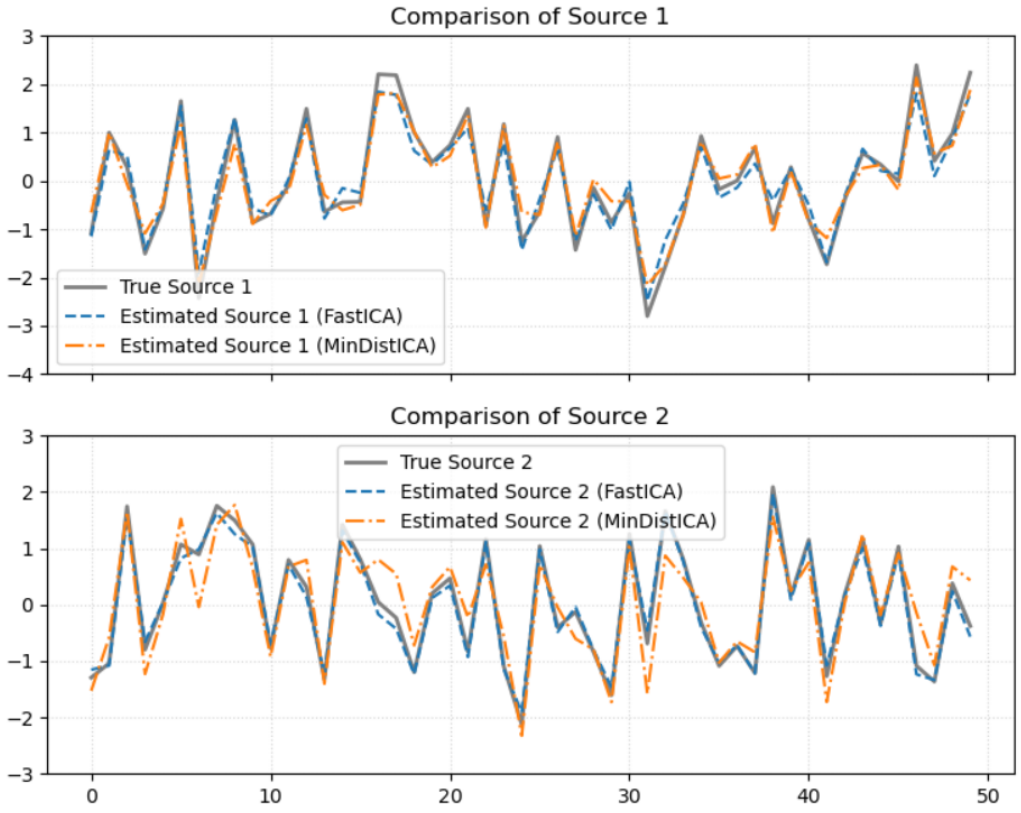

Figure 4.17: A comparison of each of the sources and their estimates for Example 3.

Table 4.4: Quantitative performance comparison between FastICA and MinDistICA (Example 3)

Metric	**FastICA**	**MinDistICA**
Amari Index	0.1407	0.3049
Mean SIR (dB)	17.3044	10.7655
$\text{SIR}_i; i = 1, 2.$	14.7991, 19.8096	14.5407, 6.9903
Avg. Corr. w/ True Sources	0.9897	0.9478
Corr. (MinDist vs Fast)	0.9022	

4.5 Smooth Function Replacement

In Section 1.7, where we proposed our MinDistICA approach, we mentioned the rationale behind using the empirical distribution functions. Specifically, the squared difference between the joint and the product of the marginal empirical distribution functions, to gauge the sources' independence based on the data. Here, we take a similar path to discuss if and how some other smooth continuous functions can be used to replace them and yield similar results.

