Dimension reduction using Independent Component Analysis with an application in business psychology

by

Elena Shchurenkova

Specialist, Plekhanov Russian University of Economics, 2011

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The Faculty of Graduate and Postdoctoral Studies

(Statistics)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

March 2017

Abstract

Independent component analysis (ICA) is used for separating a set of mixed signals into statistically independent additive subcomponents. The methodology extracts as many independent components as there are dimensions or features in the original dataset. Since not all of these components may be of importance, a few solutions have been proposed to reduce the dimension of the data using ICA. However, most of these solutions rely on prior knowledge or estimation of the number of independent components that are to be used in the model. This work proposes a methodology that addresses the problem of selecting fewer components than the original dimension of the data that best approximate the original dataset without prior knowledge or estimation of their number. The trade off between the number of independent components retained in the model and the loss of information is explored. This work presents mathematical foundations of the proposed methodology as well as the results of its application to a business psychology dataset.

Preface

This work is the result of a collaboration with VIVO Team Development. The project was funded by the Natural Sciences and Engineering Council of Canada through an ENGAGE grant, and was aimed at exploring the data collected by VIVO Team Development as part of their business. The statistical research described in this thesis is an original unpublished work performed by the author under the advice and supervision of Prof. Matias Salibian-Barrera. The collaborators from VIVO Team Development provided subject-matter expertise in business psychology. The dataset used in Chapter 4 of this thesis is property of VIVO Team Development.

Table of Contents

List of Tables

List of Figures

Acknowledgements

I would like to acknowledge Professor Matias Salibian-Barrera for his advice and supervision, which has been an immense help throughout my degree and the work presented here. I am also acknowledging the team of VIVO Team Development for providing the dataset that was used in my work and for their business insights. I owe a special thanks to all the faculty, staff, my colleagues and friends in the Statistics Department of The University of British Columbia. Another thanks goes to Natural Sciences and Engineering Council of Canada for funding my research on this project through an Engage grant.

Additionally, I offer gratitude for my colleagues in Raiffeisenbank, especially to Vladimir Snorkin for providing me with additional motivation to pursue Master's level studies. A special thanks goes to Garnik Kakosyan for his help and insistence when it came to education abroad. Last but not least, I owe a great thanks to Andres Sanchez-Ordonez for his inexhaustible encouragement and support.

Dedication

I dedicate this work to my parents Alexander Shchurenkov and Tamara Plyushch with immense gratitude for their support and patience. This work would not be possible without their encouragement and contribution to my education and love for science.

Chapter 1

Overview

This chapter contains a brief overview of the Independent Component Analysis methodology used in this work and a detailed description of the dataset used for the application.

1.1 Introduction

One of the fundamental problems in multivariate data analysis is finding a convenient representation of random vectors for a particular application. It is customary to use linear transformations of the original data for the reasons of simplicity and computational efficiency [13]. Examples of such approaches are Principal Component Analysis, Factor Analysis and projection pursuit. My research is focused on Independent Component Analysis (ICA), a recently-developed tool [11], the goal of which is to find a linear representation of the non-gaussian data such that the resulting components are statistically independent, or as statistically independent as possible. Note that uncorrelated components are not necessarily statistically independent unless gaussian distribution is assumed.

Albeit being a relatively new methodology, ICA has been used in numerous disciplines and applications, including feature extraction and signal processing. A non-extensive list of applications includes:

- Blind source separation with applications to watermarking and audio sources [23] [16], ECG (Bell & Sejnowski [3]; Barros et al. [1], McSharry et al. [20]), EEG (Mackeig et al. [19] [14]);
- Signal and image denoising (Hyvärinen [9]), application in fMRI (Hansen [7])

- Modeling of the hippocampus and visual cortex (Lorincz, Hyvarinen [18]);
- Feature extraction and clustering, (Marni Bartlett, Girolami, Kolenda [2]);
- Compression and redundancy reduction (Girolami, Kolenda, Ben-Shalom [4]).

ICA is particularly useful in a setup where the data can be assumed to be generated by several independent processes, such as in "cocktail party problem". This problem constitutes identifying a particular source signal in a mixture of signals, and, possibly, noise, where little or no prior information is available about the signals forming the mixture or the mixing process. A widely used example of such problem is trying to separate one person's speech in a noisy setting, in other words, identifying a particular speech signal in a mixture of several speakers and some background noise. In my thesis, I apply the ICA methodology to the business psychology survey data in order to identify the independent components that drive patterns in the survey responses.

Applying the ICA methodology to the business survey data is motivated by the fact that since the entire survey is tailored to assess team productivity, all the questions will have highly correlated responses. This happens because all of the survey questions represent different approaches to measuring the same indicator. It is important to identify the driving forces behind the team's productivity, in particular, to assess how many independent sources affect the team's performance. Independent Component Analysis is a natural tool to use for such inquiries. Since the survey contains quite a few questions, we are additionally interested in finding out whether it is possible to reduce the dimensionality of the data with as little information loss as possible. Thus, my research focuses on the dimension reduction using ICA with an application to the business psychology survey data.

Some methodologies used for finding a convenient representation of multivariate data, along with Independent Component Analysis (ICA), include

Principal Component Analysis (PCA) and Factor Analysis (FA) . The key difference of ICA from PCA and FA is that ICA aims for the statistical independence of the resulting components, while the components obtained using PCA and FA are only linearly independent (uncorrelated), which in general does not imply statistical independence. There are other technical differences in the assumptions made in each case, which I will cover in detail in the following chapter.

1.2 Data description

The dataset used in this work consists of accumulated survey responses gathered by VIVO Team Development over the course of their business. VIVO Team Development is an organization that performs consulting services for other companies aimed at improving the companies' staff productivity. In particular, VIVO offers a data-driven approach to identifying productivity flaws as well as training for personnel to repair dysfunctional and improve moderately-functional teams. Productivity assessment is done via analyzing the team's responses to a survey composed by VIVO and calculating the six key team performance indicators. If the training is done to improve the clients' team's performance, subsequent surveys are conducted to assess the progress and track changes in the team productivity with training. This section contains details on the survey as well as the productivity indicators that are measured using the survey.

The Vital Statistics Survey designed by VIVO team development is aimed at a comprehensive assessment of a team's productivity level. The initial survey is conducted before any training takes place to identify the team's strengths and areas for improvement. The survey is taken by each member of the team as well as the team's leader. In case training is done by the team, the subsequent surveys follow after 3 months in training and at conclusion of the training, which can take from 6 months to a year. All the team members take the initial survey as well as the subsequent ones. Some of the VIVO's clients choose to not do training, hence only the initial survey results are recorded for these companies.

The survey consists of over 50 questions, each of which is aimed at assessing the team productivity. Each question asks the responder to choose a degree to which they agree or disagree with a statement based on a scale from 1 to 10, where 1 indicates complete disagreement and 10 – full agreement. Therefore, on each test occasion the response for each team member is a vector with the same length as the number of questions, each entry being an integer from 1 to 10. The responses to each question are then combined to calculate the six key productivity indicators and the three sub-indicators of each key indicator, forming eighteen values. These values represent each team member's perception of the team productivity. These 18 indicators of productivity will be used in this work as the variables of interest. By the nature of the data, all the 18 variables of interest are measured in the same units.

The team members' responses are summarized team-wise to form the average perception of productivity and these results are reported to the client and used for choosing the training plan in case of the initial survey or for tracking the team's progress in case of the subsequent surveys. For the purposes of the independent components extraction, individual responses were used as opposed to the aggregated team-wise data. Also, no distinction was made between the initial and the subsequent surveys.

The dataset used for the analysis in this project contains the individual responses to the Vital Statistics Survey gathered by VIVO Team Development over the course of their business. The latest survey version was introduced in the middle of 2015 and since then has been used for more than 15 companies to assess and improve the performance of over 30 teams, which overall amounts to over 270 individual responses.

Chapter 2

Independent Component analysis

This chapter focuses on the methodology of Independent Component Analysis and the approaches to defining ICA proposed in the existing literature. First, the motivation and intuition behind the method is explained, followed by the rigorous definition and mathematical properties.

2.1 Motivation

Independent Component Analysis is closely related to what is known as a "blind source separation" problem in signal processing [10]. This problem deals with the separation of the original (source) signals from a mixed data with little or no information about the source signals or the mixing process. The problem can be formulated as follows: a set of original unobserved signals recorded in time $\mathbf{s}(t) = (s_1(t), \ldots, s_p(t))^t$ is mixed into the observed signals $\mathbf{x}(t) = (x_1(t), \ldots x_k(t))^t$ with an unknown mixing matrix $A = \{a_{ij}\}_{k \times p}$:

$$\mathbf{x}(t) = A\mathbf{s}(t). \tag{2.1}$$

Note that the number of the original signals k is usually assumed to be equal to the number of the observed signals p. In case $k > p$ linear unmixing can be used, while if $p > k$, one must use the non-linear methods to derive the source signals from the mixtures. Blind source separation aims to find the unmixing matrix $W = \{w_{ij}\}_{p \times k}$ to recover (or approximate) the source

signals from the mixture, as

$$\mathbf{y}(t) = W\mathbf{x}(t), \tag{2.2}$$

where $\mathbf{y}(t)$ is an approximation of $\mathbf{s}(t)$. The methods for blind source separation aim at estimating the unmixing matrix utilizing the assumptions made about the source signals or the mixing process.

An example of blind source separation problem is a well-known "cocktail-party problem". I will be using this example to motivate ICA, following Hyvärinen and Oja [11]. Assume that there are two speakers speaking simultaneously in a room as well as two microphones recording both speakers. The locations of the microphones are unknown and each microphone records a mixture of the speakers' speech signals. If we denote the recorded signals as $x_1(t)$ and $x_2(t)$, where t is the time index, and $s_1(t)$ and $s_2(t)$ as the respective speakers' speech signals, we can write a linear equation:

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t)$$
$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t)$$

Blind source separation in this case refers to estimating the original speech signals $s_1(t)$ and $s_2(t)$ using only the recorded signals $x_1(t)$ and $x_2(t)$, assuming that the mixing parameters a_{11}, a_{12}, a_{21} and a_{22} are also unknown.

ICA attempts to solve this problem by estimating the mixing coefficients using statistical properties of the signals $s_i(t)$. In particular, one can estimate the parameters a_{ij} using the assumption of statistical independence of the original signals $s_i(t)$ and their non-gaussian distributions. Other methods rely on different assumptions about the signals or the mixing process. For example, if one is using Principal Component Analysis to identify the independent signals, one should assume that the original signals, and hence the mixtures, follow multivariate gaussian distribution. Such assumption needs to be made because the components extracted by PCA are linearly independent (uncorrelated), which in general does not imply statistical in-

dependence unless the variables follow a multivariate gaussian distribution. Hence, if one aims for extraction of statistically independent signals, a multivariate gaussian distribution must be assumed. In a wide range of applications the signals are not necessarily gaussian, which provides an additional motivation for ICA.

In my work I use ICA for separating the independent components related to team productivity using the key productivity indicators as mixed signals. It is not unreasonable to assume that the team productivity is driven by several independent sources or components, which are measured in the productivity survey, but not necessarily separated from one another. Since all the survey questions are intended to measure productivity, one can assume that the answers to all questions are driven by combinations of these independent sources.

In the following section I state the rigorous definitions of ICA given in the literature and also describe some ambiguities related to this method, which will be of importance in this research.

2.2 Definition and mathematical notation

In order to define ICA more rigorously we use a statistical "latent variables" model. Assume we are observing k linear mixtures of k independent components. Dropping the time index t we now have:

$$x_i = a_{i1}s_1 + a_{i2}s_2 + \cdots + a_{ik}s_k, \ i = 1, \ldots, k.$$

We assume that each component s_j is a random variable , $j = 1, \ldots, k$. Hence, each mixture x_i is also a random variable $i = 1, \ldots, k$. Also, without loss of generality we assume that all the mixtures as well as all the independent components have zero mean.

For convenience, vector-matrix notation is often used. Let $\mathbf{x}$ be a random vector of the observed mixtures, thus $\mathbf{x} = (x_1, \ldots, x_k)^t$ and $\mathbf{s} = (s_1, \ldots, s_k)^t$ – a vector of independent components. Additionally, define A as a $k \times k$

mixing matrix with elements $\{a_{ij}\}$. Now we can write the ICA model as

$$\mathbf{x} = A\mathbf{s}. \tag{2.3}$$

This model is generative, which means that it describes how the observed data is generated via mixing the independent components $\mathbf{s}$, which are the non-observed latent variables. Our goal is to estimate both the independent components $\mathbf{s}$ and the mixing matrix A using the observed values of $\mathbf{x}$ under as general assumptions as possible.

Note that the definition of independent components in equation (2.3) implies two ambiguities of the method [11]:

1. The variances of the independent components cannot be determined. This is because since both A and $\mathbf{s}$ are unknown, we can scale any element of $\mathbf{s}$ by a constant and divide the corresponding column of A by the same constant to achieve the same multiplication result as before scaling. Thus, in practice we fix the magnitude of the components to be 1 ($Var(s_j) = 1$ for all $j = 1, \ldots, k$) and adjust the matrix A accordingly. This still leaves the ambiguity of the sign, which fortunately is not a big problem in our application.

2. The order of the independent components cannot be determined. We cannot determine the order of the components because for any (invertible) permutation matrix P we can write $A\mathbf{s} = AP^{-1}P\mathbf{s}$, where $P\mathbf{s}$ are the same independent components, just in a different order. This ambiguity will be important later since each run of the ICA algorithm produces independent components in a different order.

The starting point of ICA is assuming that all components of $\mathbf{s}$ are statistically independent. This means that their joint densities are factorisable into a product of k densities. Note that statistical independence implies linear independence for random variables with finite second moments, thus the independent components produced by ICA will be uncorrelated. Furthermore, since uncorrelated variables are not independent unless gaussian distribution is assumed, the principal components obtained using PCA will

not necessarily be statistically independent if the original data is not multivariate gaussian.

