DOTTORATO DI RICERCA
in
SCIENZE COMPUTAZIONALI E INFORMATICHE
Ciclo XIX

Consorzio tra Università di Catania, Università di Napoli Federico II,
Seconda Università di Napoli, Università di Palermo, Università di Salerno

SEDE AMMINISTRATIVA: UNIVERSITÀ DI NAPOLI FEDERICO II

LARA GIORDANO

EMBEDDED INDEPENDENT COMPONENT ANALYSIS ON SINGLE CHANNEL MIXTURE

TESI DI DOTTORATO DI RICERCA

IL COORDINATORE
Prof. Aldo DE LUCA

DOTTORATO DI RICERCA
in
SCIENZE COMPUTAZIONALI E INFORMATICHE
Ciclo XIX

Consorzio tra Università di Catania, Università di Napoli Federico II,
Seconda Università di Napoli, Università di Palermo, Università di Salerno

SEDE AMMINISTRATIVA: UNIVERSITÀ DI NAPOLI FEDERICO II

LARA GIORDANO

EMBEDDED INDEPENDENT COMPONENT ANALYSIS ON SINGLE CHANNEL MIXTURE

TESI DI DOTTORATO DI RICERCA

IL COORDINATORE
Prof. Aldo DE LUCA

Author's address:
Lara Giordano
Dipartimento di Matematica e Applicazioni "R. Caccioppoli"
Università degli Studi di Napoli "Federico II"
Complesso Universitario di Monte Sant'Angelo,
Via Cintia - 80126 Napoli, Italy
e-mail: lara.giordano@na.infn.it

There are very few human beings who receive the truth,
complete and staggering, by istant illumination.
Most of them acquire it fragment by fragment, on a small scale,
by successive developments, cellularly, like a laborious mosaic.

Anais Nin; 1903-1977

A Carmine,
per quello che è stato,
per quello che è,
ma soprattutto per quello che sarà.

Ringraziamenti

Alla fine di un percorso come quello del Dottorato di Ricerca, nasce spontanea l'esigenza di ringraziare tutti coloro che hanno contribuito a rendere possibile questo successo.
Prima di tutto desidero ringraziare il Coordinatore ed il Collegio del Dottorato di Ricerca in Scienze Computazionali e Informatiche per l'opportunità che mi hanno offerto.
Desidero ringraziare il prof. Leopoldo Milano per i suoi preziosi insegnamenti sia scientifici che umani e il prof. Roberto Tagliaferri per la sua fiducia e le lunghe "chiacchierate scientifiche".
Ringrazio Angelo Ciaramella per la sua immensa disponibilità, la sua intuizione e la sua capacità di rendere tutto semplice; Antonio Eleuteri perchè qualsiasi sia la domanda lui ha sempre la risposta pronta.
Come non ringraziare tutti i componenti dell'allegra brigata del laboratorio Virgo. Proviamo a ricordarli tutti: Rosario che mi ha insegnato a lavorare con tutta me stessa; Fabio con il suo sanissimo spirito pratico; Iolanda e Adele che con la loro tenerezza e la loro amicizia hanno reso più leggeri anche i momenti difficili; Silvio, Saverio, Alessio che con la loro simpatia sono sempre pronti a farti sorridere. Ancora, in questo affollatissimo laboratorio ringrazio Fabrizio, Fausto, Luciano, Enrico, Ketino, Daniele e Simona.
Ringrazio tra i colleghi del dottorato Paolo per la sua amicizia e per la comune passione felina.
Ringrazio gli amici dei tempi dell'università Lara, Tony, Francesca, Francesco, Alessandro, sempre vicini e disponibili.
A questo punto non posso che ringraziare la mia famiglia: mia madre, mio padre e mio fratello Antonio. Se tutto questo, e molto altro ancora, è stato possibile lo devo soprattutto al loro amore, al loro sostegno e alla loro pazienza.
Ringrazio i miei zii Alfonso e Adua, per il loro amore incondizionato e tenerissimo.
Ringrazio, inoltre, i miei suoceri, Adele, Angelo e il piccolo Rodolfo.
Infine il pensiero conclusivo di questa pagina va a Carmine. Lo ringrazio perchè la sua presenza accanto a me mi dona gioia e serenità anche nei momenti più complicati. A lui è dedicato questo lavoro.

Abstract

Obtaining information from measured data is a general problem which is encountered in numerous applications and fields of science.
A goal of many data analysis methods is to transform the observed data into a representation which reveals the information contained in the data. Methods for obtaining such representations include principal component analysis, projection pursuit, and neural unsupervised learning methods.
In the last years, a great interest in the field of *signal processing* and of *neural networks* has been turned to the Independent Component Analysis (ICA). The main reason is because this method permits to obtain the separation of independent signals from mixture of them.
The ICA model based on Neural Networks (NNs) has been applied with good results to the Blind Source Separation (BSS). ICA is a statistical and computational technique for revealing hidden factors that underlie sets of random variables, measurements or signals. A more difficult problem in ICA is encountered if the number of the mixtures x_i is smaller than the number of independent components s_i. This means that the mixing system is not invertible: we cannot obtain the independent components (ICs) by simply inverting the mixing matrix $\mathbf{A}$. Therefore, even if we knew the mixing matrix exactly, we could not recover the exact values of the independent components. This is because information is lost in the mixing process.
The situation is often called ICA with overcomplete bases and we have to note that basic ICA methods cannot be used as such. In this situation, we have two different problems. First, how to estimate the mixing matrix, and second, how to estimate the realizations of the independent components. This is in stark contrast to the ordinary ICA, where these two problems are solved at the same time.
When the basis is overcomplete, the formulation of the likelihood is difficult, since the problem belongs to the class of missing data problems. Methods based on maximum likelihood estimation are therefore computationally rather inefficient. To obtain computationally efficient algorithms, strong approximations are necessary.
Our work focuses its attention on the problem of separating sources signals from a single observed mixture, exploiting new ideas for the solution of this problem.
We must note that this is a very important issue, because in practice this is the more

common situation to present: we have one sensors and multiple source that have been registered by that. We want to extract from the observation of the sensor each single source, separating them one from another.

At the moment, the literature on this topic is not so much and the technique proposed to accomplish this problem make a large use of *a priori* knowledge about the searched source or about the mixing process. This is clearly, not so good, because we lost one of the important feature of ICA system: blindness. We don't know anything about the sources or the mixing process. We only have the observations vector and from this we need to extract all the information needed.

In this work, we propose an interesting integration about two "field": dynamic system theory and non linear principal component analysis.

The first theory gives us the possibility to exploit the data vector, underlining the structure and the feature that are shift invariant. While the second theory gives us the separation algorithm.

At the hearth of this work there is the study and the realization of an algorithm capable to integrate this two theory for obtaining good separations also in the case of a single mixture. We show how it is possible to construct a NN architecture that has the structure of a non linear PCA NN, but where the parameters of the net are chosen from the dynamic system theory. This permits to analyze a single mixture as it would be a series of more mixtures shifted in time.

We give also some detail about the problem of ICA on a single mixture and why this is solvable by a Neural Network composed in this way.

At the end of this work, we present two important field of application of the proposed method: in astrophysics and in music. In the first case, we apply the method to data coming from Virgo Interferometer. This is an Italian-French experiment about the detection of gravitational waves.

We use the proposed method for the detection of gravitational wave signal in the output signal producted from the interferometric antenna. This is a challenge problem, because we are talking of a colored noise environment of really small amplitude and of signal with an very limited amplitude and relatively short in time.

From the application of the proposed method to some simulation, we got very good results obtaining the recognition of the signal at very low signal to noise ratio. Com-

paring that with the technique used for doing that, the matched filter, we can say to obtain good result in Signal to Noise Ratio terms, with an important feature that is the complete blindness of the source signal. We stress that the matched filter technique needs a template of the target signal and who can assure that we are supposing the right formulation for it?

Another important application field of the proposed method is in music signal analysis.

We found that, with the proposed method we can separate the harmonic from the sound of a single note for many musical instruments. Then, we also found that it is possible to separate from mixture of different music instruments, the single source in the case of single note, but also in the case of harmony.

We make several simulation for that field of application getting really good results of correlations between the original source and the extracted components.

In the next chapter we will give an overview of the problem and a specific view about the proposed method.

In particular, in chapter 1 we will give an introduction to the problem of independent component analysis from a statistical point of view and exploring the affinity of this technique with other similar.

In chapter 2, we will describe the principal algorithms used to accomplish classical independent component analysis; we divide this chapter in two part the first explain the contrast function used and the second explain the optimization technique used for each contrast function in order to get the algorithm for ICA.

In chapter 3, we will focus our attention to the case of single mixture independent component analysis, exploring the problem, its innate difficulty and the algorithm proposed in literature for accomplish this problem.