Consider the joint moment generating function (MGF) $M(x_1, x_2)$ of $\mathbf{X}$ and the marginal MGFs $M_1(x_1)$ and $M_2(x_2)$ of X_1 and X_2. We know that

$$M(x_1, x_2) - M_1(x_1)M_2(x_2) = 0 \; \forall x_1, x_2 \iff X_1 \text{ and } X_2 \text{ are independent.}$$

So, it stands to reason that a measure like (20) but involving continuous functions like MGFs can be used as a substitute. As such, to measure deviations from independence, we define a functional based on the M, M_1 and M_2 as follows:

$$\zeta_w(\mathbf{B}) = \int_{\mathbb{R}^2} \left[M(x_1, x_2) - M_1(x_1)\, M_2(x_2)\right]^2 w_1(x_1)\, w_2(x_2)\, dx_1\, dx_2, \tag{117}$$

where w_i are positive, integrable weight functions. This non-negative function attains zero if and only if X_1 and X_2 are independent.

Given n observations $\{\mathbf{Y}_i\}_{i=1}^n$, we can approximate the MGFs empirically. The joint and marginal MGFs are approximated from the sample by:

$$m_n(x_1, x_2) = \frac{1}{n}\sum_{k=1}^{n} \exp\big(x_1 \mathbf{b}_1^\top \mathbf{y}_k + x_2 \mathbf{b}_2^\top \mathbf{y}_k\big),$$
$$m_{n1}(x_1) = \frac{1}{n}\sum_{k=1}^{n} \exp\big(x_1 \mathbf{b}_1^\top \mathbf{y}_k\big), \quad m_{n2}(x_2) = \frac{1}{n}\sum_{k=1}^{n} \exp\big(x_2 \mathbf{b}_2^\top \mathbf{y}_k\big).$$

One choice for the weight functions is assigning them the Gaussian density $w_i(x) = \dfrac{1}{\sqrt{2\pi}} e^{-x^2/2}$, $i = 1, 2$. Noting w_i are standard Gaussian densities, we may rewrite the integral as an expectation

over $X_i \sim N(0,1)$, i.e.

$$\zeta_w(B) = \mathbb{E}_{X_1,X_2}\big[\Delta_n(X_1,X_2)^2\big], \quad \Delta_n(x_1,x_2) \coloneqq m_n(x_1,x_2) - m_{n1}(x_1)\, m_{n2}(x_2).$$

The gradient with respect to $\mathbf{b}_1$ and $\mathbf{b}_2$ are defined as the vector gradients

$$\nabla_{\mathbf{b}_i}\, \zeta_w(B) = 2\,\mathbb{E}\Big[\Delta_n(X_1,X_2)\, \nabla_{\mathbf{b}_i}\Big[\Delta_n(X_1,X_2)\Big]\Big], \quad i = 1,2.$$

Now, we have

$$\nabla_{\mathbf{b}_1}\, m_n(x_1,x_2) = \frac{1}{n}\sum_{k=1}^{n}\big(x_1\mathbf{y}_k\big)\, \exp\big(x_1\mathbf{b}_1^\top\mathbf{y}_k + x_2\mathbf{b}_2^\top\mathbf{y}_k\big),$$

$$\nabla_{\mathbf{b}_1}\, m_{n1}(x_1) = \frac{1}{n}\sum_{k=1}^{n}\big(x_1\mathbf{y}_k\big)\, \exp\big(x_1\mathbf{b}_1^\top\mathbf{y}_k\big), \quad \nabla_{\mathbf{b}_1}\, m_{n2}(x_2) = 0,$$

and similarly for $\nabla_{\mathbf{b}_2}$. Hence

$$\nabla_{\mathbf{b}_1}\Delta_n = \frac{1}{n}\sum_{k=1}^{n} x_1\mathbf{y}_k\, e^{x_1\mathbf{b}_1^\top\mathbf{y}_k + x_2\mathbf{b}_2^\top\mathbf{y}_k} \;-\; m_{n2}(x_2)\, \frac{1}{n}\sum_{k=1}^{n} x_1\mathbf{y}_k\, e^{x_1\mathbf{b}_1^\top\mathbf{y}_k},$$

with the analogous expression for $\nabla_{\mathbf{b}_2}\Delta_n$.