This assumption also implies a property of independent random variables, namely that if we have functions $h_1, \ldots, h_k$ then

$$E[h_1(s_1) \times \cdots \times h_k(s_k)] = E[h_1(s_1)] \times \cdots \times E[h_k(s_k)]$$

Note that if the functions $h_1(\cdot), \ldots, h_k(\cdot)$ are linear in $s_1, \ldots, s_k$, this property holds for uncorrelated or linearly independent variables $s_1, \ldots, s_k$. For statistically independent variables this property will hold for a more general class for functions $h_1(\cdot), \ldots, h_k(\cdot)$, such as power functions of the form $h(s) = s^p$, where $p \in \mathbb{R}$, polynomials and others.

For ICA we must assume that the independent components $\mathbf{s}$ are nongaussian. The reason behind the nongaussianity assumption is that any orthogonal transformation of a gaussian random vector has exactly the same distribution as the original random vector (taking into account the zero-mean constraint), therefore in case of gaussian components we can estimate the matrix A only up to an orthogonal transformation, which makes the matrix unidentifiable. In case only one of the components is gaussian, the independent components could still be estimated [11], which makes the methodology useful for a mixture of several signals and a gaussian noise. However, we do not assume the distributions of the components known.

For simplicity, we also assume that the matrix A is square, though this assumption is non-essential and can be relaxed. In Chapter 3 I will describe the methods useful for dealing with non-square mixing matrices.

Note that in most applications of ICA the algorithms first estimate the "inverse" of the mixing matrix A, denoted as W, which can then be used to compute the extracted independent components $\mathbf{s}$ as:

$$\mathbf{s} = W\mathbf{x}. \tag{2.4}$$

The formula above raises the question of whether the assumption that the mixing matrix A is invertible needs to be made. Here we discuss the pre-

processing steps usually performed on the dataset before the ICA methodology is applied to it, that, among other benefits, provide the invertible mixing matrix without making additional assumptions.

2.3 Pre-processing data

Customarily, several pre-processing steps are performed on the data before the ICA methodology is applied to it. These steps include centering and whitening (or sphering) of the original data and could be performed using multiple approaches. Here a few methods are outlined and the benefits of the preprocessing procedures are explained.

1. Centering. This is a simple procedure aimed at making $\mathbf{x}$ a zero-mean vector, thus, if we denote $E(\mathbf{x}) = \mathbf{m}$ setting $\mathbf{x}' = \mathbf{x} - \mathbf{m}$. In practice, where one is working with a dataset consisting of n realizations of the random vector $\mathbf{x}$, thus a data matrix $X = (x_{ij})_{n\times k}$, the sample means are subtracted. Hence, we obtain $X' = (x'_{ij})_{n\times k}$, where

$$x'_{ij} = x_{ij} - \frac{1}{n}\sum_{l=1}^{n} x_{lj}.$$

This step is made solely for simplification of the ICA algorithm, as it makes both the original data and, by relation (2.4), the independent components, centered. This does not imply that the mean of $\mathbf{s}$ cannot be estimated. In fact, after the mixing matrix is estimated with the centered data, one can translate the centered estimates of $\mathbf{s}$ to the original scale by adding $W\mathbf{m}$, which, because of (2.4) serves as the estimate of the mean of $\mathbf{s}$.

2. Whitening (or sphering). Whitening is a linear transformation of the centered data $\mathbf{x}'$, which makes the resulting data $\mathbf{z}$ have an identity covariance matrix. Thus, each component of $\mathbf{z}$ is uncorrelated with the rest and has unit variance. There are numerous ways to perform a whitening transform, among which most researchers use a PCA-type transform through eigenvalue decomposition (or singular value decomposition) of the covariance matrix of the centered data $E(\mathbf{x}'(\mathbf{x}')^t)$. Note that as the covari-

ance matrix is by definition symmetric, the eigenvalue and the singular value decompositions provide the same result. In particular, we decompose the covariance matrix as $E(\mathbf{x}'(\mathbf{x}')^t) = VDV^t$, were V is the $k \times k$ orthonormal matrix, the columns of which are the eigenvectors of $E(\mathbf{x}'(\mathbf{x}')^t)$ and D is the diagonal matrix of its k eigenvalues. We assign

$$\tilde{\mathbf{x}} = VD^{-\frac{1}{2}}V^t\mathbf{x}'.$$

It is easy to see that this transformation indeed "whitens" the data as follows:

$$\begin{aligned} E(\tilde{\mathbf{x}}\tilde{\mathbf{x}}^t) &= VD^{-\frac{1}{2}}V^tE(\mathbf{x}'(\mathbf{x}')^t)VD^{-\frac{1}{2}}V^t \\ &= VD^{-\frac{1}{2}}V^tVDV^tVD^{-\frac{1}{2}}V^t \\ &= VD^{-\frac{1}{2}}DD^{-\frac{1}{2}}V^t \\ &= VV^t = I. \end{aligned}$$

This step is performed for reducing the number of parameters to be estimated as well as for simplification of the ICA algorithm. In case of the whitened original data, the mixing matrix A will always be orthogonal. To see this, we consider:

$$E(\tilde{\mathbf{x}}\tilde{\mathbf{x}}^t) = AE(\mathbf{s}\mathbf{s}^t)A^t = AIA^t = I.$$

The primary goal of this step is reducing the number of the parameters to be estimated from k^2 to $k(k-1)$ as the degrees of freedom of the orthogonal square matrix A are $k(k-1)$. Additionally, as A is now orthogonal, it is implicitly invertible with A^t being the inverse.

An additional step performed in some applications is discarding the eigenvalues of the covariance matrix that are too small and working with the reduced dimension dataset. This step, similar to PCA, offers the benefits of reducing noise and preventing over-learning. In this work, I discuss the

dimension reduction after the ICA algorithm has been performed, thus I do not discard the low eigenvalues at this step.

It can be added that whitening techniques are simple and plentiful and a whitening transformation can be found for any data. Thus, it is advisable to perform the data whitening before any ICA algorithm to reduce the problem complexity.

2.4 ICA with negentropy maximization

Most of existing algorithms for ICA aim to maximize the non-gaussianity of the independent components, that is, maximize the departure of the individual independent component distributions from a gaussian distribution. To see how this principle works first assume we have a vector $\mathbf{x}$ distributed as in equation (2.3). We would like to estimate one of the independent components as a linear combination of the observed variables with some coefficients $y = \mathbf{w}^t\mathbf{x}$. Ideally, we would like $\mathbf{w}^t$ to be equal to one of the rows of W, the inverse of A.

Making a change of variables $\mathbf{z} = A^t\mathbf{w}$ we have (since $\mathbf{x} = A\mathbf{s}$ (2.3)) :

$$y = \mathbf{w}^t A\mathbf{s} = \mathbf{z}^t\mathbf{s}$$

Hence y is a linear combination of s_j with coefficients z_j, $j = 1, \dots k$. For simplicity assume that all the s_j have the same distribution. Since by CLT any linear combination of independent variables is going to be "more gaussian" than the original variables, then by maximizing the non-gaussianity of y we arrive at the solution with only one non-zero z_j, corresponding to one of the s_j, $j = 1, \dots k$ (since identical distributions of the s_j are assumed). The vector $\mathbf{w}$ such that the non-gaussianity of $y = \mathbf{w}^t\mathbf{x}$ is maximized will provide us with one of the independent components.

The optimization landscape for the non-gaussianity of the k-dimensional space of vectors $\mathbf{w}$ has $2k$ local maxima, each corresponding to one of the s_j or $-s_j$ (recall the ambiguities of ICA). Finding all of these local maxima is not very challenging since the independent components are uncorrelated. A

stepwise procedure can be devised, where to estimate the next component, or the next $\mathbf{w}$, we restrict the space to all vectors that are orthogonal to the vectors $\mathbf{w}$ which provided the independent components in the previous steps. This othogonalization step will further explained in section 2.5.

We determined that to estimate the independent components we need to maximize the non-gaussianity of $y = \mathbf{w}^t\mathbf{x}$. Now we review the existing measures of non-gaussianity and choose the one most useful for our purposes.

There are several measures of non-gaussianity, such as kurtosis. Kurtosis of a random variable y is a measure of the "tailedness" of its distribution, i.e. how many extreme observations does the distribution of y produce. Kurtosis is defined as:

$$Kurt(y) = \frac{E(y - E(y))^4}{E[(y - E(y))^2]^2}$$

In general, the further the kurtosis is from the value of 3, which is the kurtosis value for any univariate gaussian distribution, the further the distribution is from a gaussian distribution. The distributions with the kurtosis value greater than 3 (supergaussian) tend to produce more extreme observations than gaussian distributions, while the kurtosis values less than 3 (subgaussian) generally indicate less heavy tails than in gaussian case. Strictly speaking, a kurtosis value of 3 is not unique to the gaussian distributions, other distributions with specific parameters can have such a kurtosis value as well. An example is the binomial distribution with $p = \frac{1}{2} \pm \sqrt{\frac{1}{12}}$. However, it is difficult to find such distributions and such specific values of parameters. Gaussian distributions have kurtosis value of 3 regardless of the parameter values, which is why kurtosis in customarily used to assess the gaussianity of a distribution. However, lately kurtosis is mostly not used in applications because it may not be defined for heavy-tailed distributions [15]. Among all other possible non-gaussainity measures most of the ICA methods use negentropy.

Entropy is one of the basic concepts of information theory and can be interpreted as a measure of deviation from uniformity of a distribution of a

random variable. For continuous random variables and vectors we calculate differential entropy. For example, a random vector $\mathbf{y}$ with a density $f_{\mathbf{y}}(\mathbf{t})$ has entropy:

$$H_{\mathbf{y}} = -\int \cdots \int f_{\mathbf{y}}(\mathbf{t}) \log f_{\mathbf{y}}(\mathbf{t}) d\mathbf{t}$$

It is useful to note that if all the components of $\mathbf{y}$ are mutually independent then we can calculate the entropy for this vector as a sum of the entropies of its components. This can be seen as follows:

$$\begin{aligned}
H_{\mathbf{y}} &= -\int f_{\mathbf{y}}(\mathbf{t}) \log f_{\mathbf{y}}(\mathbf{t}) d\mathbf{t} \\
&= -\int \cdots \int f_{\mathbf{y}}(t_1, \ldots, t_k) \log f_{\mathbf{y}}(t_1, \ldots, t_k) dt_1 \ldots dt_k \\
&= -\int \cdots \int f_{y_1}(t_1) \ldots f_{y_k}(t_k) \sum_{j=1}^{k} \log f_{y_j}(t_j) dt_1 \ldots dt_k \\
&= -\sum_{j=1}^{k} \int \cdots \int f_{y_1}(t_1) \ldots f_{y_k}(t_k) \log f_{y_j}(t_j) dt_1 \ldots dt_k \\
&= -\sum_{j=1}^{k} \int f_{y_j}(t_j) \log(f_{y_j}(t_j)) dt_j \\
&= \sum_{j=1}^{k} H_{y_j}.
\end{aligned}$$

One of the fundamental results of information theory is that the gaussian distribution parametrized as $N(\mu, \sigma^2)$ has the maximum entropy among all distributions with mean μ and variance σ^2 [11],[6].

The concept of negentropy (or negative entropy) can be introduced and used to measure the departure from gaussianity. Negentropy for a random vector $\mathbf{y}$ can be calculated as

$$J_{\mathbf{y}} = H_{\mathbf{y}^{gauss}} - H_{\mathbf{y}} \tag{2.5}$$

Here $\mathbf{y}^{gauss}$ denotes a gaussian random vector with the same mean and covariance matrix as the vector $\mathbf{y}$. Maximizing negentropy to achieve the maximum departure from gaussianity is well-justified by statistical theory and negentropy is considered to be an optimal measure of non-gaussianity as far as statistical properties are concerned [11]. Alternatively, one can minimize the entropy $H_{\mathbf{y}}$ to achieve the same results.

Unfortunately, entropy is hard to calculate due to the need to numerically approximate the density. Therefore, several less computationally expensive approximations for negentropy as a distance measure from a guassian distribution have been proposed in the literature. Among others I will mention *logcosh* and *exponential* approaches shown in [8], since these are the most widely used approximations for ICA implemented in most software packages (R, Matlab, etc.).

In general, both *logcosh* and *exponential* approximations have the following form:

$$J_{\mathbf{y}} \approx \sum_{j=1}^{k} [E(G(y_j)) - E(G(y_j^{gauss}))]^2 \tag{2.6}$$

Here y_j^{gauss} is a gaussian random variable of the same mean and variance as y_j and $G(\cdot)$ is some non-quadratic function. Since we assume that all independent components have zero mean and unit variance, all y_j^{gauss} have a standard normal distribution. Since the second addendum in the equation above is constant, in order to maximize negentropy, one must maximize $E(G(y_j))$. *Logcosh* and *exponential* approaches correspond respectively to

the following choices of the function G:

$$G_{logcosh}(u) = \frac{1}{a} \log \cosh au \tag{2.7}$$

$$G_{exp}(u) = -\exp(-\frac{u^2}{2}) \tag{2.8}$$

Here a is some parameter between 1 and 2.

There are several existing algorithms for performing ICA on a given dataset. Most efficient and widely-used algorithm is FastICA. FastICA uses negentropy to measure non-gaussianity and offers a choice of *logcosh* and *exponential* approaches to the approximation of negentropy. Both of these approaches represent more robust estimations than kurtosis, since kurtosis is very sensitive to extreme observations because of the fourth moment used in its calculation. The comparison of the three approaches is presented in Figure 2.1.