In chapter 4, we will describe the theory of dynamical systems and chaos. We explore the theory and the method to analyze time series and getting information regarding embedding dimension. We present also a method of separation based on the projection of the mixture in the phase space and then applying standard ICA algorithms. We present this method as a way of comparison for the ability of the proposed method.

In chapter 5, we will describe the Non Linear PCA network and the integration of this with the embedding dimension. We give some detail about the NN and we formulate

the new algorithm. We give also some theoretical explanation to the way of working of the new NN.

In chapter 6, we will present the application to the Virgo Interferometer data.

Finally in chapter 7, we will present the application to music mixture.

Contents

List of Figures

Chapter 1

Introduction to Independent Component Analysis

In this chapter, we give a general introduction to Independent Component Analysis (ICA). The features of the ICA method are shown from a statistical point of view. In the first part of the chapter we focus our attention to a general description of ICA model. Then we show the relation between ICA and classical statistical methods. At the end, we show an ICA application to solve the Blind Source Separation problem.

1.1 Introduction

Obtaining information from measured data is a general problem which is encountered in numerous applications and fields of science. A goal of many data analysis methods is to transform the observed data into a representation which reveals the information contained in the data. Methods for obtaining such representations include principal component analysis, projection pursuit, and neural unsupervised learning methods.
In the last years, a great interest in the field of *signal processing* and of *neural networks* has been turned to the Independent Component Analysis (ICA). The main reason is because this method permits to obtain the separation of independent signals from mixture of them.
The ICA model based on Neural Networks (NNs) has been applied with good results to the Blind Source Separation (BSS). ICA is a statistical and computational technique

for revealing hidden factors that underlie sets of random variables, measurements or signals.

In the model, the data variables are assumed to be linear or non-linear mixtures of some unknown latent variables and the mixing system is also unknown. The latent variable are assumed non-Gaussian and mutually independent and they are called independent components of the observed data. ICA can be seen as an extension of Principal Component Analysis (PCA) and of Factor Analysis (FA) [38, 41].

ICA is a much more powerful technique, however, capable of finding the underlying factors or sources when these classic methods fail completely. The data analyzed by ICA could originate from many different kinds of application fields, including digital images and document databases, as well as economic indicators and psychometric measurements.

The technique of ICA was first time introduced in the early 1980s in the context of the NNs modeling. In mid-1990s, some highly successful algorithms were introduced by several research groups, together with impressive demonstration on problems like the cocktail-party effect, where the individual speech waveforms are found from their mixture. ICA became one of the exciting new topics, both in the field of NNs, especially unsupervised learning and, more generally, in advanced statistics and signal processing [38, 41].

1.2 The Statistical Setting

A long-standing problem in statistics and related areas is how to find a suitable representation of multivariate data, which means transform the data so that is essential structure is made more visible or accessible. In neural computation, this fundamental problem belongs to the area of unsupervised learning, since the representation must be learned from the data itself without any external input from a supervising "'teacher"'. A good representation is also a central goal of many techniques in data mining and exploratory data analysis. In signal processing, the same problem can be found in feature extraction and also in the source separation. To explain the last case, let us assume that the data consists of a number of variables that we have observed together. Let us denote the number of variables by m and the number of observations by T.

We can then denote the data by $x_{i(t)}$, where the indices take the values *i=1,...,m* and *t=1,...,T*. The dimension m and T can be very large. A very general formulation of the problem can be stated as follows: what could be a function from a m-dimensional space to an n-dimensional space such that the transformed variables give information on the data that is otherwise hidden in the large data set. That is, the transformed variables should be the underlying *factors* or *components* that describe the essential structure of the data. It is hoped that these components correspond to some physical causes that were involved in the process that generated the data in the first place.
Let us denote by $\mathbf{x}$ an m-dimensional random variable; the problem is then to find a function $\mathbf{f}$ so that the n-dimensional transform $\mathbf{y}(t) = (y_1(t), ..., y_n(t))$ denoted by

$$\mathbf{y}(t) = \mathbf{f}(\mathbf{x}(t)) \tag{1.1}$$

has some desirable properties.

In most cases, we consider linear functions only, because in this case the interpretation of the representation is simpler and so is its computation. Thus, every component, say y_i, is expressed as a linear combination of the observed variables:

$$y_i(t) = \sum_j w_{ij} x_j(t) \tag{1.2}$$

for $i = 1,..,n$, $j = 1,...,m$, and where the w_{ij} are some coefficients that define the representation. The problem can then be rephrased as the problem of determining the coefficients w_{ij}. Using linear algebra, we can express the linear transformation in equation 1.2 as a matrix multiplication. Collecting the coefficients w_{ij} in a matrix $\mathbf{W}$, the equation becomes

$$\mathbf{y} = \mathbf{W}\mathbf{x} \tag{1.3}$$

where $\mathbf{y} = [y_1(t), ..., y_n(t)]'$ and $\mathbf{x} = [x_1(t), ..., x_m(t)]'$. A basic statistical approach consists of considering the $x_i(t)$ as a set of T realizations of m random variables. Thus each $x_i(t)$, *t=1,...,T* is a sample of one random variable; let us denote the random variable by x_i. In this framework, we could determine the matrix $\mathbf{W}$ by the statistical properties of the transformed components y_i.

1.3 Dimension Reduction Methods and Independence

One statistical principle for choosing the matrix $\mathbf{W}$ is to limit the number of components y_i to be quite small and to determine $\mathbf{W}$ so that the y_i contain as much information on the data as possible. This leads to a family of techniques as Principal Component Analysis (PCA) and Factor Analysis (FA) [46, 29].
Another principle that has been used for determining $\mathbf{W}$ is independence: the components y_i should be statistically independent. This means that the value of any one of the components gives no information on the values of the other components. In fact, in FA it is often claimed that the factors are independent, but this is only partly true, because FA assumes that the data has a Gaussian distribution. If the data is Gaussian, it is simple to find components that are independent, because for Gaussian data, uncorrelated components are always independent. However, the data often does not follow a Gaussian distribution and the situation is not as simple as those methods assume.
This is the starting point of ICA: we want to find *statistically independent* components, in the general case where the data is *non-Gaussian*.

1.3.1 Second Order Methods

The most popular methods for finding a linear transform as in equation 1.3 are second-order methods. This means methods that find the representation using only the information contained in the covariance matrix of the data vector $\mathbf{x}$. Of course, the mean is also used in the initial centering. The use of second-order techniques is to be understood in the context of the classical assumption of Gaussianity. The two classical second-order methods are PCA and FA [46, 29]. One might roughly characterize the second-order methods by saying that their purpose is to find a faithful representation of the data, in the sense of reconstruction (mean-square) error.

1.3.2 Higher-Order Methods

Higher-order methods use information on the distribution of $\mathbf{x}$ that is not contained in the covariance matrix. In order for this to be meaningful, the distribution of $\mathbf{x}$ must not be assumed to be Gaussian, because all the information of (zero mean) Gaussian

variables is contained in the covariance matrix.

For more general families of density functions, however, the representation problem has more degrees of freedom. Thus much more sophisticated techniques may be constructed for non Gaussian random variables. Indeed, the transform defined by second-order methods like PCA is not useful for many purposes where optimal reduction of dimension in the mean-square sense is not needed. This is because PCA neglects such aspects of non-Gaussian data as clustering and independence of the components (which, for non-Gaussian data, is not the same as uncorrelatedness). We shall review in the next sections three conventional methods based on higher-order statistics: projection pursuit, redundancy reduction and blind deconvolution.