Setting each gradient to zero yields the system of vector-valued equations

$$\mathbb{E}_{X_1,X_2}\Big[\Delta_n(X_1,X_2)\,\Big(\tfrac{1}{n}\sum_{k=1}^{n} X_1\, y_k\, e^{X_1 b_1^\top y_k + X_2 b_2^\top y_k} \;-\; m_{n2}(X_2)\,\tfrac{1}{n}\sum_{k=1}^{n} X_1\, y_k\, e^{X_1 b_1^\top y_k}\Big)\Big] = \mathbf{0},$$

$$\mathbb{E}_{X_1,X_2}\Big[\Delta_n(X_1,X_2)\,\Big(\tfrac{1}{n}\sum_{k=1}^{n} X_2\, y_k\, e^{X_1 b_1^\top y_k + X_2 b_2^\top y_k} \;-\; m_{n1}(X_1)\,\tfrac{1}{n}\sum_{k=1}^{n} X_2\, y_k\, e^{X_2 b_2^\top y_k}\Big)\Big] = \mathbf{0}.$$

Each expectation can be evaluated in closed form by exchanging sums and Gaussian-moment integrals, leading to expressions involving $e^{\left(\frac{1}{2}(\mathbf{b}_i^\top\mathbf{y}_k)^2\right)}$ and mixed terms $e^{\left(\frac{1}{2}\|(\mathbf{b}_1,\mathbf{b}_2)^\top\mathbf{y}_k\|^2\right)}$. In principle, these yield a system of four non-linear equations in the four unknowns $b_{ij}, i,j = 1,2$. The resulting non-linear system involves composite functions of polynomials and exponents. Any further closed-form theoretical calculation, thus extracting an estimate of $\mathbf{B}$, is not analytically possible. One can proceed to implement and compare numerical optimization methods to obtain the estimator $\hat{\mathbf{B}}$.

The intractable double integral can be approximated numerically. Common strategies include:

- In case of Gaussian weights, one can apply Gauss–Hermite Quadrature which is a numerical tool for approximating values of integrals of the form:

$$\int_{\mathbb{R}} e^{-x^2} f(x)\,\mathrm{d}x \approx \sum_{i=1}^{n} p_i f(x_i)$$

  where x_i are roots of the Hermite polynomial $\mathtt{H}_n^*(x)$ (cf. Appendix A.3), and p_i are weights of the form

$$p_i = \frac{2^{n-1} n! \sqrt{\pi}}{[n\mathtt{H}_{n-1}^*(x_i)]^2}.$$

- Monte Carlo Integration by sampling (x_1, x_2) from the distributions of w_1 and w_2, and approximating the integral as a finite average.

There are certain limitations that can not be overlooked. Firstly, the absolute convergence of the integral is not guaranteed and much depends on the choice of the continuous function used (like the MGF). In particular, MGFs do not always exist. The components would have to be restricted (either known or assumed) to bounded exponential moments. Second, a weight function can be implemented with some hyper-parameter (e.g. $e^{-\nu x_i}$), but these are notoriously difficult to fine tune. For now, there is no theoretical guideline for an optimal choice of the weights. In fact, a different option may be necessary for each dataset. This can be explored in detail in a future study.

Chapter 5

Future Research

5.1 Limitations and Directions for Future Work

The primary limitation of our proposed method is the scalability with both the sample size n and the dimension d. The statistic ρ_w is a U-statistic, whose computational complexity is high for large samples even for $d = 2$. As the number of dimensions increases, both the cost of computing ρ_w as well as the cost of optimization using the MinDistICA algorithm increases dramatically. To alleviate this burden, one can adopt *randomized incomplete U-statistics* with sparse sampling, as developed by X. Chen and Kato (2019). The methods discussed there are computationally less demanding, while retaining important inferential properties.

Optimizations to the base MinDistICA algorithm (Algorithm 9) can also be considered. Although certain provisions were made to enhance the speed, like *memoization* for repeating calculations, vectorizing the operations (instead of looping), etc., it does not make use of the computer's GPU. Rewriting the GDA in PyTorch would enable the use of tools like automatic differentiation (`torch.autograd`) to compute the gradients, allow GPU support for faster computations, and access to AdamW. AdamW is an upgraded version of the Adam optimizer, introduced in Loshchilov and Hutter (2019), that decouples the weight decay and does not let it accumulate in the gradient updates. Both Adam and AdamW are readily available in PyTorch's `torch.optim` module.