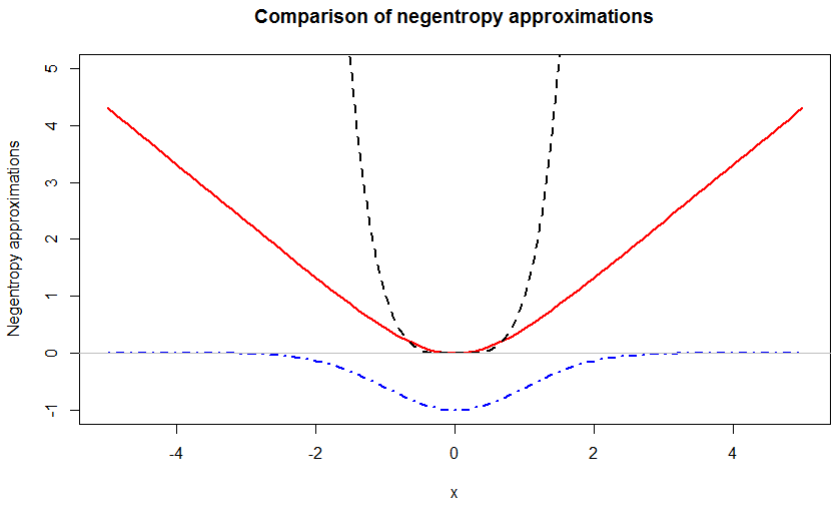

Figure 2.1: The comparison of different approaches to negentropy approximation: kurtosis (based on the 4th degree) is given by the dashed line, *logcosh* approach with $\alpha = 1$ is represented by the solid red line and *exponential* – by the dash-dotted blue curve.

2.5 FastICA algorithm

Here I present the FastICA algorithm, which is used to extract the independent components in this work.

FastICA focuses on finding a matrix W, or a set of k vectors $\mathbf{w}_1, \ldots, \mathbf{w}_k$ that correspond to the k independent components by maximizing the non-gaussianity of $\mathbf{w}_j^t\mathbf{x}, j = 1, \ldots, k$. This section contains a summary of the mathematical justification and derivation of this algorithm, for further details please refer to [11]. Assume that we are working with the pre-processed data $\mathbf{x}$, thus $E(\mathbf{x}) = 0$ and $E(\mathbf{x}\mathbf{x}^t) = I$. For details on the pre-processing steps refer to section 2.3.

Our goal is to maximize negentropy of $\mathbf{w}_j\mathbf{x}$ for all j. Independent of which approximation for negentropy is used, we want to maximize an expression of form:

$$\sum_{j=1}^{k} E(G(y_j)) = \sum_{j=1}^{k} E(G(\mathbf{w}_j^t\mathbf{x})),$$

where $\mathbf{w}_j^t$ is the j-th row vector of the matrix W. Now we write this maximization problem for a particular $\mathbf{w}_j$ dropping the index, in form of Lagrange multipliers, denoting the objective function with O:

$$O(\mathbf{w}) = E(G(\mathbf{w}^t\mathbf{x})) - \frac{\beta}{2}(\mathbf{w}^t\mathbf{w} - 1).$$

The second term is needed because for the pre-whitened data the rows and the columns of matrix W, the inverse of the mixing matrix A, are normalized. We take a derivative of the objective function and set it to zero, getting:

$$D(\mathbf{w}) = \frac{\partial O(\mathbf{w})}{\partial \mathbf{w}} = E(\mathbf{x}g(\mathbf{w}^t\mathbf{x})) - \beta\mathbf{w} = 0. \tag{2.9}$$

Here $g(\cdot)$ is a non-linear function, which is the derivative of $G(\cdot)$ discussed in the previous section. Hence for the *logcosh* and *exponential* approaches respectively:

$$g_{logcosh}(u) = \tanh(au), \tag{2.10}$$

$$g_{exp}(u) = u \exp(-\frac{u^2}{2}). \tag{2.11}$$

The system of equations in (2.9) can be iteratively solved by the Newton-Raphson method, setting each next estimate of $\mathbf{w}$ as

$$\mathbf{w} \leftarrow \mathbf{w} - J_D^{-1}(\mathbf{w})D(\mathbf{w}),$$

where the Jacobian function has the form:

$$J_D(\mathbf{w}) = E(\mathbf{x}\mathbf{x}^t g'(\mathbf{w}^t\mathbf{x})) - \beta I.$$

We can simplify the first term by approximating $E(\mathbf{x}\mathbf{x}^t g'(\mathbf{w}^t\mathbf{x})) \approx E(\mathbf{x}\mathbf{x}^t)E(g'(\mathbf{w}^t\mathbf{x})) = IE(g'(\mathbf{w}^t\mathbf{x}))$ because of the pre-whitening step. With this approximation the Jabobian matrix becomes diagonal, thus can be easily inverted to obtain the following update formula:

$$\mathbf{w} \leftarrow \mathbf{w} - \frac{1}{E(g'(\mathbf{w}^t\mathbf{x})) - \beta}(E(\mathbf{x}g(\mathbf{w}^t\mathbf{x})) - \beta\mathbf{w})$$

Multiplying both sides by a scalar $\beta - E(g'(\mathbf{w}^t\mathbf{x}))$, we get a simple update step:

$$\mathbf{w} \leftarrow E\{\mathbf{x}g(\mathbf{w}^t\mathbf{x}) - E\{g'(\mathbf{w}^t\mathbf{x})\mathbf{w}\}\} \tag{2.12}$$

Note that $\mathbf{w}$ on the left side is multiplied by a scalar, thus renormalization is needed at every step of the iterative algorithm. Also, since the

independent components need to be uncorrelated, when estimating multiple $\mathbf{w}_1, \ldots, \mathbf{w}_k$ one needs to perform orthogonalization of the estimated matrix W at every step, otherwise some $\mathbf{w}_j$'s risk to converge to the same value. The orthogonalization of W can be performed using matrix square roots:

$$W \leftarrow (WW^t)^{-1/2}W. \tag{2.13}$$

The matrix square root shown in equation (2.13) is defined using eigenvalue decomposition of WW^t. Let $WW^t = UDU^t$, where U is the orthonormal matrix of the eigenvectors of WW^t and D is the diagonal matrix of its eigenvalues. We define $(WW^t)^{-1/2} = UD^{-1/2}U^t$, where the inverse and the square root of the diagonal matrix D is applied element-wise to the eigenvalues on the main diagonal. The orthogonalization shown in (2.13) will indeed make W orthonormal as:

$$\begin{aligned} WW^t &= ((WW^t)^{-1/2}W)((WW^t)^{-1/2}W)^t \\ &= (WW^t)^{-1/2}WW^t(WW^t)^{-1/2} \\ &= UD^{-1/2}U^tUDU^tUD^{-1/2}U^t \\ &= UD^{-1/2}DD^{-1/2}U^t \\ &= UU^t = I. \end{aligned}$$

Notice that the procedure described above (Gram-Schmidt orthogonalization) is not the only way orthogonalization can be performed. Alternatively, Cholesky decomposition, Householder transformation and Givens rotation could be used. A few approximate methods also exist for high-dimensional problems.

With all the required steps outlined above, we are ready to write the FastICA algorithm.

Data: The initial data $\mathbf{x}$, the number of independent components k, initial guess for W as $W_0 = (\mathbf{w}_1, \ldots, \mathbf{w}_k)^t$

Whiten $\mathbf{x}$ as described in section 2.2;

repeat

  for $j = 1$ *to* k **do**

    Update $\mathbf{w}_j \leftarrow E\{\mathbf{x}g(\mathbf{w}_j^t\mathbf{x}) - E\{g'(\mathbf{w}_j^t\mathbf{x})\mathbf{w}_j\}\}$, where $g(\cdot)$ is of form (2.10) or (2.11);

  end

  Perform the symmetric orthogonalization of $W = (\mathbf{w}_1, \ldots, \mathbf{w}_k)^t$ by (2.13);

until *convergence*;

Algorithm 1: Pseudo-code for the parallel FastICA algorithm.

Note that the algorithm above could be easily extended to a dataset consisting of n realizations of the random vector $\mathbf{x}$, thus a data matrix $X = (x_{ij})_{n\times p}$. In this case, the expected values in the algorithm above are calculated as sample means (the averages of the rows of the resulting matrices). To simplify the algorithm and decrease computation time when working with large datasets, one can take sample means of randomly chosen subsets of the data.

The convergence here is defined by two subsequent estimates of W being sufficiently close together. Hence, no computation of negentropy or approximations of densities are needed. This is one of the reasons the FastICA algorithm is very efficient.

2.6 Other approaches to Independent Component Analysis

A few other approaches for Independent Component Analysis have been proposed in the literature. Likelihood may seem like a natural tool to use for ICA, especially since deriving the likelihood for the model is not very mathematically challenging. However, there are a few drawbacks to using this approach. First of all, it requires the estimation of the densities of the independent components, which is a difficult task. The solutions proposed

include either using prior information about the independent component densities or restricting the densities of all components to a finite-parameter family. This option leads to another potential drawback: usually the densities of the subgaussian and supergaussian independent components need to be modeled separately because they belong to different distribution families.

Another approach is the minimization of mutual information. Mutual information of a random vector is defined as

$$I(y_1, \ldots, y_k) = \sum_{j=1}^{k} H(y_j) - H(\mathbf{y})$$

and is a natural measure of dependence between the components of the vector. A small difference of this approach with maximizing negentropy is that by maximizing non-gaussianity one restricts the components to be uncorrelated, while in case of mutual information minimization this does not necessarily have to be the case.

Another approach to estimating the unmixing matrix in ICA discussed by K. Nordhausen et al. in [22] relies on scatter matrices with the so-called "independent property". The essence of this method relies on using scatter matrices other than the sample covariance matrix to whiten the data as described in section 2.3. A matrix-valued functional of a random vector $\mathbf{x}$ with CDF $F_{\mathbf{x}}$ is called a scatter matrix if it is positive-definite, symmetric and affine invariant. A sample covariance matrix is an example of a scatter matrix, however there are numerous alternative techniques to constructing those, for example M-functionals, S-functionals and etc. A scatter matrix is said to have the independent components property if $S(F_{\mathbf{x}})$ is a diagonal matrix for all random vectors $\mathbf{x}$ with mutually independent components. A covariance matrix is an example of scatter matrix with this property. M-functionals and S-functionals do not generally have this property but can be symmetrized to possess the independent components property. Using two of scatter matrices with the independent components property yields an estimate of an unmixing matrix that will depend on which scatter matrices are used to construct it. Nordhausen et al. [22] recommend using robust options

for both scatter matrices, which yields favorable estimates of the unmixing matrix in case of symmetrically distributed independent components. The procedure is however computationally costly and known to produce poorer results for asymmetric independent components.

A few other methods include tensorial methods proposed by L. De Lathauwer [17], nonlinear ICA reviewed in [5] and Bayesian approach to Independent Component Analysis recently revisited by Mohammad-Djafari [21].

Chapter 3

Component selection

One of the assumptions made in ICA is that the mixing matrix A is square, therefore, there are as many independent components as there are the recorded mixed signals. In order to relax this assumption, a few methods have been proposed, most of which require the researcher to choose or estimate the number of components to be extracted prior to the application. For example, one can use a Factor Analysis-type model of form

$$\mathbf{x} = A\mathbf{s} + \epsilon,$$

where A is a k-by-p real matrix, $\mathbf{x}$ is a k-dimensional random vector of the original data and $\mathbf{s}$ is a p-dimensional vector of the independent components. For noiseless ICA ϵ is assumed to be a k-dimensional deterministic vector of zeros.

My research is concerned with using fewer independent components than the dimensionality of the original data. That is, finding a way to choose the most "useful" components out of the ones extracted in case the number of components is not specified prior to the application. In this chapter I will first briefly introduce some existing approaches to extracting less components than the number of features in the original dataset and then discuss the proposed approach and its mathematical basis.

3.1 Existing approaches

There is little literature available on the topic of dimensionality reduction using ICA because ICA was not originally developed for that purpose [24].

One method for producing fewer independent components than the dimensionality of the original data proposed in [11] suggests to reduce the dimension of the problem while performing the pre-processing (whitening) of the data. Recall, data whitening is often performed with a Principal Components-type procedure, where the eigenvector decomposition of the covariance matrix is used as follows: $E(\mathbf{x}\mathbf{x}^t) = VDV^t$, where D is the diagonal matrix of eigenvalues of the covariance matrix and V is the orthonormal matrix, the columns of which are the eigenvectors. The whitened data are then computed as:

$$\mathbf{z} = VD^{-\frac{1}{2}}V^t\mathbf{x}',$$

where $\mathbf{x}'$ is the centered version of $\mathbf{x}$ and the inverse and the square root are applied element-wise to the main diagonal of D. It is possible to reduce dimensions to a pre-determined value p by discarding the smallest $k - p$ eigenvalues from D and the corresponding eigenvectors from V. The whitened data with the reduced dimensions is then fed to the ICA model, which produces p independent components. This assumes prior knowledge of the number of independent components that should be retained in the model. Furthermore, the independent components will be estimated on the whitened dataset, with some features excluded when dropping the small eigenvalues, which may be undesirable and complicate the interpretability of the results.

Another recently proposed approach in [24] uses virtual dimensionality to determine the number of components to be retained in the model. Virtual dimensionality is an approach to estimating the number of sources present in a signal/image processing context. This estimate in convolution with a prioritization mechanism is then utilized to choose which components are to be retained. The prioritization mechanism mentioned above is based on the sample skewness and kurtosis of the extracted independent components. Another approach proposed in the same paper suggests running ICA multiple times starting at various initial values and comparing the components

received on each run. The independent components that are output by the algorithm often for various starting points are then deemed statistically important and retained in the analysis, while the components that appear rarely are discarded. The last two approaches were developed in application to hyperspectral image analysis.