1.4 Independent Component Analysis

Before to introduce the ICA method, we shall recall some basic definitions. Denote by y_1,y_2,...,y_m some random variables with joint density f(y_1,y_2,...,y_m). For simplicity, assume that the variable are zero mean. The variables y_i are (mutually) independent, if the density function can be factorized:

$$f(y_1,y_2,...,y_m) = f(y_1)f(y_2)...f(y_m) \tag{1.4}$$

where $f(y_i)$ denotes the marginal density of y_i. To distinguish this form of independence from other concepts of independence, for example linear independence, this property is sometimes called statistical independence. Independence must be distinguished from uncorrelatedness, which means that:

$$E\{y_i y_j\} - E\{y_i\} E\{y_j\} = 0 \quad \forall \; i \neq j \tag{1.5}$$

Independence is in general a much stronger requirement than uncorrelatedness. Indeed, if the y_i are independent, one has

$$E\{g_1(y_i)\, g_2(y_j)\} - E\{g_1(y_i)\} E\{g_2(y_j)\} = 0 \quad \forall \; i \neq j \tag{1.6}$$

for any measurable function g_1 e g_2 [61]. This is clearly a more constrained condition than that of uncorrelatedness. There is, however, an important special case where independence and uncorrelatedness are equivalent. This is the case when y_1,y_2,...,y_m

have a joint Gaussian distribution. Due to this property, ICA is not interesting (or possible) for Gaussian variables.
Now we shall define the problem of ICA. We shall only consider the linear case here, though non linear form of ICA also exist. In the literature, at least three different basic definitions for linear ICA can be found [38, 41], though the differences between the definitions are usually not emphasized. This is probably due to the fact that ICA is such a new research topic: most research has concentrated on the simplest one of these definitions. In the definitions, the observed m-dimensional random vector is denoted by $\mathbf{x} = (x_1, ..., x_m)^T$.
The first and most general definition is as follows:

Definition 1.4.1 *(General definition) ICA of the random vector $\boldsymbol{x}$ consists of finding a linear transform $\boldsymbol{s} = \boldsymbol{W}\boldsymbol{x}$ so that the components s_i are as independent as possible, in the sense of maximizing some function $\boldsymbol{F}(s_1, ..., s_m)$ that measures independence.*

This definition is the most general in the sense that no assumptions on the data are made, which is in contrast to the definitions below. Of course, this definition is also quite vague as one must also define a measure of independence for the s_i. One cannot use the definition of independence as in equation 1.4, because it is not possible, in general, to find a linear transformation that gives strictly independent components. The problem of defining a measure of independence will be treated in the next section. A different approach is taken by the following more estimation theoretically oriented definition:

Definition 1.4.2 *(Noisy ICA model) ICA of a random vector $\boldsymbol{x}$ consists of estimating the following generative model for the data:*

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{n} \tag{1.7}$$

where the latent variables (components) s_i in the vector $\boldsymbol{s} = (s_1, ..., s_n)^T$ are assumed independent. The matrix $\boldsymbol{A}$ is a constant $m \times n$ "mixing" matrix, and $\boldsymbol{n}$ is a m-dimensional random noise vector.

This definition reduces the ICA problem to ordinary estimation of a latent variable model. However, this estimation problem is not very simple and therefore the great majority of ICA research has concentrated on the following simplified definition:

Definition 1.4.3 *(Noise-free ICA model) ICA of a random vector* $\boldsymbol{x}$ *consists of estimating the following generative model for the data:*

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s} \tag{1.8}$$

where $\boldsymbol{s}$ *and* $\boldsymbol{A}$ *are defined as in the previous definition.*

Here the noise vector has been omitted. This is also the model introduced by Jutten and Hérault in their seminal paper [48], which was probably the earliest explicit formulation of ICA. Here, we shall concentrate on this noise - free ICA model definition. This choice can be partially justified by the fact that most of the research on ICA has also concentrated on this simple definition. Even the estimation of the noise - free model has proved to be a task difficult enough. The noise - free model may be thus considered a tractable approximation of the more realistic noisy model. The justification for this approximation is that methods using the simpler model seem to work for certain kinds of real data. It can be shown [26], in fact, that if the data does follow the generative model in equation 1.8, we have that the models described by 1.8 and 1.7 and the equation 1.6 become asymptotically equivalent, if certain measures of independence are used in Definition 1.4.1., and the natural relation $\mathbf{W} = \mathbf{A}^{-1}$ is used with $n = m$. In the figures 1.1, 1.2, 1.3, we show an illustration of ICA application on data sets characterized by different distributions. In figure 1.1, we show the case of superGaussian data, in figure 1.2 we show the case of subGaussian (uniform) data and in figure 1.3 we show the case of data with different distribution.

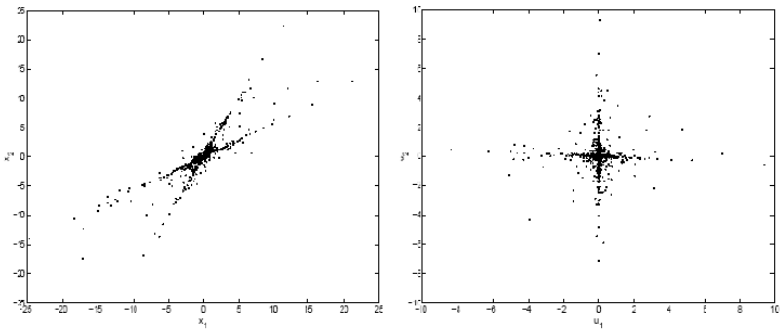


Figure 1.1: Scatter plot of 2 linearly mixed superGaussian data set (left), ICA applied to the data set (right).

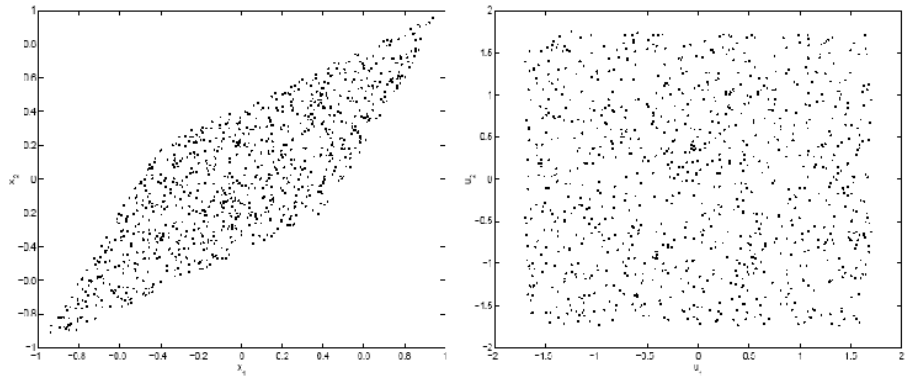


Figure 1.2: Scatter plot of 2 linearly mixed subGaussian (uniform) data set (left), ICA applied to the data set (right).

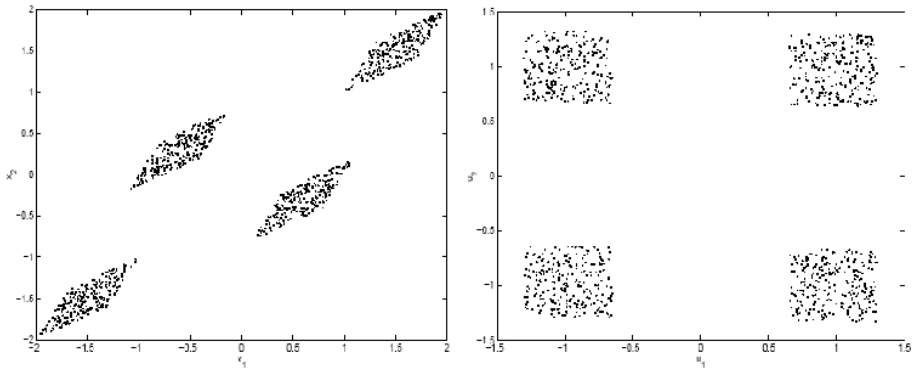


Figure 1.3: Scatter plot of 2 linearly mixed data set with different distribution (left), ICA applied to the data set (right).

1.4.1 Identifiability of the ICA Model

The identifiability of the noise - free ICA model has been treated in [26]. By imposing the following fundamental constraints (in addition to the basic assumption of statistical independence), the identifiability of the model can be assured:

1. All the independent components s_i, with the possible exception of one component, must be non-Gaussian;

2. The number of the observed linear mixtures m must be at least as large as the number of the independent components n;

3. The matrix $\mathbf{A}$ must be of full column rank.

Usually, it is also assumed that $\mathbf{x}$ and $\mathbf{s}$ are centered, which is equivalently in practice, to do not have restriction, as this can always be accomplished by subtracting the mean from the random vector. If $\mathbf{x}$ and $\mathbf{s}$ are interpreted as stochastic processes instead of simply random variables, additional restrictions are necessary. At the minimum, one has to assume that the stochastic processes are stationary in the strict sense. Some constraints of ergodicity with respect to the quantities estimated are also necessary [61]. These assumptions are fulfilled, for example, if the process is i.i.d. over time. After such assumptions, one can consider the stochastic process as random variable, as we do here.
A basic, but rather insignificant indeterminacy in the model is that the independent components and the columns of $\mathbf{A}$ can only be estimated up to a multiplicative constant, because any constant multiplying an independent component in equation 1.8 could be canceled by dividing the corresponding column of the mixing matrix $\mathbf{A}$ by the same constant. For mathematical convenience, one usually defines that the independent components s_i have unit variance. This makes the independent components unique, up to a multiplicative sign (which may be different for each component) [26]. The definitions of ICA given above imply no ordering of the independent components, which is in contrast to, e.g. PCA. It is possible, however, to introduce an order between the independent components. One way is to use the norms of the columns of the mixing matrix, which give the contributions of the independent components to the variances of the x_i. Ordering the s_i according to descending norm of the corresponding

columns of **A**, for example, gives an ordering reminiscent of PCA. A second way, is to use the non-Gaussianity of the independent components. Non-Gaussianity may be measured, for example, using one of the projection pursuit indexes or other contrast functions. Ordering the s_i according to non-Gaussianity gives an ordering related to projection pursuit.