The current algorithmic implementation has minimal control over the hyperparameters of the

MinDistICA method like learning rate and the Adam parameters. Hyperparameter-tuning frameworks can be used to remedy this problem. One such framework is Optuna (which can be implemented in Python version 3.8 or higher), based on the work by Akiba, Sano, Yanase, Ohta, and Koyama (2019).

In Section 4.5, we briefly explore the notion of replacing the empirical distribution function (EDF) by a smooth function, such as the MGF (empirical for practical purposes). Although promising in the fact that it's easier to deal with in terms of differentiation compared to EDF which is not smooth, a rigorous characterization of its properties is necessary to ascertain its feasibility in replacing the EDF in the proposed approach.

Last but not least, there is the lack of real-world evaluation. The current study is limited to synthetic simulations involving known distributions. We have not applied the methodology to real-world datasets where underlying distributions of the sources are unknown and inherent noise in the data may violate assumptions about independence and non-Gaussianity. Future extensions should include application to real-world domains like EEG/MEG (where ICA is commonly used for artifact removal) or audio source separation. Such studies would assess assumption violations, as well as address any pre- and post-processing challenges.

5.2 Conclusion

In this work, we have introduced a novel distance-based approach for Independent Component Analysis (ICA) that uses the squared-error distance between the joint empirical distribution and the product of marginal empirical distributions as an objective criterion.

We showed in Chapter 2 that this can be viewed as a U-statistic, admits an explicit asymptotic characterization and permits construction of confidence regions for the unmixing matrix. In Chapter 3, we extend the analysis using the empirical-process, studying the limiting behavior of our minimum-distance estimator. Building on these theoretical foundations, Chapter 4 developed a practical optimization procedure based on gradient descent with the Adam optimizer. Through extensive simulations, we established that MinDistICA achieves competitive source-separation performance, measured through the Amari index, signal-to-interference ratio, and average absolute

correlation coefficient, across a variety of source distributions, often matching or exceeding the results of FastICA. It should be noted that the performances improve with sample size; however due to technical reasons, the simulations were conducted with sample sizes of only 50. Additionally, there was no fine-tuning attempted on the hyperparameters, which can significantly improve results.

The proposed method combines a strong theoretical foundation with a straightforward, data-driven implementation for practical utility. Nevertheless, as discussed in above, challenges unaddressed remain in computational scalability, high-dimensional settings, and real-world use. Addressing these limitations — through randomized incomplete U-statistics, GPU-accelerated implementations, and application to real medical and audio data — provide avenues for future research.

Appendix A

Supplemental Results and Theory

A.1 Eigenvalue Decomposition

Let $\mathbf{M}$ be a $n \times n$ diagonalizable matrix with n linearly independent eigenvectors $\mathbf{p}_i$, $i = 1, \ldots, n$. Then $\mathbf{M}$ can be factorized as

$$\mathbf{M} = \mathbf{PDP}^{-1},$$

where $\mathbf{P}$ is a $n \times n$ matrix with its i^{th} column being the eigenvector $\mathbf{p}_i$ and $\mathbf{D}$ is a diagonal matrix whose diagonal entries are the corresponding eigenvalues λ_i.

A.2 QR-Factorization

Any square matrix $\mathbf{M}$ can be factorized into a product of an orthonormal matrix $\mathbf{Q}$ and an upper-triangular matrix $\mathbf{R}$, i.e.,

$$\mathbf{M} = \mathbf{QR}.$$

The QR-factorization of a matrix can be computed using, among others, the *Gram–Schmidt Orthogonalization* process.

A.3 Hermite Polynomial

The *Hermite polynomials* are an orthogonal polynomial sequence, defined as follows:

$$\mathtt{H}_n(x) = (-1)^n e^{\frac{x^2}{2}} \cdot \frac{\mathrm{d}}{\mathrm{d}x}\left(e^{-\frac{x^2}{2}}\right),$$

with the first few being:

- $\mathtt{H}_0(x) = 1$,
- $\mathtt{H}_1(x) = x$,
- $\mathtt{H}_2(x) = x^2 - 1$
- $\mathtt{H}_3(x) = x^3 - 3x$, and so on.