3.2 Proposed approach

The approach developed in course of my research focuses on a different way to address the dimension reduction using ICA. When there is no prior knowledge about the number of independent components that should be retained, it may be prudent to perform ICA keeping the original dimensions of the data. After obtaining the independent components, one can choose the components that should be retained and discard the unneeded components. This choice could be made by comparing the predictive power of different combinations and numbers of the retained components or in another way as suits the application goals. An example of a situation, where this method could be applicable, is a multivariate dataset, where each variable is a mix of several signals as smooth functions of time, and noise, or several sources of noise. As long as the condition of independence of the noise from the signal is met, ICA will be able to separate the source signals, among which the noise will be easily identifiable. Once the sources are separated, one can proceed with choosing the components that one is interested in. My research focuses on whether or not there is a meaningful way to choose the components to be retained as well as exploring the trade-off between the retained number of components and the amount of information lost when reducing dimensions in such way.

Assume we are working with a dataset, where each observation has k features and there are n observations overall. Assume here that $n > k$. Let us denote the vector of reduced (or truncated) independent components as $\mathbf{s}_i$, where out of original k components, p have been retained, $i = 1, \ldots, n$. Of course, in practice, for each p there are $\binom{k}{p}$ possibilities of $\mathbf{s}_i$ (the choice of

the retained components is assumed to be consistent across the $i = 1, \ldots, n$). Ideally, we would like to solve the equations:

$$\mathbf{x}_i = A^* \mathbf{s}_i, i = 1, \ldots, n \tag{3.1}$$

for the mixing matrix $A^*_{k \times p}$ in order to fully reconstruct the original data without any information loss using only p out of the k independent components.

As explained in section 2.3, customarily the data is pre-processed, i.e. centered, scaled and whitened, before the application of ICA. Denote the pre-processed data $\mathbf{z}_i$, where $\mathbf{z}_i$ for $i = 1, \ldots, n$ can be written as:

$$\mathbf{z}_i = V D^{-\frac{1}{2}} V^t (\mathbf{x}_i - \mu) \tag{3.2}$$

Here μ is the vector of expected values of $\mathbf{x}_i$, D is the diagonal matrix of the eigenvalues of the covariance matrix $\frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^t$ and V is the matrix, whose columns are the eigenvectors. Hence, $E(\mathbf{z}) = \mathbf{0}$ and $\frac{1}{n} \sum_{i=1}^{n} \mathbf{z}_i \mathbf{z}_i^t = I_{k \times k}$.

We assume that independent components $\mathbf{s}_i$ have mean zero and covariance matrix $I_{p \times p}$. Therefore, for (3.1) to be satisfied we need:

$$I_{k \times k} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{z}_i \mathbf{z}_i^t = \frac{1}{n} \sum_{i=1}^{n} A^* \mathbf{s}_i \mathbf{s}_i^t (A^*)^t = A^* I_{p \times p} (A^*)^t = A^* (A^*)^t$$

Thus, we would need the rank of A^* to be k, but since A^* is a $k \times p$ matrix, where $p < k$, this situation cannot occur.

We have shown that an exact reconstruction of the data is not possible using fewer than the original number of components. Now, consider the case that we want to get the best possible approximation to the original data by using the reduced independent components extracted from that data. We can measure the quality of approximation by the square of the L_2 norm of the difference between the original (or the whitened) data and the estimated values, obtained by reconstruction using the reduced independent components. For now we are working with the whitened data $\mathbf{z}_i$, $i = 1, \ldots, n$. In

order to minimize the information loss, thus getting the best approximation of the original data, we need to obtain a mixing matrix which minimizes the following target function:

$$\mathcal{F}(A) = \sum_{i=1}^{n} \|\mathbf{z}_i - A\mathbf{s}_i\|_2^2 \tag{3.3}$$

Note the $\mathbf{s}_i$ vectors here are obtained from the whitened data as described in Chapter 2. Additionally, please notice that our target function is a square of Euclidean norm. We need to find:

$$\min_A \mathcal{F}(A) = \min_A \sum_{i=1}^{n} \sum_{j=1}^{k} \left(z_{ij} - \sum_{l=1}^{p} A_{jl} \times s_{il}\right)^2 \tag{3.4}$$

This is a problem of a function minimization on a manifold of non-square matrices. In order to solve the problem in (3.4) we compute the Frechet derivative of the L_2-norm in (3.3) as a function of a $k \times p$ matrix A.

Frechet derivative is a generalization of a derivative of real-valued function of a single variable to (possibly) vector-valued function of multiple real variables. It is defined as follows. Let W and V be Banach spaces and $U \subset V$ be an open subset of V. A function $F : U \to W$ is called Frechet differentiable at $A \subset U$ if there exists a bounded linear operator $Df(A, H)$ such that

$$\lim_{\|H\|_V \to 0} \frac{\|f(A+H) - f(A) - Df(A,H)\|_W}{\|H\|_V} = 0 \tag{3.5}$$

Frechet derivative is defined on Banach spaces, which include matrix manifolds. In our case, space W is the real line and space V is a matrix manifold of $p \times k$ real-valued matrices, while H is a $p \times k$ real matrix (the direction matrix). Hence the function $\mathcal{F} : \mathbb{R}^{p \times k} \to \mathbb{R}$ defined in (3.3) could be Frechet differentiable. The limit in the equation above is a usual limit of a function defined on a metric space, thus the expression in (3.5) is a function of H in V, hence the first-order expansion holds. We attempt to find the linear operator $D\mathcal{F}(A, H)$ using this first-order expansion:

$$\mathcal{F}(A+H) = \mathcal{F}(A) + D\mathcal{F}(A,H) + o(\|H\|) \tag{3.6}$$

We apply the definition given in equation (3.6) to the minimization problem in (3.4) to get the expression for the Frechet derivative. $H^{(1)}$ denotes the direction matrix for the first order derivative.

$$\begin{aligned}
\mathcal{F}(A+H^{(1)}) &= \sum_{i=1}^{n}\sum_{j=1}^{k}\big(z_{ij} - \sum_{l=1}^{p}(A_{jl}+H_{jl}^{(1)})\times s_{il}\big)^2 \\
&= \sum_{i=1}^{n}\sum_{j=1}^{k}\big((z_{ij} - \sum_{l=1}^{p}A_{jl}\times s_{il}) - \sum_{l=1}^{p}H_{jl}^{(1)}\times s_{il}\big)^2 \\
&= \sum_{i=1}^{n}\sum_{j=1}^{k}\big((z_{ij} - \sum_{l=1}^{p}A_{jl}\times s_{il})^2 - 2(z_{ij} - \sum_{l=1}^{p}A_{jl}\times s_{il})(\sum_{l=1}^{p}H_{jl}^{(1)}\times s_{il}) \\
&+ [\sum_{l=1}^{p}H_{jl}^{(1)}\times s_{il}]^2\big) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{k}(z_{ij} - \sum_{l=1}^{p}A_{jl}\times s_{il})^2 - 2\sum_{i=1}^{n}\sum_{j=1}^{k}(z_{ij} - \sum_{l=1}^{p}A_{jl}\times s_{il})(\sum_{l=1}^{p}H_{jl}^{(1)}\times s_{il}) \\
&+ \sum_{i=1}^{n}\sum_{j=1}^{k}[\sum_{l=1}^{p}H_{jl}^{(1)}\times s_{il}]^2 \\
&= \mathcal{F}(A) + D\mathcal{F}(A,H^{(1)}) + o(\|H^{(1)}\|)
\end{aligned}$$

Thereby, the Frechet derivative of the cost function in (3.3) has the form:

$$D\mathcal{F}(A,H^{(1)}) = -2\sum_{i=1}^{n}\sum_{j=1}^{p}(z_{ij} - \sum_{l=1}^{k}A_{jl}\times s_{il})(\sum_{l=1}^{k}H_{jl}^{(1)}\times s_{il})$$

In order for A_0 to solve the minimization problem in (3.4) we need two conditions to hold:

1. $D\mathcal{F}(A,H^{(1)}) = 0$ for all $H^{(1)}$;

2. There exists an $a > 0$ such that $D^2\mathcal{F}(A, H^{(2)}, H^{(2)})) \geq a\|H^{(2)}\|$ for all $H^{(2)}$.

We would like to find such A_0 that both conditions above are satisfied. We propose A_0 as:

$$A_0 = \frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_i(\mathbf{s}_i)^t. \tag{3.7}$$

This is a $k \times p$ matrix as $\mathbf{z}_i$ is a k-dimensional vector of the original data for all $i = 1, \ldots n$ and $\mathbf{s}_i$ is a p-dimensional truncated vector of the independent components. We need to show that this choice of A_0 satisfies the condition 1 above, or $D\mathcal{F}(A_0, H^{(1)}) = 0$ for all choices of $H^{(1)}$. First, we notice that:

$$\begin{aligned} D\mathcal{F}(A, H^{(1)}) &= -2\sum_{i=1}^{n}\sum_{j=1}^{k}(\mathbf{z}_{ij} - (A\mathbf{s}_i)_j)(H^{(1)}\mathbf{s}_i)_j \\ &= 2\sum_{i=1}^{n}\sum_{j=1}^{k}(\mathbf{z}_i - A\mathbf{s}_i)_j(H^{(1)}\mathbf{s}_i)_j \\ &= -2\sum_{i=1}^{n}(\mathbf{z}_i - A\mathbf{s}_i)^t H^{(1)}\mathbf{s}_i. \end{aligned}$$

Further, rewriting the product above as a trace and using the properties of independent components we have:

$$
\begin{aligned}
D\mathcal{F}(A_0, H^{(1)}) &= -2\sum_{i=1}^{n}(\mathbf{z}_i - A_0\mathbf{s}_i)^t H^{(1)}\mathbf{s}_i \\
&= -2\sum_{i=1}^{n} tr\big((\mathbf{z}_i - A_0\mathbf{s}_i)^t H^{(1)}\mathbf{s}_i\big) \\
&= -2\sum_{i=1}^{n} tr\big(\mathbf{s}_i(\mathbf{z}_i - A_0\mathbf{s}_i)^t H^{(1)}\big) \text{ using the cyclic invariance of the trace} \\
&= -2\sum_{i=1}^{n} tr\big((\mathbf{s}_i\mathbf{z}_i^t - \mathbf{s}_i\mathbf{s}_i^t A_0^t)H^{(1)}\big) \text{ using the transposition properties} \\
&= -2n \times tr\left((\frac{1}{n}\sum_{i=1}^{n}\mathbf{s}_i\mathbf{z}_i^t - [\frac{1}{n}\sum_{i=1}^{n}\mathbf{s}_i\mathbf{s}_i^t]A_0^t)H^{(1)}\right) \\
&= -2n \times tr\left((A_0^t - I_{p\times p}A_0^t)H^{(1)}\right) = 0 \text{ using the independence of } \mathbf{s}_i.
\end{aligned}
$$

We have shown that A_0 satisfies the condition 1 above. Now we would like to show that the condition 2 also holds for our choice of A_0. In fact, the condition is a definition of a strongly convex (or α-convex) function. Recall, that our target function is a square of L_2 norm (Euclidean norm). We can prove the strong convexity of our function by first noticing that $\mathcal{F}$ can be rewritten as:

$$\mathcal{F}(A) = \sum_{i=1}^{n}\sum_{j=1}^{k}\|\mathbf{z}_{ij} - A_j\mathbf{s}_i\|_2^2, \tag{3.8}$$

where A_j denotes the j-th row of A, $A_j \in \mathbb{R}^p$, $j = 1, \dots k$. As $\mathcal{F}$ can be written as a sum of nk squares of L_2 norms of affine functions of A_j, it is enough to show that the square of L_2 norm of a vector is a strongly convex function and then use the property that the sum of strongly convex functions is still strongly convex.

One of the forms of definition of a strongly convex function f with parameter α is that for all k-dimensional real-valued vectors $\mathbf{x}$ and $\mathbf{y}$ and for $t \in [0, 1]$ is:

$$f(t\mathbf{x} + (1-t)\mathbf{y}) - tf(\mathbf{x}) - (1-t)f(\mathbf{y}) \leq -\frac{1}{2}\alpha t(1-t)\|\mathbf{x}-\mathbf{y}\|_2^2 \qquad (3.9)$$

In our case $f(\mathbf{x}) = \|\mathbf{x}\|_2^2$. Let $\mathbf{x}$ and $\mathbf{y}$ be p-dimensional vectors, let $\langle \mathbf{x}, \mathbf{y} \rangle$ denote the inner product of $\mathbf{x}$ and $\mathbf{y}$. We expand the definition above to get:

$$\begin{aligned}
&f(t\mathbf{x} + (1-t)\mathbf{y}) - tf(\mathbf{x}) - (1-t)f(\mathbf{y}) \\
&\qquad = t^2\|\mathbf{x}\|_2^2 + 2t(1-t)\langle \mathbf{x}, \mathbf{y} \rangle + (1-t)^2\|\mathbf{y}\|_2^2 - t\|\mathbf{x}\|_2^2 - (1-t)\|\mathbf{y}\|_2^2 \\
&\qquad = t(t-1)\|\mathbf{x}\|_2^2 - t(1-t)\|\mathbf{y}\|_2^2 + 2t(1-t)\langle \mathbf{x}, \mathbf{y} \rangle \\
&\qquad = -t(1-t)\left(\|\mathbf{x}\|_2^2 - 2\langle \mathbf{x}, \mathbf{y} \rangle + \|\mathbf{y}\|_2^2\right) \\
&\qquad = -t(1-t)\|\mathbf{x}-\mathbf{y}\|_2^2 \\
&\qquad \leq -\frac{1}{2}t(1-t)\|\mathbf{x}-\mathbf{y}\|_2^2 \text{ as } t(1-t)\|\mathbf{x}-\mathbf{y}\|_2^2 \geq 0
\end{aligned}$$

We have shown that a square of an L_2-norm of a vector is strongly convex with $\alpha = 1$. As the function $\mathcal{F}$ defined in (3.3) is a sum of nk squares of L_2 norms of affine functions of p-dimensional vectors, hence it is strongly convex with the same parameter $\alpha = 1$.