The first restriction (non-Gaussianity) in the list above, is necessary for the identifiability of the ICA model [26]. Indeed, for Gaussian random variables mere uncorrelatedness implies independence, and thus any decorrelating representation would give independent components. Nevertheless, if more than one of the components s_i are Gaussian, it is still possible to identify the non-Gaussian independent components, as well as the corresponding columns of the mixing matrix.

On the other hand, the second restriction, $m \geq n$, is not completely necessary. Even in the case where $m < n$, the mixing matrix **A** seems to be identifiable [41] (though no rigorous proofs exist to our knowledge), whereas the realizations of the independent components are not identifiable, because of the non-invertibility of **A**. However, most of the existing theory for ICA is not valid in this case, and therefore we have to make the second assumption. Recent works on the case $m \geq n$, often called ICA with overcomplete bases can be found in [38, 41].

Some rank restriction on the mixing matrix, like the third restriction given above, is also necessary, though the form given here is probably not the weakest possible. As regards the identifiability of the noisy ICA model, the same three restrictions seem to guarantee partial identifiability, if the noise is assumed to be independent from the components s_i [38, 41]. In fact, the noisy ICA model is a special case of the noise-free ICA model with $m \geq n$, because the noise variables could be considered as additional independent components. In particular the mixing matrix **A** is still identifiable. In contrast, the realizations of the independent components s_i can no longer be identified, because they cannot be completely separated from noise. It would seem that the noise covariance matrix is also identifiable [38, 41].

1.4.2 Ambiguities of ICA

In the ICA model it is easy to see that the following ambiguities will necessary hold:

1. We cannot determine the variances (energies) of the independent components

2. We cannot determine the order of the independent components

For the first case the reason is that, both $\mathbf{s}$ and $\mathbf{A}$ being unknown, any scalar multiplier in one of the sources s_i could always be canceled by dividing the corresponding column $\mathbf{a}_i$ of $\mathbf{A}$ by the same scalar, say α_i:

$$\mathbf{x} = \sum_i \left(\frac{1}{\alpha_i} \mathbf{a}_i \right) (s_i \alpha_i) \tag{1.9}$$

As a consequence, we may quite as well fix the magnitudes of the independent components. Since they are random variables, the most natural way to do this is to assume that each has unit variance: $E\{s_i^2\} = 1$. Then the matrix $\mathbf{A}$ will be adapted in the ICA solution methods to take into account this restriction. Note that this still leaves the *ambiguity of the sign*: we could multiply an independent components by -1 without effecting the model.

For the second case the reason is that, again both $\mathbf{A}$ and $\mathbf{s}$ are unknown, we can freely change the order of the terms in equation 1.8, and call any of the independent components the first one. Formally, a permutation matrix and its inverse can be substituted in the model to give $\mathbf{x} = \mathbf{A}\mathbf{P}^{-1}\mathbf{P}\mathbf{s}$. The element of $\mathbf{P}\mathbf{s}$ are the original independent variables $\mathbf{s}_j$, but in another order. The matrix $\mathbf{x} = \mathbf{A}\mathbf{P}^{-1}$ is just a new unknown mixing matrix, to be solved by the ICA algorithms. In other words, we have that the separation matrix $\mathbf{W}$ is $\mathbf{W} = \Lambda\mathbf{P}$ for some permutation matrix $\mathbf{P}$ and some diagonal matrix Λ whose diagonal elements are ± 1.

1.5 Beyond Classical ICA: Overcomplete Bases

A more difficult problem in ICA is encountered if the number of the mixtures x_i is smaller than the number of independent components s_i. This means that the mixing system is not invertible: we cannot obtain the independent components (ICs) by simply inverting the mixing matrix $\mathbf{A}$. Therefore, even if we knew the mixing matrix exactly, we could not recover the exact values of the independent components. This is because information is lost in the mixing process.

The situation is often called ICA with overcomplete bases and we have to note that basic ICA methods cannot be used as such. In this situation, we have two different problems. First, how to estimate the mixing matrix, and second, how to estimate the

realizations of the independent components. This is in stark contrast to the ordinary ICA, where these two problems are solved at the same time.
When the basis is overcomplete, the formulation of the likelihood is difficult, since the problem belongs to the class of missing data problems. Methods based on maximum likelihood estimation are therefore computationally rather inefficient. To obtain computationally efficient algorithms, strong approximations are necessary.
Our work focus its attention on the problem of separating sources signals from a single observed mixture, exploiting new ideas for the solution of this problem.

1.6 Applications of ICA

The classical application of the ICA model is Blind Source Separation (BSS) [48]. We will speak in more detail about BSS in the next section. Another application of ICA is feature extraction [38, 41]. In this case the columns of $\mathbf{A}$ represent features and s_i is the coefficient of the i-th feature in an observed data vector $\mathbf{x}$. The use of ICA for feature extraction is motivated by the theory of redundancy reduction.
A less direct application of the ICA methods can be found in blind deconvolution.
Due to the close connection between ICA and projection pursuit on the one hand and between ICA and FA on the other, it should be possible to use ICA on many of the applications where projection pursuit and FA are used. These include (exploratory) data analysis in such areas as economics, psychology and other social sciences, as well as density estimation and regression.

1.7 Blind Source Separation

A classical example of BSS is the "cocktail party" problem. Assume that several people are speaking simultaneously in the same room. Then the problem is to separate the voices of the different speakers, using recordings of several microphones in the room.

More formally, we suppose to have a situation where there are a number of signals emitted by some physical objects or sources. Further, we assume that there are several sensors or receivers. These sensors are in different positions, so that each one records

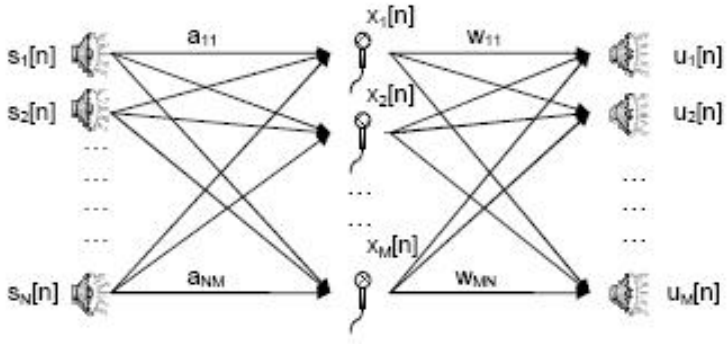


Figure 1.4: Example of the Cocktail Party Problem.

a mixture of the original source signals with slightly different weights. For the sake of simplicity of exposition, let us say there are three underlying source signals and also three observed signals. Denote by $x_1(t)$, $x_2(t)$ and $x_3(t)$ the observed signals, and by $s_1(t)$,$s_2(t)$ and $s_3(t)$ the original signals. The $x_i(t)$ are the weighted sums of the $s_i(t)$, where the coefficients depend on the distances between the sources and the sensors:

$$\begin{aligned} x_1(t) &= a_{11}s_1(t) + a_{12}s_2(t) + a_{13}s_3(t) \\ x_2(t) &= a_{21}s_1(t) + a_{22}s_2(t) + a_{23}s_3(t) \\ x_3(t) &= a_{31}s_1(t) + a_{32}s_2(t) + a_{33}s_3(t) \end{aligned} \tag{1.10}$$

The a_{ij} are constant coefficients that give the mixing weights. They are assumed *unknown*, since we cannot know the values a_{ij} without knowing all properties of the physical mixing system.

What we would like to do is to find the original signals from the mixtures $x_1(t)$, $x_2(t)$ and $x_3(t)$. This is the *Blind Source Separation problem*. Blind means that we know very little if anything about the original signals. We can safely assume that the mixing coefficients a_{ij} are different enough to make the matrix invertible. Thus there exists a matrix $\mathbf{W}$ with coefficients w_{ij} such that can separate the $s_i(t)$ as

$$\begin{aligned} s_1(t) &= w_{11}x_1(t) + w_{12}x_2(t) + w_{13}x_3(t) \\ s_2(t) &= w_{21}x_1(t) + w_{22}x_2(t) + w_{23}x_3(t) \\ s_3(t) &= w_{31}x_1(t) + w_{32}x_2(t) + w_{33}x_3(t) \end{aligned} \tag{1.11}$$

Such matrix $\mathbf{W}$ could be found as the inverse of the matrix that consists of the mixing coefficients in equation 1.11 if we knew those coefficients a_{ij}.