Specifically, $\mathtt{H}_n(x)$ is called *Probabilist's Hermite Polynomial*. A slightly modified version, called *Physicist's Hermite Polynomial*, given by

$$\mathtt{H}_n^*(x) = (-1)^n e^{x^2} \cdot \frac{\mathrm{d}}{\mathrm{d}x}\left(e^{-x^2}\right),$$

is used in Gauss–Hermite quadrature approximation.

A.4 Rotation Matrices

In the Euclidean space, a *Rotation Matrix* is a matrix that is used to rotate a vector by a certain angle. Mathematically, the matrix

$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

rotates points in the xy-plane anticlockwise through an angle θ about the origin. For clockwise rotation, one can replace θ by $-\theta$.

Result A.4.1. *Let* $\mathbf{Q}$ *be a* 2×2 *matrix with real entries. Then* $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_2$ *iff* $\mathbf{Q}$ *represents a rotation or a reflection matrix.*

Proof. Let $\mathbf{Q} \in \mathbb{R}^{2\times 2}$ be an orthogonal matrix, i.e., $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_2$. Write

$$\mathbf{Q} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

Then the condition $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_2$ implies

$$\mathbf{Q}^\top \mathbf{Q} = \begin{bmatrix} a & c \\ b & d \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} a^2 + c^2 & ab + cd \\ ab + cd & b^2 + d^2 \end{bmatrix} = \mathbf{I}_2.$$

So we must have

$$a^2 + c^2 = 1, \quad b^2 + d^2 = 1, \quad ab + cd = 0.$$

These conditions imply that the columns of $\mathbf{Q}$ are orthonormal vectors in $\mathbb{R}^2$. Define

$$\mathbf{v}_1 \coloneqq \begin{bmatrix} a \\ c \end{bmatrix}, \quad \mathbf{v}_2 \coloneqq \begin{bmatrix} b \\ d \end{bmatrix}.$$

Then $\mathbf{v}_1$ and $\mathbf{v}_2$ are unit vectors and orthogonal, and $\mathbf{Q} = [\mathbf{v}_1 \ \mathbf{v}_2]$.

Consider the determinant $\det(\mathbf{Q}) = ad - bc = \pm 1$.

If $\det(\mathbf{Q}) = 1$, then $\mathbf{Q}$ is a *rotation matrix* with the form:

$$\mathbf{Q} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \theta \in \mathbb{R}.$$

representing a rotation by an angle of θ in the anticlockwise direction.

If $\det(\mathbf{Q}) = -1$, then $\mathbf{Q}$ is a *reflection matrix*. Any such matrix can be written as:

$$\mathbf{Q} = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix}, \theta \in \mathbb{R},$$

which represents a reflection about a line through the origin.

Hence, any 2×2 real orthogonal matrix is either a rotation or a reflection. ■

A.5 Probability Integral Transformation

The *Probability Integral Transformation* is a fundamental yet simple result in probability theory. The statement for this transformation result, as stated in Casella and Berger (2024), is as follows:

Result A.5.1. *Let X have a continuous CDF $F_X(x)$ and define a random variable Y as $Y = F_X(x)$. Then Y is uniformly distributed on $(0, 1)$, i.e., $\mathbb{P}\{Y \leq y\} = y,\ 0 < y < 1$.*

A.6 Bessel Function

The modified Bessel function of the second kind of order $t \in \mathbb{R}$ is defined, for $x > 0$, by

$$K_t(x) = \int_0^\infty e^{-x \cosh y} \cosh(ty)\, \mathrm{d}y. \tag{118}$$

Bibliography

Abelson, H., & Sussman, G. J. (1996). *Structure and Interpretation of Computer Programs* (2nd ed.). MIT.

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

Amari, S.-i., & Cardoso, J.-F. (1997). Blind source separation—Semiparametric statistical approach. *IEEE Transactions on Signal Processing*, *45*(11), 2692–2700.

Amari, S.-i., Cichocki, A., & Yang, H. (1995). A new learning algorithm for blind signal separation. *Advances in Neural Information Processing Systems*, *8*.