We have shown that our choice of A_0 satisfies the necessary and the sufficient conditions for the global minimum, solving the problem in (3.4).

Let us write $\mathbf{z}_i$ as rows of matrix $Z_{n\times k}$ and $\mathbf{s}_i$ as rows of matrix $S_{n\times p}$. Thus, we get:

$$A_0 = \frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_i \mathbf{s}_i^t = \frac{1}{n} Z^t S \qquad (3.10)$$

It turns out that the matrix A_0 chosen as in (3.10) satisfies $A_0^t A_0 = I_{p\times p}$. In order to show this, first we assume without loss of generality that the independent components retained are the first p out of the k independent components. Hence, we can write $S_{n\times p} = \tilde{S}_{n\times k} P_{k\times p}$, where $\tilde{S}$ is the full

matrix of independent components, thus $\tilde{S}_{n\times k}$, and P is a matrix that allows us to choose first p components of $\tilde{S}$, which can be written as:

$$P = \begin{pmatrix} I_{p\times p} \\ 0_{(k-p)\times p} \end{pmatrix}, \tag{3.11}$$

where $0_{(k-p)\times p}$ is a $(k-p)\times p$ matrix of zeros. Therefore, $P^tP = I_{p\times p}$. We need to remember that $\mathbf{s}_i$ are the independent components that satisfy $Z = SA$ for a $k \times k$ mixing matrix A, furthermore, $I_{k\times k} = \frac{1}{n}Z^tZ = \frac{1}{n}A^tS^tSA = A^tA$, therefore, by the normality of orthogonal matrices, $AA^t = I_{k\times k}$. Now,

$$\begin{aligned} A_0^tA_0 &= (\frac{1}{n}Z^tSP)^t(\frac{1}{n}Z^tSP) \\ &= \frac{1}{n^2}P^tS^tZZ^tSP \\ &= \frac{1}{n^2}P^tS^tSAA^tS^tSP \\ &= P^t\frac{1}{n}S^tSAA^t\frac{1}{n}S^tSP \\ &= P^tI_{k\times k}AA^tI_{k\times k}P = I_{p\times p}. \end{aligned}$$

Hence, A_0^t is the left inverse to A_0, and $tr(A_0^tA_0) = p$. Notice that we can rewrite the cost function defined in (3.3) as:

$$\begin{aligned}
\mathcal{F}(A) &= \sum_{i=1}^{n}\sum_{j=1}^{p}\big(z_{ij} - \sum_{l=1}^{k} A_{jl} * s_{il}\big)^2 \\
&= \sum_{i=1}^{n}\sum_{j=1}^{p}(z_{ij} - (A\mathbf{s}_i)_j)(z_{ij} - (A\mathbf{s}_i)_j) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{p}(\mathbf{z}_i - (A\mathbf{s}_i))_j(\mathbf{z}_i - (A\mathbf{s}_i))_j \\
&= \sum_{i=1}^{n}(\mathbf{z}_i - (A\mathbf{s}_i))^t(\mathbf{z}_i - (A\mathbf{s}_i)) \\
&= \sum_{i=1}^{n} tr\left((\mathbf{z}_i - A\mathbf{s}_i)^t(\mathbf{z}_i - A\mathbf{s}_i)\right).
\end{aligned}$$

Now, we evaluate the loss function (3.3) at A_0:

$$\begin{aligned}
\mathcal{F}(A_0) &= \sum_{i=1}^{n} tr\left((\mathbf{z}_i - A_0\mathbf{s}_i)^t(\mathbf{z}_i - A_0\mathbf{s}_i)\right) \\
&= \sum_{i=1}^{n} tr\left(\mathbf{z}_i^t - \mathbf{s}_i^t A_0^t)(\mathbf{z}_i - A_0\mathbf{s}_i)\right) \\
&= \sum_{i=1}^{n} tr\left(\mathbf{z}_i^t\mathbf{z}_i - \mathbf{s}_i^t A_0^t\mathbf{z}_i - \mathbf{z}_i^t A_0\mathbf{s}_i + \mathbf{s}_i^t A_o^t A_0\mathbf{s}_i\right) \\
&= n\big(tr(\frac{1}{n}\sum_{i=1}^{n}\mathbf{z}_i^t\mathbf{z}_i) - tr(\frac{1}{n}\sum_{i=1}^{n}\mathbf{s}_i^t A_0^t\mathbf{z}_i) - tr(\frac{1}{n}\sum_{i=1}^{n}\mathbf{z}_i^t A_0\mathbf{s}_i) + tr(\frac{1}{n}\sum_{i=1}^{n}\mathbf{s}_i^t A_0^t A_0\mathbf{s}_i)\big) \\
&= n\big(tr(\frac{1}{n}\sum_{i=1}^{n}\mathbf{z}_i^t\mathbf{z}_i) - tr(A_0^t\frac{1}{n}\sum_{i=1}^{n}\mathbf{s}_i^t\mathbf{z}_i) - tr(A_0\frac{1}{n}\sum_{i=1}^{n}\mathbf{s}_i\mathbf{z}_i^t) + tr(A_0^t A_0\frac{1}{n}\sum_{i=1}^{n}\mathbf{s}_i\mathbf{s}_i^t)\big) \\
&= n\big(k - tr(A_0^t A_0)\big), \text{ recall that } tr(A_0^t A_0) = p \\
&= n(k-p).
\end{aligned}$$

This result holds no matter which p independent components out of the original k are retained in the truncated vectors $\mathbf{s}$. So, working on the whitened data, the information loss depends only on the number of compo-

nents that are excluded from the model and not on which components these are.

Fortunately, when we return to the original scale of the data, i.e. do the reverse transformation to (3.2), we see that which components are selected does make a difference.

For the minimization results above to hold the data does not have to be whitened, it can just be scaled. Denote $\mathbf{y}_i$ as the original data that has been centered and scaled, but not whitened. i.e $\mathbf{y}_i = K^{-1}(\mathbf{x}_i - \mu)$, where K is a diagonal matrix of the standard deviations of each components of $\mathbf{x}_i$. The covariance matrix of $\mathbf{y}_i$ here will have ones on the main diagonal but does not have to have zero off-diagonal elements (the correlations of the components of $\mathbf{x}$ remain unchanged). In the exact same way it can be shown that in order to minimize the loss function

$$\mathcal{F}(A) = \sum_{i=1}^{n} \|\mathbf{y}_i - A\mathbf{s}_i\|_2^2 \tag{3.12}$$

One should choose A as $A_0 = \frac{1}{n}\sum_{i=1}^{n} \mathbf{y}_i \mathbf{s}_i^t$. The results for the minimization will still hold, but A_0 will no longer be semi-orthogonal and left-invertible, also the value of $\mathcal{F}(A_0)$ will change.

Overall, we conclude that given $\mathbf{s}_i$ as the truncated independent components and $\mathbf{x}_i$ as the original data, one can find a matrix A_0 that will minimize the loss function $\mathcal{F}(A)$ in equation (3.12) as $A_0 = \frac{1}{n}\sum_{i=1}^{n} \mathbf{y}_i \mathbf{s}_i^t$, where $\mathbf{y}_i$ are the scaled non-whitened original data. After the optimal weighting matrix is found for the particular (truncated) subset of the independent components, the subset can be updated and the process can be repeated. Also, we found out that which components we choose does not change the loss in case of the whitened data, but does change it if the data is only centered and scaled.

The methodology shown here allows to perform the dimensionality reduction and components selection without any prior knowledge of the number of independent components to be retained. First, we extract the full set of independent components. Then, for every possible combination and number

of components to be retained, we can easily compute the information loss as defined in (3.12), using the optimal non-square weighting matrix A_0. Once this is done, the researcher can choose the number and the combination of the independent components to be retained in the model in accordance to his or her tolerance of the information loss, thus modeling the data with fewer variables than are present in the original dataset.

3.2.1 Similarity to multivariate multiple regression

One can view the minimization problem in (3.4) as finding a Least Squares solution to a multivariate multiple linear regression problem, that can be written as:

$$Z = S \times A + \epsilon \tag{3.13}$$

Here Z is the $n \times k$ matrix of the whitened data, S is the $n \times p$ matrix of the reduced independent components, A is the $k \times p$ non-square weighting matrix and ϵ is the k-dimensional vector of errors. Or, similarly, replacing Z with Y, where Y is the matrix of centered and scaled, but not whitened data. The Least Squares solution to this problem is given as [12]:

$$\hat{A} = (S^t S)^{-1} S^t Z = \frac{1}{n} S^t Z = A_0^t. \tag{3.14}$$

Similarly, for the correlated data, the least squares estimate is:

$$\hat{A} = (S^t S)^{-1} S^t Y = \frac{1}{n} S^t Y. \tag{3.15}$$

This fact provides additional confidence that our chosen solution in fact minimizes the loss function in (3.3).

The rest of this thesis focuses on the application of the results above to the data obtained from VIVO Team Development and observing how the choice of the number of components retained in the model influences the amount of loss when reconstructing the original data.

Chapter 4

Component selection implementation and results

This chapter focuses on the implementation of the methodology described in the previous chapter. The techniques are applied to the VIVO Team Development data, described in Chapter 1. First, some exploratory analysis results are presented. Then, some standard dimension reduction techniques are applied to the dataset. Next, some issues with the implementation of the existing ICA algorithms are discussed. Following this, I show the results of implementation of the findings from Chapter 3 and the comparison of this method to an existing method of dimensionality reduction using ICA. Finally, I present some findings about the components that tend to be chosen by the algorithm and contrast them with some practices that are currently employed.

4.1 Some exploratory analysis results

Before proceeding with implementation of the method discussed in Chapter 3, I present some results of the exploratory analysis of the data. The dataset $X \in \mathbb{R}^{n \times k}$ for the ICA algorithm is assumed to contain n independent and identically distributed (i.i.d.) observations of the random vector $\mathbf{x} \in \mathbb{R}^k$. In case of the VIVO data, the number of features is $k = 18$, while the number of observations is $n = 278$. As mentioned in Chapter 1, each of the values in the dataset is measured on the same scale from 3 to 30, as each of the 18 variables is calculated as a sum of the scores for three survey questions, where the score for every question is an integer between 1 and 10.

In this section I will be working with the centered and scaled, but not

whitened data. It is obtained from the original dataset as follows:

$$Y = (X - \mathbf{m})K^{-1}$$

Here $\mathbf{m}$ is the vector of sample means of the columns of X and K is the matrix, the diagonal elements of which are the sample standard deviations of the columns of X and the off-diagonal elements are zero.

First, I present the estimated densities of the components of Y. In Figure 4.1 we can see that the original densities of the 18 variables are unimodal and mostly slightly left-skewed. The distributions exhibit no extreme observations, which is due to the nature of the data.

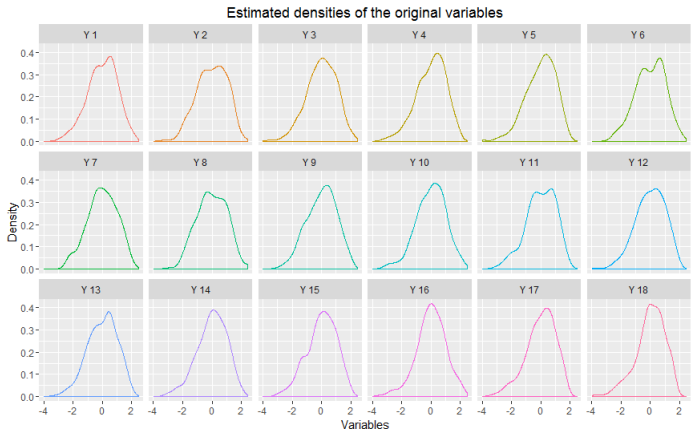

Figure 4.1: Estimated densities of the original variables (centered and scaled).

A correlation matrix is an important part of descriptive analysis of multivariate data. The correlation matrix of Pearson pairwise correlations for the VIVO data is presented in Figure 4.2. All the variables in the dataset are highly correlated, which is often a feature of survey datasets. In our case, all the survey questions are aimed at evaluating different sides of productivity, which are interrelated.

In the next section, we will apply Principal Component Analysis and

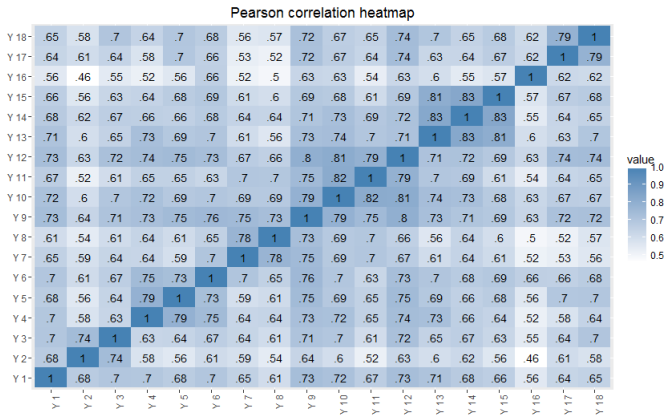

Figure 4.2: Pearson Correlation coefficients between the original variables. The colors indicate the strength of linear correlations.

Factor Analysis, widely used dimension reduction methods to the VIVO dataset and briefly discuss the results and the number of components each method indicates fit for the data.

4.2 Principal Component Analysis and Factor Analysis

In this section I show the application of "traditional", in a sense of popular and widely used, dimensionality reduction techniques.