1.7.1 Source Separation Based on Independence

The question, that arises, is: how can we estimate the coefficients w_{ij} in equation 1.11? We use very general statistical properties. A surprisingly simple solution to the problem can be found by considering just the statistical independence of the signals. In fact, if the signals are not Gaussian, it is enough to determine the coefficients w_{ij} so that the signals

$$\begin{aligned} y_1(t) &= w_{11}x_1(t) + w_{12}x_2(t) + w_{13}x_3(t) \\ y_2(t) &= w_{21}x_1(t) + w_{22}x_2(t) + w_{23}x_3(t) \\ y_3(t) &= w_{31}x_1(t) + w_{32}x_2(t) + w_{33}x_3(t) \end{aligned} \tag{1.12}$$

are statistically independent. If the signal $y_1(t)$, $y_2(t)$ and $y_3(t)$ are independent, then they are equal to the original signals $s_1(t)$, $s_2(t)$ and $s_3(t)$. More formally, we have that

$$\mathbf{y} \approx \mathbf{s} = \mathbf{W}\mathbf{x} \tag{1.13}$$

Using just this information on the statistical independence, we can in fact estimate the coefficient matrix $\mathbf{W}$

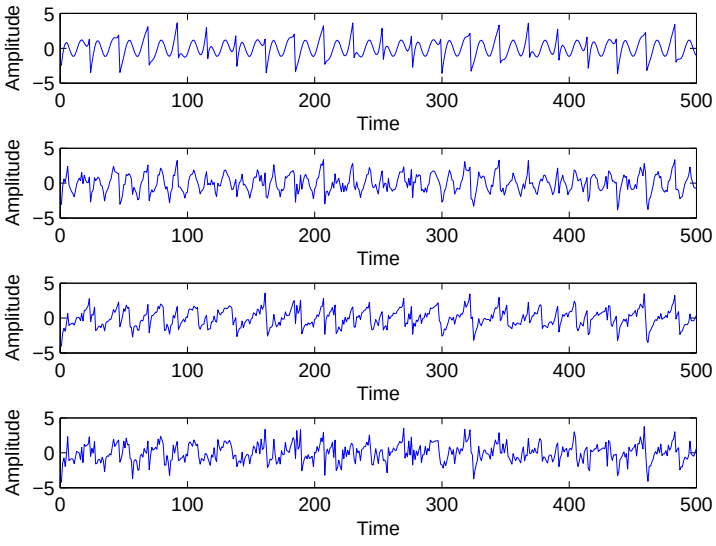


Figure 1.5: Example of source separation based on independence: mixed signals

for the signals in figure 1.5 that are the mixture of the signals in figure 1.6 . The

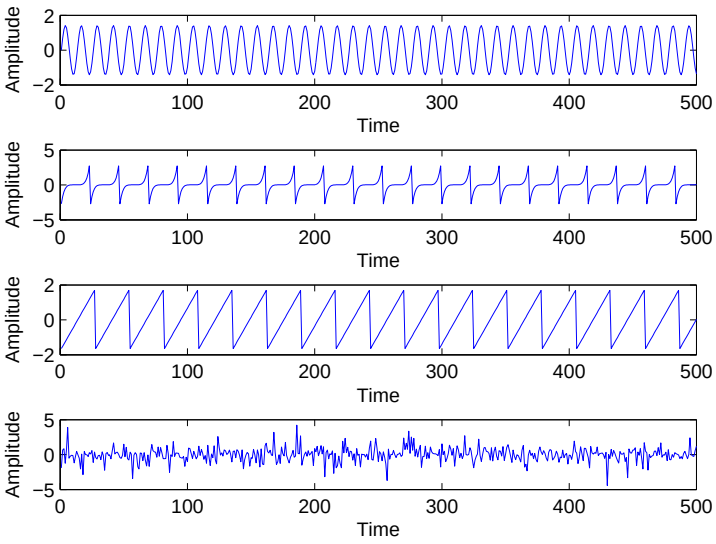


Figure 1.6: Example of source separation based on independence: source signals.

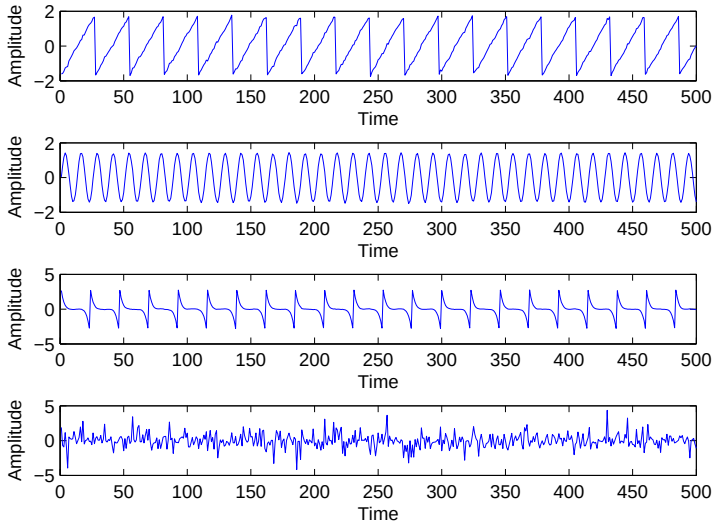


Figure 1.7: Example of source separation based on independence: separated signals

separated signals are shown in figure 1.7 . Formally, ICA consists of estimating both the matrix **A** and the $s_i(t)$, when we only observe $x_i(t)$.
Alternatively, we could define ICA as follows: find a linear transformation given by a matrix **W** so that the random variables y_i in equation 1.13 are as independent as possible.
We note that after estimating **A**, its inverse gives **W**.

1.8 History of ICA

The technique of ICA was introduced in the early 1980s by J. Hérault, C. Jutten and B. Ans [5, 28]. The problem first came up in 1982 in a neurophysiological setting.[48] A related field was higher-order spectral analysis, on which the first international workshop was organized in 1989. In this workshop, early papers on ICA by J. F. Cardoso and P. Comon [25] were given. Cardoso used algebraic methods, especially higher-order cumulant tensors, which eventually led to the Jade algorithm [19].
The work of the scientists in the 1980s was extended by, among other, A. Cichocki and R. Unbehauen, who first propose one of the presently most popular ICA algorithms [21, 24]. The "non-linear PCA" approach was introduced by E. Oja and J. Karhunen [50, 59]. ICA attained wider attention and growing interest after that A. J. Bell and T. J. Sejnowski published their approach based on infomax principle [10, 9] in the mid-90s. This algorithm was further refined by S. I. Amari and his co-workers using the natural gradient [4] and its fundamental connections to maximum likelihood estimation. In 2001, A. Hyvärinen, J. Karhunen, E. Oja presented the fixed-point algorithm or FastICA algorithm [39, 41] which has contributed to the application to large scale problems due to its computational efficiency.
A recent trend in BSS / ICA is to consider problems in the framework of matrix factorization or more general signals decomposition with probabilistic generative and tree structured graphical models and exploit a *priori* knowledge about true nature and structure of latent (hidden) variables or sources. So in the last time we get a lot of extensions of ICA such as Topographic ICA (2001)[36], Kernel ICA (2002)[7], Tree-Dependent Component Analysis (2003)[8], Non-negative Matrix Factorization (1999) [54], Multichannel Blind Deconvolution (2004) [72].

Chapter 2

Algorithms on Independent Component Analysis

In the previous chapter, we have shown the statistical properties of the ICA method. In this chapter, we describe the principal objective function and optimization algorithm for the ICA problem.

2.1 Introduction

The estimation of the data model of independent component analysis is usually performed by formulating an objective function and then minimizing or maximizing it. Often such a function is called a contrast function, but some authors reserve this term for a certain class of objective functions [26]. Also the terms loss function or cost function are used. We shall here use the term contrast function rather loosely, meaning any function whose optimization enables the estimation of the independent components. Although many different source separation algorithms are available, their principles can be summarized by the following four fundamental approaches:

- the most popular approach exploits as the cost function some measure of signals statistical independence, non-Gaussianity or sparseness. When original sources are assumed to be statistically independent without a temporal structure, the higher - order statistics (HOS) are essential (implicitly or explicitly) to solve the BSS problem. In such a case, the method does not allow more than one Gaussian