Anderson, T. W., & Darling, D. A. (1952). Asymptotic theory of certain "Goodness of Fit" criteria based on stochastic processes. *Annals of Mathematical Statistics*, *23*(2), 193–212.

Casella, G., & Berger, R. L. (2024). *Statistical Inference* (2nd ed.). Chapman & Hall/CRC.

Cauchy, A.-L. (1847). Méthode générale pour la résolution des systemes d'équations simultanées. *Comptes rendus de l'Académie des Sciences*, *25*(1847), 536–538.

Chen, A., & Bickel, P. (2005). Consistent Independent Component Analysis and prewhitening. *IEEE Transactions on Signal Processing*, *53*(10), 3625–3632.

Chen, A., & Bickel, P. J. (2006). Efficient Independent Component Analysis. *Annals of Statistics*, *34*(6), 2825–2855.

Chen, X., & Kato, K. (2019). Randomized Incomplete U-statistics in high dimensions. *Annals of Mathematical Statistics*, *47*(6), 3127–56.

Dynkin, E. B., & Mandelbaum, A. (1983). Symmetric statistics, Poisson point processes, and

multiple Wiener integrals. *Annals of Statistics*, *11*(3), 739–745.

Gaunt, R. E. (2018). Products of Normal, Beta and Gamma random variables: Stein operators and distributional theory. *Brazilian Journal of Probability*, *32*(2), 437–466.

Hallin, M., & Mehta, C. (2015). R-estimation for asymmetric Independent Component Analysis. *Journal of the American Statistical Association*, *110*(509), 218–232.

Hyvärinen, A. (1999). Fast and robust fixed-point algorithms for Independent Component Analysis. *IEEE Transactions on Neural Networks*, *10*(13), 626–634.

Hyvärinen, A., Karhunen, J., & Oja, E. (2001). *Independent Component Analysis*. John Wiley & Sons.

Hyvärinen, A., & Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural networks*, *13*(4-5), 411–430.

Ilmonen, P., & Paindaveine, D. (2011). Semiparametrically efficient inference based on signed ranks in symmetric Independent Component Models. *Annals of Statistics*, *39*(5), 2448–2476.

Jin, K. (1992). Empirical smoothing parameter selection in adaptive estimation. *Annals of Statistics*, *20*(4), 1844–1874.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Y. Bengio & Y. LeCun (Eds.), *Proceedings of the 3rd International Conference on Learning Representations (ICLR).*

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval research logistics quarterly*, *2*(1-2), 83–97.

Lewis, D. (1991). *Matrix Theory*. World Scientific.

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations.*

Löwdin, P.-O. (1970). On the nonorthogonality problem. *Advances in Quantum Chemistry*, *5*, 185–199.

Ma, H., Zheng, X., Wu, X., Yu, L., & Xiang, P. (2022). A Blind Separation Algorithm for underdetermined Convolutional Mixed Communication Signals based on Time–Frequency Soft Mask. *Physical Communication*, *53*.

Nocedal, J., & Wright, S. J. (1999). *Numerical Optimization*. Springer.

Polyak, B. T. (2010). *Introduction to optimization*. Optimization Software, Inc.

Samarov, A., & Tsybakov, A. (2004). Nonparametric Independent Component Analysis. *Bernoulli*, *10*(4), 565–582.

Samworth, R. J., & Yuan, M. (2012). Independent Component Analysis via nonparametric maximum likelihood estimation. *Annals of Statistics*, *40*(6), 2973–3002.

Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear Component Analysis as a kernel eigenvalue problem. *Neural Computation*, *10*(5), 1299–1319.

Serfling, R. J. (2009). *Approximation Theorems of Mathematical Statistics*. John Wiley & Sons.

Shorack, G. R., & Wellner, J. A. (2009). *Empirical Processes with Applications to Statistics*. Society for Industrial and Applied Mathematics.

Tsumura, Y. (2020). *A positive definite matrix has a unique positive definite square root.* https://yutsumura.com/a-positive-definite-matrix-has-a-unique-positive-definite-square-root/. (Accessed: 2025-05-08)