First, we apply Principal Component Analysis to the VIVO data. This technique uses an orthogonal rotation to convert the correlated multivariate data into a set of uncorrelated principal components. The first principal component is chosen in such way that it has the highest possible variance, the subsequent components have the highest variance under the constraint of being orthogonal to the previously computed components. Dimensionality reduction using PCA refers to keeping fewer principal components than the original dimension of the data, which in our case is 18. The number of components to be kept can be decided, among other options, based on the

eigenvalues of the sample correlation matrix of the data or on how much variance of the data the selected number of components captures.

Table 4.1 shows the variance percentage attributable to each of the principal components (up to the 9th principal component). In our case, the first component holds 69% of the variance and clearly dominates over the other components, that contain less than 5% each.

Component	Standard dev	Proportion of Var	Cumulative Proportion
PC1	3.51	0.69	0.69
PC2	0.89	0.04	0.73
PC3	0.81	0.04	0.77
PC4	0.79	0.03	0.80
PC5	0.72	0.03	0.83
PC6	0.70	0.03	0.86
PC7	0.64	0.02	0.88
PC8	0.54	0.02	0.90
PC9	0.52	0.01	0.91

Table 4.1: The standard deviations, the proportion of variance and the cumulative proportions of variance associated with the principal components.

Figure 4.3 shows the loadings associated with the first three principal components. It can be observed that the first principal component has approximately equal loadings associated with each of the original variables, thus can be interpreted as a total questionnaire score. The signs of the principal components are non-unique, hence the negative loadings could be viewed as positive ones. The second and the third principal components have little variance associated with them and have distinct loadings for several of the original variables. They can be interpreted as contrasts, which is always the case when the first principal component includes all the original variables with the same sign.

Overall, Principal Component Analysis shows one dominating component, which includes all the original variables with approximately equal weights and accounts for 69% of the variance of the data.

After applying PCA, the natural extension is to apply Factor Analysis to the VIVO data. Factor analysis is a method that attempts to describe a

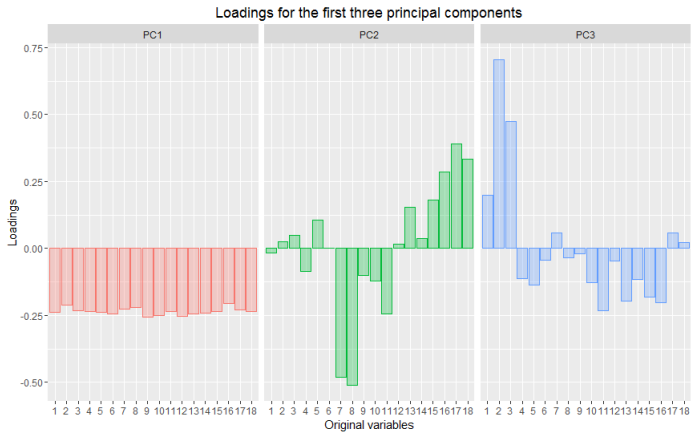

Figure 4.3: The loadings associated with the first three principal components.

multivariate dataset of correlated variables in terms of a set of unobserved orthogonal factors, the number of which is potentially smaller than the dimension of the original dataset. Exploratory Factor Analysis differs from PCA in the sense that it includes error term in the model and thus regards the PCA eigenvalues as inflated by errors. Unlike PCA, estimating the factors is an optimization procedure, aiming for the highest joint variance of the factors. Factor analysis aims to find a generative model for the data, while PCA simply chooses an orthogonal rotation that maximizes each of the principal components variances one after another.

Under the framework of factor analysis, Chi-square test is performed to test for ideal model fit. In other words, it tests the null hypothesis that the number of factors included in the model is sufficient. In case of the VIVO data, we fail to reject the ideal fit hypothesis starting from 7 factors. Together these 7 factors capture 80.5% of the variance of the data. Table 4.2 illustrates the percentage of variance associated with each of the factors.

Another option to determine the sufficient number of factors is using fit statistics to compare the observed correlation matrix and the structured correlation matrix based on p factors, $p = 1, 2, \ldots, k$.

We see that the first 5 factors each account for over 10% of the data

Factor	SS loadings	Proportion of Var	Cumulative Proportion
1	2.69	0.15	0.15
2	2.60	0.15	0.30
3	2.25	0.13	0.42
4	2.03	0.11	0.53
5	2.02	0.11	0.64
6	1.59	0.09	0.73
7	1.31	0.07	0.81

Table 4.2: The sums of squares, the proportions of variance and the cumulative proportions of variance associated with the factors.

variance and the percentages do not decrease as drastically as they do in case of PCA. The individual loadings of the first six factors are displayed in Figure 4.4. All of the factors include all of the original variables with non-zero weights.

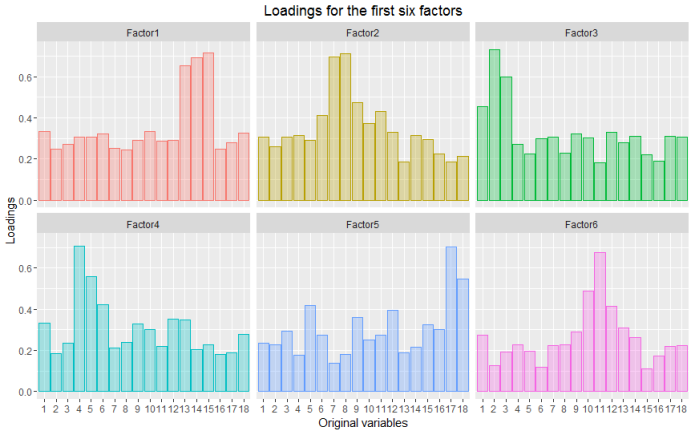

Figure 4.4: The loadings associated with the first six factors extracted using factor analysis.

In this section we applied the standard dimension reduction tools to the VIVO dataset, namely Principal Component Analysis and Factor Analysis. PCA indicates the presence of one dominating component, which includes all of the original variables with approximately equal weights. FA, in turn,

identifies that the data may be a contaminated mixture of at least 7 orthogonal factors, each of which include all of the original variables with non-zero (but non-equal) weights.

In the remainder of the chapter, I will apply dimensionality reduction using ICA, the method described in Chapter 3 of this thesis.

4.3 Choosing a suitable negentropy approximation

Before performing the component selection and dimension reduction, one needs to run the ICA algorithm to extract the independent components from the dataset.

The dataset $X \in \mathbb{R}^{n \times k}$ for the ICA algorithm is assumed to contain n i.i.d. realizations of the random vector $\mathbf{x} \in \mathbb{R}^k$. The data is assumed to be preprocessed as described in section 2.3 to form the input matrix Z as:

$$Z = X'VD^{-1/2}V^t, \tag{4.1}$$

where $X' = (x'_{ij})_{n \times k}$ and $x'_{ij} = x_{ij} - \frac{1}{n}\sum_{i=1}^{n} x_{ij}$ is the centered data matrix. D in equation (4.1) is the diagonal matrix of the eigenvalues of the sample covariance matrix computed as $\frac{1}{n}(X')^t X'$ and V is the matrix of its eigenvectors.

As discussed in the previous chapter, for now we retain as many components as there are features in the original dataset, thus, by making the reverse transformation $Z = \tilde{S}A^t$ we are able to reconstruct the original data perfectly. Here $\tilde{S}$ is the full matrix of the estimated independent components , $\tilde{S} \in \mathbb{R}^{n \times k}$ and A is the square independent components loadings matrix, $A \in \mathbb{R}^{k \times k}$.

I used the FastICA algorithm to extract the independent components from the VIVO survey data. This algorithm looks for an orthogonal rotation of the data that maximizes statistical independence of the resulting components. Note that maximizing statistical independence here in achieved

by maximizing non-gaussianity as described in Chapter 2, i.e. the algorithm maximizes a measure of non-gaussianity for each of the resulting components. As mentioned in Chapter 2, this algorithm provides two choices of negentropy approximation as measured of non-gaussianity, *exponential* and *logcosh*. This section discusses which approximation is better suited for the VIVO dataset, and, additionally, reviews some of the properties of the FastICA algorithm, that came to light while running this analysis.

The FastICA algorithm works by updating the orthogonal rotations of the data, so that the resulting components show the highest negentropy. That is, the algorithm iteratively estimates the matrix W in (2.4), which is the inverse of A in equation (2.3), the independent components loadings matrix. Given the iterative procedure of the FastICA algorithm, an initial point for W needs to be specified. Usually the the starting W is chosen at random (in this work, `rnorm` was used to generate the initial values). Notice that the initial guess for W does not have to be orthogonal as Gram-Schmidt orthogonalization is performed at every step of the FastICA algorithm, as described in section 2.5.

If two subsequent values for W are close enough, then the algorithm converges and uses the output estimate of W to compute the full matrix of independent components as:

$$\tilde{S} = ZW^t. \tag{4.2}$$

The algorithm implemented in the R `fastIca` package by default contains no convergence checks at the output, which means that an inverse loadings matrix will be output every time the algorithm is run. The algorithm stops in the following cases:

- If two subsequent values of W are close enough, i.e. if the norm of their difference is below the pre-specified tolerance level. In this case the algorithm converges and the latest value of W is output;
- If the number of iterations reached the pre-specified maximum number of iterations provided by the user. In this case, the algorithm outputs

W value from the last iteration.

I built in a simple convergence indicator into the existing *fastICA* function, which checks whether the output value of W was produced on the maximum iteration or prior to that. The algorithm is assumed to converge if the output value of W was achieved before the maximum iteration. Then I investigated which negentropy approximation provided convergence in most cases for the VIVO dataset. It can be seen in Figure 4.5, in the case of the VIVO data, *logcosh* algorithm with the parameter $\alpha = 1$ showed the best convergence results.

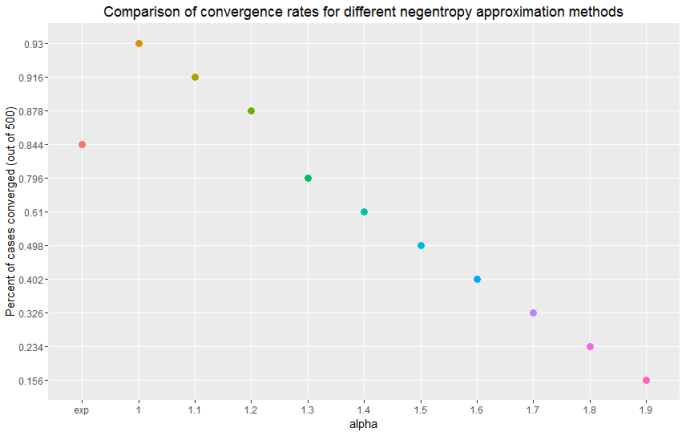

Figure 4.5: Percentage of the converged cases of the FastICA algorithm in application to the VIVO data obtained on 500 runs starting from random initial values. The first point shows the percentage of converged cases for *exponential* negentropy approximation, all other — *logcosh* approximation with varying values of the parameter α.

The FastICA algorithm is very sensitive to the starting point, which means that the returned estimates of the matrix W will vary depending on the initial guess. This holds true even when the algorithm converges with extremely low tolerance levels. Since different initial values provide different output estimates of W, the estimated $\tilde{S}$ will vary (4.2). When all the independent components are retained, this fact is not important, because the variations in $\tilde{S}$ will correspond to the variations in A and $Z = \tilde{S}A^t$

will still hold. This happens because, as both A and $\tilde{S}$ are estimated, any change in the estimate of $\tilde{S}$ will be compensated by a change in the estimate of A. However, the loss incurred when retaining only several independent components will vary depending on the initial guess. The following section will show, among other results the magnitude of this variance.

4.4 Implementation results

This section states the main results of implementation of the method discussed in Chapter 3. The dataset obtained from VIVO Team Development contains 18 variables measured 278 times. We denote this dataset as $X \in \mathbb{R}^{278\times 18}$. The implementation of the methodology consists of the following steps:

1. Center the dataset, i.e. compute $X' = (x'_{ij})_{278\times 18}$ by subtracting the sample means of each column of X as $x'_{ij} = x_{ij} - \frac{1}{n}\sum_{l=1}^{n} x_{lj}$. Record the column means of X as $\mathbf{m}$;

2. Standardize the dataset, i.e. compute $Y = X'K^{-1}$, where K is the diagonal matrix of sample standard deviations of the columns of X'. Note that this step is not compulsory for the FastICA procedure, but the output matrix Y is used in step 6;

3. Whiten the dataset, computing $Z = X'VD^{-\frac{1}{2}}V^t$, where D is the diagonal matrix of the eigenvalues of the sample covariance matrix computed as $\frac{1}{n}(X')^tX'$ and V is the matrix, the columns of which are the eigenvectors of the sample covariance matrix;

4. Run the FastICA algorithm on Z, keeping all the 18 variables. The initial guess for the matrix W is 18×18 matrix of independent random draws from a standard normal distribution;

5. Check whether the *fastICA* algorithm converged and, if so, use the output matrix W to compute an estimate of the full matrix of independent components as $\tilde{S} = ZW^t$, thereby obtaining $\tilde{S} \in \mathbb{R}^{278\times 18}$, the estimates of the 18 independent components;

6. For each p from 2 to 18:
   - Compute $\binom{18}{p}$ possible reduced (truncated) matrices S, and, consequently $A_0 = \frac{1}{n} Y^t S$ for each of the possible S matrices;
   - Compute the estimates of $\hat{Y}$ based on all possible truncated matrices S as $\hat{Y} = SA_0^t$;
   - Return to the original scale of the data by $\hat{X} = \hat{Y}K + \hat{\mu}$ and compute the loss values $\mathcal{F} = \sum_{i=1}^{n} \|\hat{X}_i - X_i\|_2^2$, where $\hat{X}_i$ represents the i-th row of $\hat{X}$ and X_i stands for the i-th row of X. For each p, $\binom{18}{p}$ of the loss values should be computed;
   - Record the combination of the independent components that provides the lowest value of loss and the value of loss at that combination, then proceed for the next p;
7. As output, for each possible number of the retained components, display the combination of the independent components that provided lowest value of loss and the value of the loss function with only the retained components included.