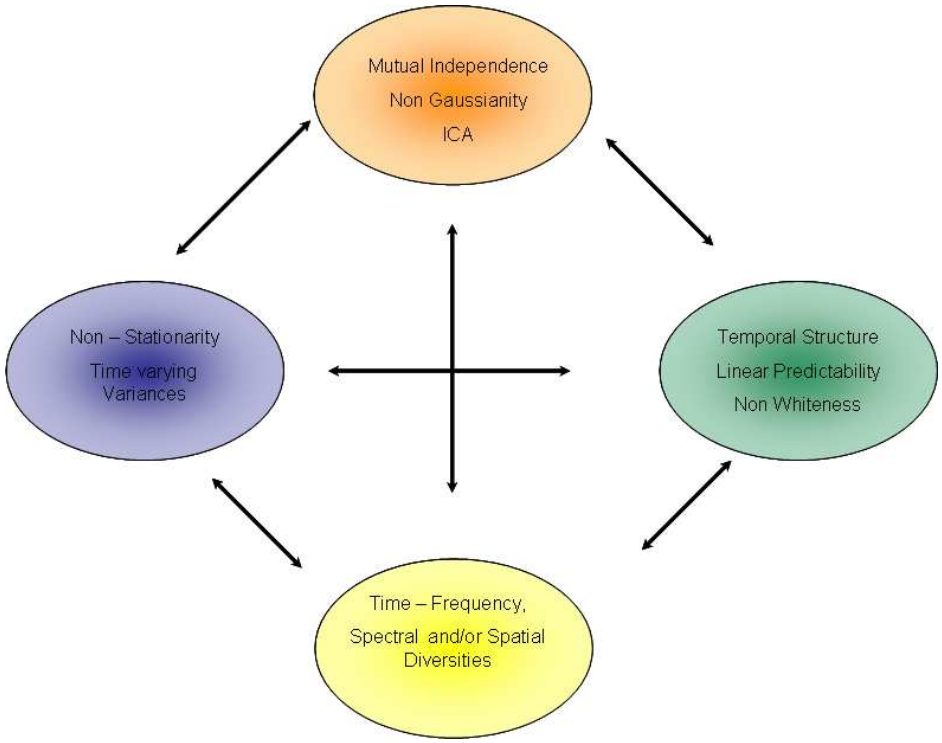


Figure 2.1: The principle of the source separation algorithms: four approaches.

sources;

- if sources have temporal structures, then each source has non-vanishing temporal correlation and less restrictive conditions than statistical independence can be used, namely, second - order statistics (SOS) are often sufficient to estimate the mixing matrix and sources. Note that the SOS methods do not allow the separation of sources with identical power spectra shapes or independent and identically distributed (i.i.d.) sources;

- the third approach exploits non - stationarity (NS) properties and second order statistics (SOS). Mainly, we are interested in the second order non - stationarity in the sense that source variances vary in time. The non - stationarity was first taken into account by [56]. However, these methods do not allow the separation of sources with identical non - stationarity properties;

- the fourth approach exploits the various diversities (we mean different characteristics or features of the signals), typically, time, frequency and/or time - frequency diversities, or more generally, joint space-time-frequency (STF) diversity.

More sophisticated or advanced approaches use combinations or integration of some of the above mentioned approaches, in order to separate or extract sources with various statistical properties and to reduce the influence of noise and undesirable interferences.

2.2 Cost Functions and Optimization Algorithms

In this section, we want to focus our attention on the formulation of the ICA method. We need to have a distinction between the formulation of the objective function and the algorithm used to optimize it, this is because the choice of the objective function is determinant for the statistical properties (e.g., consistency, asymptotic variance, robustness) of the method, while the optimization algorithm gives a characterization of the algorithmic properties (e.g., convergence speed, memory requirements, numerical stability) of the method.
In the case of explicitly formulated objective functions, one can use any of the classical methods of optimization for optimizing the objective function, like (stochastic) gradient methods, Newton-like methods, etc. In some cases, however, the algorithm and the estimation principle may be difficult to separate.
The statistical and algorithmic properties are independent in the sense that different optimization methods can be used to optimize a single objective function and a single optimization method may be used to optimize different objective functions.
Another important property in the algorithms for ICA estimation is how many independent components we want to estimate. Depending on that, we have two kind of contrast function:

- multi - unit contrast functions, in which we estimate all the independent components, or the whole data model, at the same time. Using this contrasts functions, we get a symmetric orthogonalization, this mean that the vector of the demixing matrix are not estimated one by one, but they are estimated in parallel.

- one - unit contrast functions, in which we estimate an independent component at time. In principle, we could find more independent components by running the algorithm many times and using different initial points. This would not be a reliable method of estimating many independent components, but using the

property that the vector of the demixing matrix corresponding to different components are orthogonal in the whitened space, we can orthogonalize the vectors for avoiding the convergence to the same maxima. A simple way of orthogonalization is deflationary orthogonalization using the Gram - Schmidt method.

2.3 Multi - Unit Contrast Functions

In this section, we will describe the multi unit contrast functions, so we will treat the problem of estimating all the independent components at the same time.

2.3.1 Likelihood and Network Entropy

A very popular approach for estimating the ICA model is maximum likelihood (ML) estimation. ML estimation is a fundamental method of statistical estimation and we can give an interpretation of ML estimation in ICA as taking those parameter values as estimates that gives the highest probability for the observations. It is possible to formulate the likelihood in the noise - free ICA model 1.8, which was done in [63],and then estimate the model by a maximum likelihood method.
Assuming that $\mathbf{W} \approx \mathbf{A}^{-1}$ is the unmixing matrix then, we can write:

$$\mathbf{x} = \mathbf{As} \text{ and } \mathbf{y} = \mathbf{Wx}.$$

Following a basic property of linear transformed random vectors:

$$f_x(\mathbf{x}) = \left|det(\mathbf{A}^{-1})\right| f_s(\mathbf{s}) \tag{2.1}$$

Assuming that $f_y(\mathrm{y}) \approx f_s(\mathrm{s})$ and statistical independence between the estimated sources $\mathbf{u}$, we can write:

$$f_x(\mathbf{x}) = |det(\mathbf{W})|\, f_y(\mathbf{y}) = |det(\mathbf{W})| \prod_{i=1}^{N} f_i(y_i) \tag{2.2}$$

Let $\mathbf{W} = [\mathbf{w}_1,...,\mathbf{w}_N]^T$. Therefore we can write:

$$f_x(\mathbf{x}) = |det(\mathbf{W})| \prod_{i=1}^{N} f_i(\mathbf{w}_i^T \mathbf{x}) \tag{2.3}$$

Assume that we have T observations of $\mathbf{x}$. Then the likelihood can be obtained as the product of this density evaluated at the T points. This is denoted by L and considered as a function of $\mathbf{W}$:

$$L(\mathbf{W}) = \prod_{t=1}^{T} \prod_{i=1}^{N} f_i(\mathbf{w}_i^T \mathbf{x(t)}) \, |det(\mathbf{W})| \tag{2.4}$$

Very often for practice reason it is used the logarithm of the likelihood.The log-likelihood takes the form [63]:

$$L = \sum_{t=1}^{T} \sum_{i=1}^{m} log f_i(\mathbf{w}_i^T \mathbf{x}\,(t)) + T \; ln \, |det \mathbf{W}| \tag{2.5}$$

where the f_i are the density functions of the s_i (here assumed to be known) and the $\mathbf{x}\,(t)$, $t = 1, ..., T$ are the realizations of $\mathbf{x}$.

Another related contrast function was derived from a neural network viewpoint in [9]. This was based on maximizing the output entropy (or information flow) of a neural network with non-linear outputs. Assume that $\mathbf{x}$ is the input to the neural network whose outputs are of the form $g_i\left(\mathbf{w}_i^T \mathbf{x}\right)$, where the g_i are some non-linear scalar functions and the $\mathbf{s}_i$ are the weight vectors of the neurons. One then wants to maximize the entropy of the outputs:

$$L_2 = H\left(g_1\left(\mathbf{w}_1^T \mathbf{x}\right), ..., g_m\left(\mathbf{w}_m^T \mathbf{x}\right)\right) \tag{2.6}$$

If the g_i are well chosen, this framework also enables the estimation of the ICA model. Indeed, several authors [16, 62], proved the surprising result that the principle of network entropy maximization , or "infomax", is equivalent to maximum likelihood estimation. This equivalence requires that the non-linearities g_i used in the neural network are chosen as the cumulative distribution functions corresponding to the densities f_i, i.e., $g_i'(.) = f_i(.)$.

The advantage of the maximum likelihood approach is that under some regularity conditions, it is asymptotically efficient; this is a well-known result in estimation theory. However, there are also some drawbacks. First, this approach requires the knowledge of the probability densities of the independent component. A second drawback is that the maximum likelihood solution may be very sensitive to outliers, if the pdf's of the independent components have certain shapes ([33]), while robustness against outliers is an important property for an estimator.

2.3.2 Mutual Information and Kullback-Leibler Divergence

An important approach for ICA estimation, inspired by information theory, is minimization of mutual information. The motivation of this approach is that we want to have a general purpose measure of the dependence of the components of a random vector. Using such measure, we could define ICA as a linear decomposition that minimizes that dependence measure. Such an approach can be developed using mutual information, which is a well-motivated information theoretic measure of statistical dependence.

One of the main utilities of mutual information is that it serves as a unifying framework for many estimation principles, in particular ML estimation and maximization of nongaussianity.