The steps above produce a vector of minimum losses across the combinations of components when retaining any number from 2 to 18 independent components for each random starting point. The value of loss decreases as the amount of independent components retained increases and the loss is zero when all of the 18 components are retained in the model. Such vector obtained for a single run of the procedure above can be seen in Figure 4.6.

Figure 4.7 shows mean squared error (MSE) for the dataset reconstructed using different numbers of retained components corresponding to the losses shown in Figure 4.6. The mean squared error is computed as:

$$MSE = \frac{1}{n}\frac{1}{k}\sum_{i=1}^{n}\sum_{j=1}^{k} \|x_{ij} - \hat{x}_{ij}\|_2^2. \tag{4.3}$$

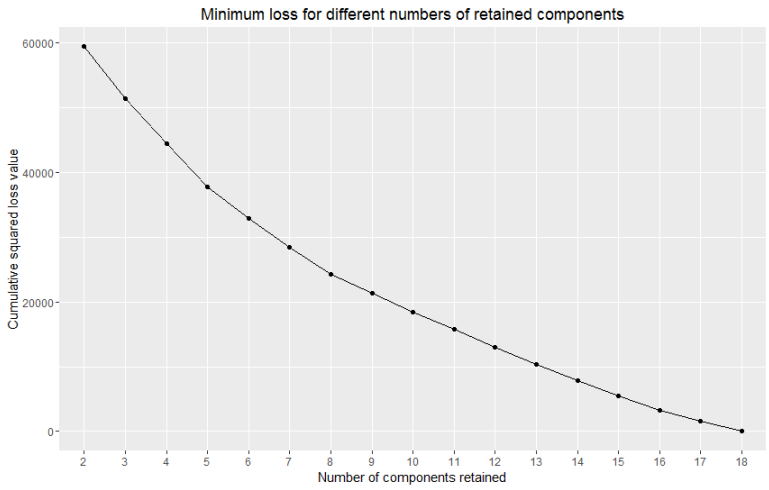

Figure 4.6: The minimum of the loss function across all possible retained independent components for each number of independent components retained in the model.

Here x_{ij} is the j-th element of the i-th row of the original dataset and $\hat{x}_{ij}$ is the j-th element of the i-th row of the reconstructed dataset obtained as described in the point 6 of the list above.

Figure 4.7 provides the reader with a chance to access the magnitude of MSE compared to the sale of the original data. Each value in the original dataset is on a scale from 3 to 30.

As was mentioned in the previous section, the FastICA methodology is sensitive to the initial value for W. Therefore, different starting points provide different loss and MSE values. Figure 4.8 displays the boxplots of MSE values based on varying initial values (in case of my implementation, varying seeds). I used pseudo-random number generator seeds from 2001 to 2030. On the seed 2003 the FastICA algorithm failed to converge, therefore this point has been excluded. The variation of the mean squared errors is high when only a few components are included and decreases as the number of components to be retained increases.

Additionally, Figure 4.9 displays the lines representing the mean squared

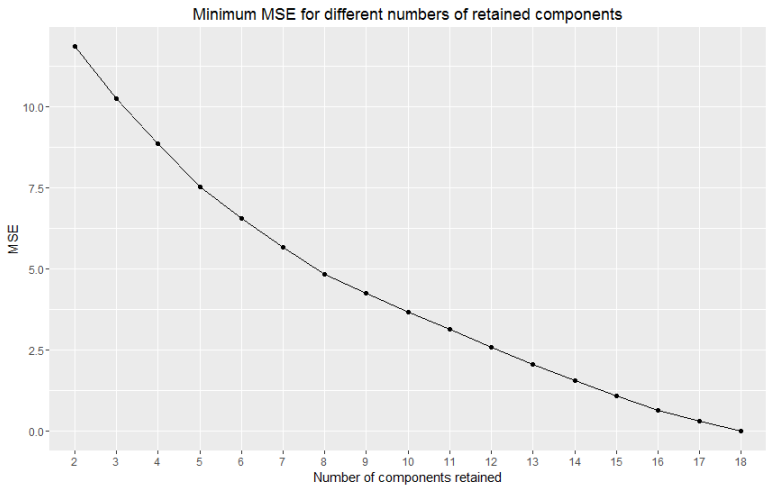

Figure 4.7: The minimum MSE across all possible combinations of the retained independent components for each number of independent components retained in the model. Each value in the original dataset is on a scale from 3 to 30.

errors for each starting point. This visual gives the reader a chance to assess whether some starting points provide the independent components estimates that are consistently better in terms of data reconstruction, no matter how many components are retained in the model. For example, seed 2029 seems to consistently produce the highest MSE values, while seed 2020 often corresponds to the lowest MSE. The detailed comparison of the components extracted starting from these two seeds is discussed in the next section.

As mentioned in Chapter 3, an existing method of dimensionality reduction includes first using Principal Component Analysis to select a number of linear combinations of the original variables and then implementing ICA on the retained principal components as described in section 2.3. The results of this method are shown in Figure 4.10 compared to the method explored in my research. The comparability of these approaches may be hindered by the fact that, while in my approach ICA is run once for each seed and the independent components from this run are then used to evaluate the losses,

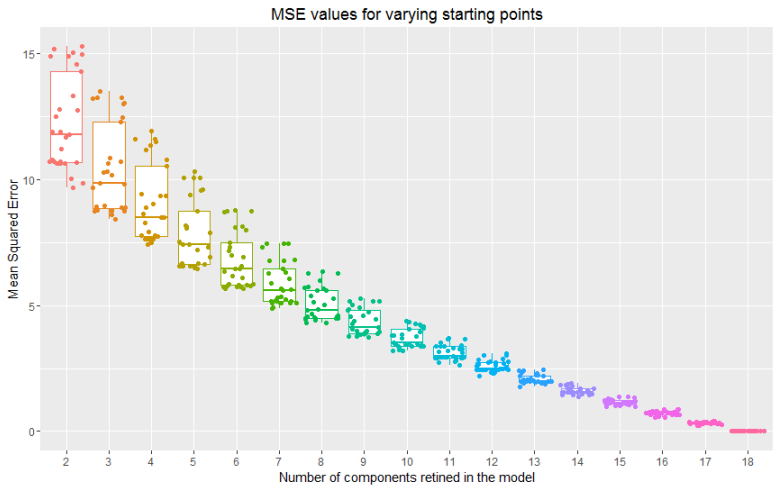

Figure 4.8: The minimum MSE values for each number of the retained components achieved on various starting points for the 29 seeds.

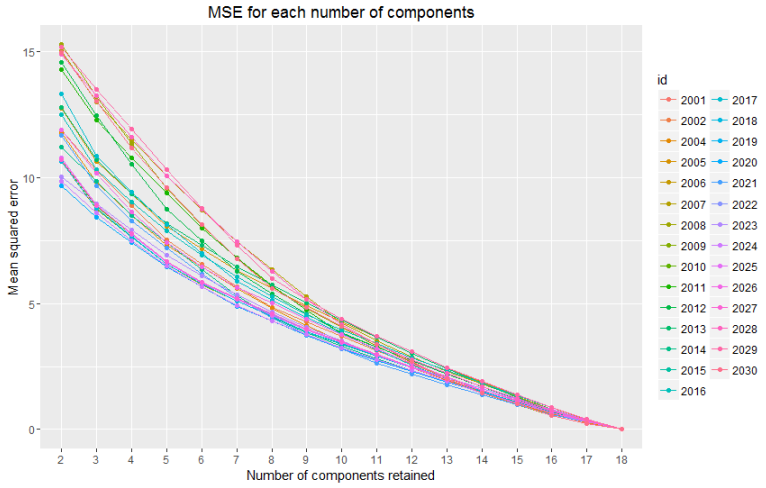

Figure 4.9: The minimum MSE values for each number of components retained achieved on various starting points for the 29 seeds shown by the seed number.

in case of PCA prior to ICA, ICA has to be run for each number of retained

components separately and there is no guarantee that each of these runs will converge. To obtain a more comparable plot, I used the same seeds as for the method discussed in this work. It is clear from Figure 4.10 that the use of PCA before applying ICA to the data reduces the losses incurred when reconstructing the data by approximately a half for any number of retained components (except when all the components are retained). This may be explained by the fact that, while discarding several principal components corresponding to the smallest eigenvalues generates some information loss, building a full ICA model on the retained principal components means that we can reconstruct those principal components perfectly with the inverse transformation (recall, no information is lost when the dimensionality of the components matches the dimensionality of the data). Hence, the entire amount of information lost in case of using PCA corresponds to discarding the lowest eigenvalues. PCA by definition gives the best L_2 approximation on linear transformations from the original data. This is also the reason behind the fact that the losses for PCA prior to ICA method do not vary depending on the starting point. Any change in the estimated values of independent components is balanced with the change in the corresponding elements of the weighting matrix A, which means we can reconstruct the principal components perfectly, thus the losses in this method only include the deterministic portion that appears when discarding the small eigenvalues at the PCA stage.

To summarize the application outcomes, the results obtained when using the method developed in course of my research highly depend on the starting point and provide reasonable loss values that decrease faster than the number of independent components retained in the model increases.

Because the results differ so much based on the starting point, I concentrated on the foundations and conceptual applications of the methodology and not the interpretation relative to the VIVO dataset. In principle, the inverse loadings of the independent components that are found to produce the smallest loss when retained in the model could be interpreted as the loadings of the 18 measurements derived from the VIVO questionnaire.

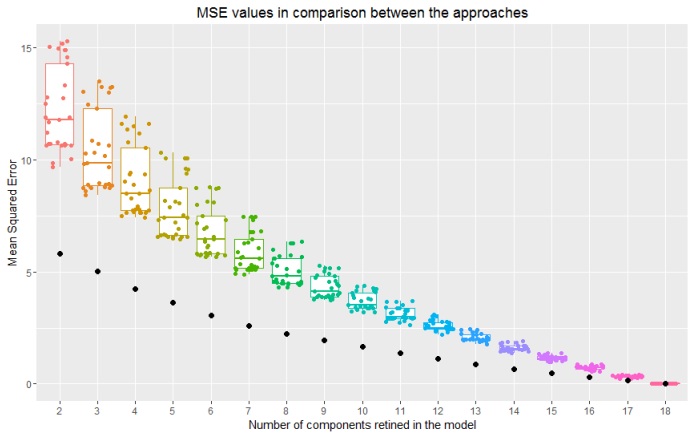

Figure 4.10: The comparison of the mean squared errors obtained by the method discussed in this thesis (boxplots) and MSEs obtained using an existing approach of first applying PCA. In case of PCA application the losses do not differ depending on the starting point.

4.5 Observations about the retained components

This section includes some observations about the components retained in the model by using the method discussed in this thesis. I will not go into detail concerning these observations, but they may be relevant to possible future research on the ICA methodology.

As mentioned in the previous sections, the estimates of the independent components vary with the initial guess of the inverse loadings matrix. Thereby, different random starts produce different estimates of the independent components. The densities of the extracted components starting at different points are compared in the figures below. The plots display the extreme cases, that is, the random starts with the highest and the lowest average MSE across the number of components retained in the model.

Figure 4.11 shows the estimated densities of the independent components for seed number 2029, which provided the highest loss estimates for most numbers of components retained in the model across the 29 seeds that the algorithm was run for.

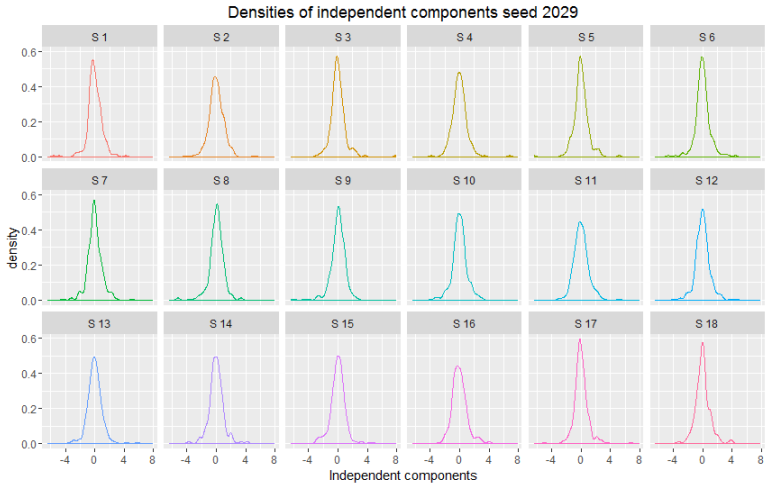

Figure 4.11: Estimated densities of the 18 independent component drawn from the VIVO data, seed 2029.

Figure 4.12 shows the estimated densities for components for the seed 2020, which provided the lowest loss values for most numbers of components retained in the model. As mentioned in Chapter 2, the order of the components is somewhat ambiguous, thus changes with every initial guess. However, in Figure 4.12 we see that the distributions of some of the independent components have no analog in Figure 4.11. For example, the component S_2 seems to have no counterpart. The ambiguities of ICA described in Chapter 2 state that one cannot determine the order of the components and their sign. None of these ambiguities should affect the distribution of the components.

As discussed in Chapter 2, the FastICA algorithm aims to maximize the statistical independence of the output components by maximizing their non-gaussianity. One measure of non-gaussianity is kurtosis, which is non-robust in general, but can nevertheless serve as an indicator. Since neither the initial data (because of the nature of survey data) nor the independent components exhibit extreme observations, we can use kurtosis, defined in section 2.4, to compare the components obtained starting from random seeds

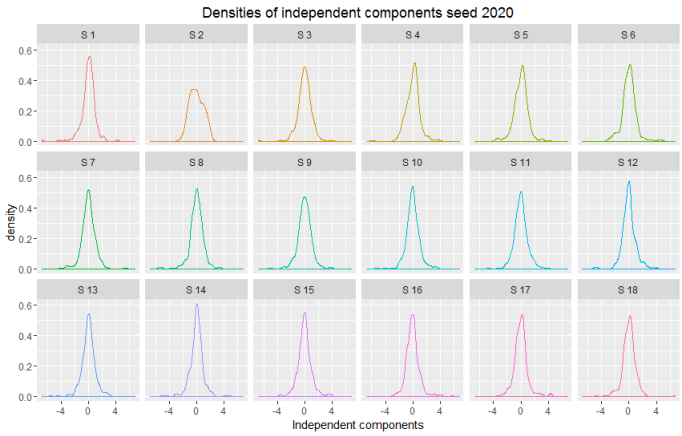

Figure 4.12: Estimated densities of the 18 independent component drawn from the VIVO data, seed 2020.

2020 and 2029, as shown in Table 4.3, which is an extract from Table A.1 given in Appendix A, showing the loss and kurtosis values for all seeds:

Seed	Minimum Kurtosis	MSE 2 components	Average MSE
2020	2.16	9.66	3.77
2029	4.74	15.19	5.67

Table 4.3: The comparison of the minimum kurtosis, the MSE incurred when retaining only 2 independent components out of the original 18 and the average MSE across the possible number of retained components for seeds 2020 and 2029.

If we consider the minimum sample kurtosis among the extracted components (among the columns of $\tilde{S}$) for each of the 29 seed used in my application, shown in Appendix A, we see that roughly in half of the cases the minimum kurtosis value is below 3, which means that at least one of the components is subgaussian (we will refer to these cases as subgaussian), while in other cases, the minimum kurtosis value is above 3, which means all the extracted components are supergaussian (these cases are referred to later as supergaussian). Note that subgaussian distributions are character-

ized by a low peak compared to gaussian distributions and a strong tail decay property, which means their tails decay at least as fast as the tails of gaussian distributions. Supergaussian distributions, on the contrary, have higher density at the peak compared to gaussian family and often heavier tails. Density plots depicting examples of both types of distributions in comparison to a gaussian distribution are shown in Figure 4.13.

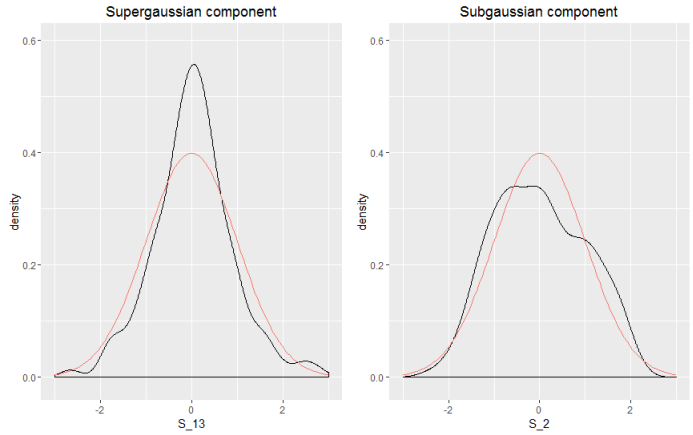

Figure 4.13: An example of a supergassian component S_{13} (left) and a subgaussian component S_2 (right) in the seed 2020.

If we plot the mean squared error obtained when reconstructing the dataset retaining only 2 of the 18 components against the minimum kurtosis value of the independent components, we notice an interesting tendency. The runs where at least one of the independent components is subgaussian produce lower MSE values than those where all the components are supergaussian, as shown in Figure 4.14. I do not have a clear explanation for why this tendency is taking place or whether it is specific to the dataset I am using or persists across different datasets, this may be a potential area of future research.

Another observation is that the subgaussian cases produce significantly lower loss values for the numbers of components retained in the model up to 14, after which the differences are no longer statistically significant. I per-

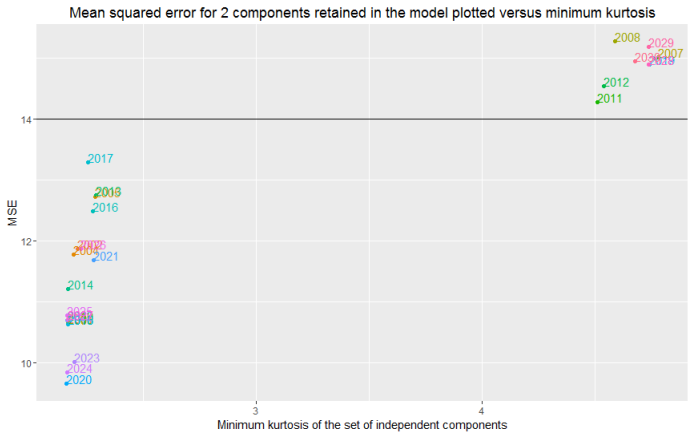

Figure 4.14: Mean squared errors when 2 components are retained out of the original 18 plotted against the minimum kurtosis of the independent components for each seed number. The points are labeled with the corresponding seed numbers.

formed two-sample t tests comparing the mean losses for the subgaussian cases and the supergaussian cases across the number of retained components. The p-values of the tests can be seen in Figure 4.15. The plot shows the cutoff value at 0.05, which does not take into account the multiple comparisons issue, but even with the conservative Bonferroni correction, the results are significant up to 12 components retained in the model.

The observations described in this section persist even when the ICA algorithm converges at a very low tolerance level. Hence, these observations may be indicating a tendency that is characteristic to the component selection method described in this thesis. Possibly, the ICA solution is not unique in case of the VIVO data. As mentioned in Chapter 2 the solutions are not unique for multivariate Gaussian initial data, thus it may be possible that the VIVO data resembles a multivariate gaussian distribution enough for multiple solutions to exist.

Another possible conclusion from these observations is that while larger kurtosis indicates more departure from gaussianity, which is interesting in many applications [11], this approach does not provide the best selection of

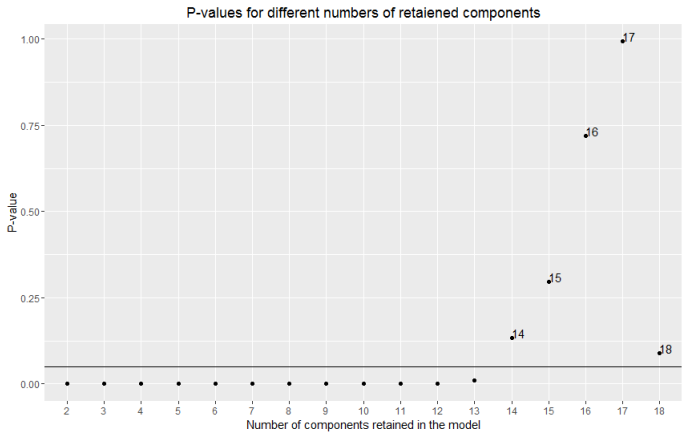

Figure 4.15: The p-values of 2-sample t tests comparing the total square loss incurred for the subgaussian and supergaussian cases averaged across the number of components retained in the model.

the independent components in terms of recreating the original data when retaining fewer components than the original dimensions of the data. Some other criteria may need to be used for the purpose of dimensionality reduction with ICA. One possible criteria is choosing the components that minimize information loss described in this thesis.

Chapter 5

Conclusion

This document provides the mathematical foundation to an approach to the component selection and dimension reduction using Independent Component Analysis. This method is applicable when the number of components to be retained in the model is not known prior to implementation and allows to choose the number of components to be retained based on the tolerance level for the information loss. The method focuses on choosing fewer components than the original dimension of the data that best approximate the initial dataset in terms of a squared Euclidean norm. Also the method offers a closed-from solution for estimating the optimal weighting matrix when not all of the independent components are retained in the model.

The methodology was applied to a business psychology dataset and the results have been provided and discussed. The results of proposed methodology have been compared to an existing method, the benefits and drawbacks of each method have been reviewed. Additionally, a few observations have been made about the independent components extracted from the data using different initial values. These observations may serve as a starting point for further research and development of theoretical and practical sides to dimension reduction using Independent Component Analysis.

Bibliography

[1] Barros, A. K., Mansour, A., and Ohnishi, N. (1998). Adaptive blind elimination of artifacts in ecg signals. *Proceedings of I and ANN*.

[2] Bartlett, M. S., Movellan, J. R., and Sejnowski, T. J. (2002). Face recognition by independent component analysis. *IEEE Transactions on neural networks*, 13(6):1450–1464.

[3] Bell, A. J. and Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159.

[4] Ben-Shalom, A., Werman, M., and Dubnov, S. (2003). Improved low bit-rate audio compression using reduced rank ica instead of psychoacoustic modeling. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 5, pages V–461. IEEE.

[5] Burel, G. (1992). Blind separation of sources: A nonlinear neural algorithm. *Neural networks*, 5(6):937–947.

[6] Cover, T. M. and Thomas, J. A. (1991). Entropy, relative entropy and mutual information. *Elements of Information Theory*, 2:1–55.

[7] Hansen, L. K. (2003). Ica if fmri based on a convolutive mixture model. In *Ninth Annual Meeting of the Organization for Human Brain Mapping,(hbm), New York, June*.

[8] Hyvärinen, A. (1998). New approximations of differential entropy for independent component analysis and projection pursuit. In Jordan, M. I.,

Kearns, M. J., and Solla, S. A., editors, *Advances in Neural Information Processing Systems 10*, pages 273–279. MIT Press.

[9] Hyvärinen, A. (1999). Sparse code shrinkage: Denoising of nongaussian data by maximum likelihood estimation. *Neural computation*, 11(7):1739–1768.

[10] Hyvärinen, A., Karhunen, J., and Oja, E. (2004). *Independent Component Analysis*. Adaptive and Cognitive Dynamic Systems: Signal Processing, Learning, Communications and Control. Wiley.

[11] Hyvärinen, A. and Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural networks*, 13(4):411–430.

[12] Johnson, R. A. and Wichern, D. W. (2014). *Applied multivariate statistical analysis*. Pearson Education Limited Essex.

[13] Jung, A. (2002). An introduction to a new data analysis tool: Independent component analysis.

[14] Jung, T.-P., Humphries, C., Lee, T.-W., Makeig, S., McKeown, M. J., Iragui, V., Sejnowski, T. J., et al. (1998). Extended ica removes artifacts from electroencephalographic recordings. *Advances in neural information processing systems*, pages 894–900.

[15] Kim, T.-H. and White, H. (2004). On more robust estimation of skewness and kurtosis. *Finance Research Letters*, 1(1):56–73.

[16] Kwon, O.-W. and Lee, T.-W. (2004). Phoneme recognition using ica-based feature extraction and transformation. *Signal Processing*, 84(6):1005–1019.

[17] Lathauwer, L. D. (1997). *Signal Processing by Multilinear Algebra*. PhD thesis, Faculty of Engineering, K.U. Leuven. unpublished thesis.

[18] Lörincz, A., Póczos, B., Szirtes, G., and Takács, B. (2002). Ockham's razor at work: Modeling of the"homunculus". *Brain and Mind*, 3(2):187–220.

[19] Makeig, S., Bell, A. J., Jung, T.-P., Sejnowski, T. J., et al. (1996). Independent component analysis of electroencephalographic data. *Advances in neural information processing systems*, pages 145–151.

[20] McSharry, P. E. and Clifford, G. D. (2004). A comparison of nonlinear noise reduction and independent component analysis using a realistic dynamical model of the electrocardiogram. In *Second International Symposium on Fluctuations and Noise*, pages 78–88. International Society for Optics and Photonics.

[21] Mohammad-Djafari, A. (2000). A bayesian approach to source separation. *arXiv preprint math-ph/0008025*.

[22] Nordhausen, K., Oja, H., and Ollila, E. (2016). Robust independent component analysis based on two scatter matrices. *Austrian Journal of Statistics*, 37(1):91–100.

[23] Toch, B., Lowe, D., and Saad, D. (2003). Watermarking of audio signals using independent component analysis. In *Web Delivering of Music, 2003. 2003 WEDELMUSIC. Proceedings. Third International Conference on*, pages 71–74. IEEE.

[24] Wang, J. and Chang, C.-I. (2006). Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis. *IEEE transactions on geoscience and remote sensing*, 44(6):1586–1600.

Appendix A

Table of loss values and lowest kurtosis values for various initial guesses

Seed	Minimum Kurtosis	MSE 2 components	Average MSE
2001	2.16	10.67	3.96
2002	2.21	11.87	4.41
2004	2.19	11.78	4.27
2005	2.29	12.73	4.86
2006	2.17	10.64	3.95
2007	4.78	15.01	5.56
2008	4.59	15.28	5.29
2009	2.17	10.65	3.96
2010	2.17	10.70	4.00
2011	4.51	14.28	5.18
2012	4.54	14.55	5.06
2013	2.29	12.76	4.89
2014	2.16	11.22	4.17
2015	2.17	10.63	3.96
2016	2.28	12.49	4.68
2017	2.25	13.29	4.66
2018	2.17	10.64	3.96
2019	4.74	14.89	5.53
2020	2.16	9.66	3.77

Seed	Minimum Kurtosis	MSE 2 components	Average MSE
2021	2.28	11.68	4.08
2022	2.16	10.71	4.01
2023	2.20	10.02	4.06
2024	2.16	9.85	3.81
2025	2.16	10.78	4.01
2026	2.22	11.87	4.41
2027	2.17	10.71	4.02
2028	4.74	14.89	5.53
2029	4.74	15.19	5.67
2030	4.67	14.94	5.27

Table A.1: The comparison of the minimum kurtosis, the MSE incurred when retaining only 2 independent components out of the original 18 and the average MSE across the possible number of retained components for all 29 seeds.