Using the concept of differential entropy [38], it is possible to define the mutual information between m scalar random variables y_i, $i = 1, ..., m$, as follows:

$$I(y_1, y_2, ..., y_m) = \sum_i H(y_i) - H(y) \tag{2.7}$$

where H denotes differential entropy. The mutual information is a natural measure of the dependence between random variables. It is always non-negative and zero if and only if the variables are statistically independent. Thus the mutual information takes into account the whole dependence structure of the variables. Finding a transform that minimizes the mutual information between the components s_i is a very natural way of estimating the ICA model [26]. This approach gives at the same time a method of performing ICA according to the general definition 1.4.1. We can note that by properties of mutual information, we have for an invertible linear transformation $\mathbf{y} = \mathbf{W}\mathbf{x}$:

$$I(y_1, y_2, ..., y_m) = \sum_i H(y_i) - H(\mathbf{x}) - log\,|det\mathbf{W}| \tag{2.8}$$

The use of mutual information can also be motivated using the Kullback-Leibler divergence, defined for two probability densities f_1 and f_2 as

$$\delta(f_1, f_2) = \int f_1(\mathbf{y}) log \frac{f_1(\mathbf{y})}{f_2(\mathbf{y})} d\mathbf{y} \tag{2.9}$$

The Kullback-Leibler divergence can be considered as a kind of a distance between the two probability densities, though it is not a real distance measure because it is not symmetric. Now, if the y_i in equation 2.7 were independent, their joint probability density could be factorized as in the definition of independence in equation 1.4. Thus one might measure the independence of the y_i as the Kullback-Leibler divergence between the real density $f(\mathbf{y})$ and the factorized density $\tilde{f}(\mathbf{y}) = f_1(y_1)f_2(y_2)...f_m(y_m)$, where $f_i(.)$ are the marginal densities of the y_i. In fact, this quantity equals the mutual information of the y_i.

The connection to the Kullback-Leibler divergence also shows the close connection between minimizing mutual information and maximizing likelihood. In fact, the likelihood can be represented as a Kullback-Leibler distance between the observed density and the factorized density assumed in the model [17]. So both of these methods are minimizing the Kullback-Leibler divergence between the observed density and a factorized density; actually the two factorized densities are asymptotically equivalent, if the density is accurately estimated as part of the ML estimation method.

The problem with mutual information is that it is difficult to estimate, because to use the definition of entropy, one needs an estimate of the density. This problem has severely restricted the use of mutual information in ICA estimation. Some authors have used approximations of mutual information based on polynomial density expansion [26, 4], which lead to the use of higher-order cumulants. The polynomial density expansions are related to the Taylor expansion. They give an approximation of a probability density $f(.)$ of a scalar random variable y using its higher-order cumulants. For example, the first terms of the Edgeworth expansion give, for a scalar random variable y of zero mean and unit variance:

$$f(\xi) \approx \varphi(\xi)(1 + \kappa_3(y)h_3(\xi)/6 + \kappa_4 h_4(\xi)/24 + ...) \tag{2.10}$$

where φ is the density function of a standardized Gaussian random variable, the $\kappa_i(y)$ are the cumulants of the random variably y and $h_i(.)$ are certain polynomial functions (Hermite polynomials).

Using such expansions, one obtains for example the following approximation for mutual information

$$I(\mathbf{y}) \approx C + \frac{1}{48}\sum_{i=1}^{m}\left[4\kappa_3(y_i)^2 + \kappa_4(y_i)^2 + 7\kappa_4(y_i)^4 - 6\kappa_3(y_i)^2\kappa_4(y_i)\right] \tag{2.11}$$

where C is constant; the y_i are here constrained to be uncorrelated. A very similar approximation was derived in [4] and also earlier in the context of projection pursuit in [47].

Cumulant-based approximations such as the one in equation 2.11 simplify the use of mutual information considerably. The approximation is valid, however, only when $f(.)$ is not far from the Gaussian density function, and may produce poor results when this is not the case. More sophisticated approximations of mutual information can be constructed by using the approximations of differential entropy that were introduce in [35], based on the maximum entropy principle. In these approximations, the cumulants are replaced by more general measures of nongaussianity.

2.3.3 Non-linear Cross-Correlations

Assume two random variables y_1 and y_2 and two functions $f(y_1)$ and $g(y_2)$, where at least one is nonlinear. We can say that y_1 and y_2 are nonlinearly decorrelated, if

$$E\left\{f(y_1)g(y_2)\right\} = 0 \tag{2.12}$$

Non-linear decorrelation can be a criterion for statistical independence. The variables y_1 and y_2 are statistically independent if

$$E\left\{f(y_1)g(y_2)\right\} = E\left\{f(y_1)\right\}E\left\{g(y_2)\right\} = 0 \tag{2.13}$$

for every continuous function f and g that are zero outside a finite interval. We can also show that, in order to satisfy the independence criterion, the functions f and g should be *odd* and y_1 and y_2 must have symmetrical probability density functions. In this general framework, we need to address the following:

- how can we choose f and g to satisfy equation 2.13;
- how can we nonlinearly decorrelate the variable y_1 and y_2.

Two attempts to address these questions was developed by Jutten and Hérault [48] in their seminal paper, and by Cichocki and Unbehauen [24]. After that several authors have used the principle of canceling non-linear cross-correlations to obtain the independent components [48, 19, 24].

2.3.4 Higher-order Cumulant Tensors

A principle of ICA estimation that is less directly connected with the objective function framework, is the eigenmatrix decomposition of higher-order cumulant tensors. Most solutions use the fourth-order cumulant tensor, whose properties and relation to the estimation of ICA have been studied extensively [14, 15, 18, 26].
The fourth-order cumulant tensor can be defined as the following linear operator T from the space of $m \times m$ matrices to itself:

$$T(\mathbf{K})_{ij} = \sum_{k,l} cum(x_i, x_j, x_k, x_l)\mathbf{K}_{kl} \tag{2.14}$$

where the subscript ij means the (i,j)-th element of a matrix and $\mathbf{K}$ is a $m \times m$ matrix. This is a linear operator and thus has m^2 eigenvalues that correspond to eigenmatrices. Solving for the eigenvectors of such eigenmatrices, the ICA model can be estimated [14].
The advantage of this approach is that it requires no knowledge of the probability densities of the independent components. Moreover, cumulants can be used to approximate mutual information [26, 4], as shown above, though the approximation is often very crude. The main drawback of this approach seems to be that the statistical properties of estimators based in cumulants are not very good.

2.4 One-Unit Contrast Functions

We use the expression one unit contrast function to designate any function whose optimization enables estimation of a single independent component. Thus, instead of estimating the whole ICA model, we try to find here simply one vector, say $\mathbf{w}$, so that the linear combination $\mathbf{w}^T\mathbf{x}$ equals one of the independent components s_i. This procedure can be iterated to find several independent components. The use of one-unit

contrast functions can be motivated by the following:

- the one-unit approach shows a direct connection to projection pursuit. Indeed, all the one-unit contrast functions discussed below can be considered as measure of non-Gaussianity and therefore this approach gives a unifying framework for these two techniques. The same contrast functions and algorithms can be interpreted in two different ways.

- In many applications, one does not need to estimate all the independent components. Finding only some of them is enough. In the ideal case where the one-unit contrast functions are optimized globally, the independent components are obtained in order of (descending) non-Gaussianity. In the light of the basic principles of projection pursuit, this means that the most interesting independent components are obtained first. This reduces the computational complexity of the method considerably, if the input data has a high dimension.

- Prior knowledge of the number of independent components is not needed, since the independent components can be estimated one-by-one.

- This approach also shows clearly the connection to neural networks. One can construct a neural network whose units learn so that every neuron optimizes its own contrast function. Thus the approach tends to lead to computationally simple solutions.

After estimating one independent component, one can use simple decorrelation to find a different independent component, since the independent components are by definition uncorrelated. Thus, maximizing the one-unit contrast function under the constraint of decorrelation (with respect to the independent components already found), a new independent component can be found, and this procedure can be iterated to find all the independent components. Symmetric (parallel) decorrelation can also be used [39, 52].

2.4.1 Negentropy

A most natural information-theoretic one-unit contrast function is negentropy. From equation 2.7, one is tempted to conclude that the independent components correspond

to directions in which the differential entropy of $\mathbf{w}^T\mathbf{x}$ is minimized. This turns out to be roughly the case. However, a modification has to be made, since differential entropy is not invariant for scale transformations. To obtain a linearly (and in fact affinely) invariant version of entropy, one defines the negentropy J as follows:

$$J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y}) \tag{2.15}$$

where $\mathbf{y}_{gauss}$ is a Gaussian random vector of the same covariance matrix as $\mathbf{y}$. Negentropy, or negative normalized entropy, is always non-negative, and is zero if and only if $\mathbf{y}$ has a Gaussian distribution [26].
The usefulness of this definition can be seen when mutual information is expressed using negentropy, giving

$$I(y_1, y_2, ..., y_n) = J(\mathbf{y}) - \sum_i J(y_i) + \frac{1}{2}\, log\, \frac{\prod \mathbf{C}^y_{ii}}{det\mathbf{C}^y} \tag{2.16}$$

where $\mathbf{C}^y$ is the covariance matrix of $\mathbf{y}$, and the $\mathbf{C}^y_{ii}$ are its diagonal elements. If the y_i are uncorrelated, the third term is 0, and we thus obtain

$$I(y_1, y_2, ..., y_n) = J(\mathbf{y}) - \sum_i J(y_i) \tag{2.17}$$

Because negentropy is invariant for linear transformations [26], it is now obvious that finding maximum negentropy directions, i.e., directions where the elements of the sum $J(y_i)$ are maximized, is equivalent to finding a representation in which mutual information is minimized. The use of negentropy shows clearly the connection between ICA and projection pursuit. Using differential entropy as a projection pursuit index, as has been suggested in [47], amounts to finding directions in which negentropy is maximized.
Unfortunately, the reservations made with respect to mutual information are also valid here. The estimation of negentropy is difficult, and therefore this contrast function remains mainly a theoretical one. As in the multi-unit case, negentropy can be approximated by higher-order cumulants, for example as follows [47]:

$$J(\mathbf{y}) \approx \frac{1}{12}\kappa_3(y)^2 + \frac{1}{48}\kappa_4(y)^2 \tag{2.18}$$

where $\kappa_i(y)$ is the i-th order cumulant of y. The random variable y is assumed to be of zero mean and unit variance. However, the validity of such approximations may be rather limited. In [35], it was argued that cumulant-based approximations of negentropy are inaccurate, and in many cases too sensitive to outliers. New approximations of negentropy were therefore introduced. In the simplest case, these new approximations are of the form:

$$J(\mathbf{y}) \approx c\left[E\{G(y)\} - E\{G(\upsilon)\}\right]^2 \tag{2.19}$$

where G is practically any non-quadratic function, c is an irrelevant constant and υ is a Gaussian variable of zero mean and unit variance (i.e., standardized). In [35], these approximations were shown to be better than cumulant-based ones in several respects. Actually, the two approximations of negentropy discussed above are interesting as one-unit contrast functions in their own right, as will be discussed next.

2.4.2 Higher-Order Cumulants

Mathematically the simplest one-unit contrast functions are provided by higher-order cumulants like kurtosis. Denote by $\mathbf{x}$ the observed data vector, assumed to follow the ICA data model 1.4.3.

Now, let us search for a linear combination of the observations x_i, say $\mathbf{w}^T\mathbf{x}$, such that its kurtosis is maximized or minimized. Obviously, this optimization problem is meaningful only if $\mathbf{w}$ is somehow bounded; let us assume $E\{(\mathbf{w}^T\mathbf{x})^2\} = 1$. Using the (unknown) mixing matrix $\mathbf{A}$, let us define $\mathbf{z} = \mathbf{A}^T\mathbf{w}$. Then, using the data model $\mathbf{x} = \mathbf{As}$ one obtains $E\{(\mathbf{w}^T\mathbf{x})^2\} = \mathbf{w}^T\mathbf{A}\mathbf{A}^T\mathbf{w} = \|\mathbf{z}\|^2 = 1$ (recall that $E\{\mathbf{ss}^T\} = \mathbf{I}$), and the well-known properties of kurtosis give

$$kurt(\mathbf{w}^T\mathbf{x}) = kurt(\mathbf{w}^T\mathbf{As}) = kurt(\mathbf{z}^T\mathbf{s}) = \sum_{i=1}^{m} z_i^4 kurt(s_i) \tag{2.20}$$

Under the constraint $\|\mathbf{z}\|^2 = 1$, the function in the equation 2.20 has a number of local minima and maxima. To make the argument clearer, let us assume for the moment that in the mixture in the equation 1.8, there is at least one independent component s_j whose kurtosis is negative, and at least one whose kurtosis is positive. Then, the extremal points in equation 2.20 are the canonical base vectors $\mathbf{z} = \pm\mathbf{e}_j$, i.e., vectors

whose all components are zero except one component which is ± 1. The corresponding weight vectors are $\mathbf{w} = \pm(\mathbf{A}^{-1})^T\mathbf{e}_j$, i.e., the rows of the inverse of the mixing matrix $\mathbf{A}$, up to a multiplicative sign. So by minimizing or maximizing the kurtosis in equation 2.20 under the given constraint, one obtains one of the independent components as $\mathbf{w}^T\mathbf{x} = \pm s_j$. These two optimization modes can also be combined into a single one, because the independent components correspond always to maxima of the *modulus* of the kurtosis.

Kurtosis has been widely used for one-unit ICA (see, for example, [41, 39]), as well as for projection pursuit [47]). The mathematical simplicity of the cumulants, and especially the possibility of proving global convergence results has contributed largely to the popularity of cumulant-based (one-unit) contrast functions in ICA, projection pursuit and related fields. However, it has been shown, for example in [33], that kurtosis often provides a rather poor objective function for the estimation of ICA, if the statistical properties of the resulting estimators are considered. Note that despite the fact that there is no noise in the ICA model in equation 1.8, neither the independent components nor the mixing matrix can be computed accurately because the independent components s_i are random variables, and, in practice, one only has a finite sample of $\mathbf{x}$. Therefore, the statistical properties of the estimators of $\mathbf{A}$ and the realizations of $\mathbf{s}$ can be analyzed just as the properties of any estimator. Such an analysis was conducted in [33] and the results show that in terms of robustness and asymptotic variance, the cumulant-based estimators tend to be far from optimal. Intuitively, there are two main reasons for this. Firstly, higher-order cumulants measure mainly the tails of a distribution, and are largely unaffected by structure in the middle of the distribution. Secondly, estimators of higher-order cumulants are highly sensitive to outliers [32]. Their value may depend on only a few observations in the tails of the distribution, which may be outliers.

2.4.3 General Contrast Functions

To avoid the problems encountered with the preceding objective functions, new one-unit contrast functions were developed in [38, 41]. Such contrast functions try to combine the positive properties of the preceding contrast functions, i.e. have statistically appealing properties (in contrast to cumulants), require no prior knowledge of

the densities of the independent components (in contrast to basic maximum likelihood estimation), allow a simple algorithmic implementation (in contrast to maximum likelihood approach with simultaneous estimation of the densities), and be simple to analyze (in contrast to non-linear cross-correlation approach).
The generalized contrast function (introduced in [39]), which can be considered generalizations of kurtosis, seem to fulfill these requirements. To begin with, note that one intuitive interpretation of contrast functions is that they are measure of non-normality. A family of such measures of non-normality could be constructed using practically any functions G and considering the difference of the expectation of G for the actual data and the expectation of G for Gaussian data. In other words, we can define a contrast function J that measures the non-normality of a zero-mean random variable y using any even, non-quadratic, sufficiently smooth function G as follows:

$$J_G(y) = |E_y \{G(y)\} - E_\upsilon \{G(\upsilon)\}|^p \tag{2.21}$$

where υ is a standardized Gaussian random variable, y is assumed to be normalized to unit variance, and the exponent p= 1,2 typically. The subscripts denote expectation with respect to y and υ.
Clearly, J_G can be considered a generalization of (the modulus of) kurtosis. For $G(y) = y^4$, J_G becomes simply the modulus of kurtosis of y. Note that G must not be quadratic, because then J_G would be trivially zero for all distributions. Thus, it seems plausible that J_G could be a contrast function in the same way as kurtosis. In fact, for p=2, J_G coincides with the approximation of negentropy given in equation 2.19. In [38], the finite sample statistical properties of the estimators based on optimizing such a general contrast function were analyzed. It was found that for a suitable choice of G, the statistical properties of the estimator (asymptotic variance and robustness) are considerably better than the properties of the cumulant based estimators. The following choice of G were proposed:

$$\begin{aligned} G_1(u) &= log(cosh(a_1 u)) \\ G_2(u) &= exp\left(-a_2 u^3/2\right) \end{aligned} \tag{2.22}$$

where a_1,$a_2 \geq 1$ are some suitable constants. In the lack of precise knowledge on the

distributions of the independent components or on the outliers, these two functions seem to approximate reasonably well the optimal contrast function in most cases. Experimentally, it was found that especially the value $1 \leq a_1 \leq 2, a_2 = 1$ for the contrast give good approximations. One reason for this is that G_1 above corresponds to the log-density of a super- Gaussian distribution and is therefore closely related to maximum likelihood estimation.

2.4.4 A Unifying View on Contrast Functions

It is possible to give a unifying view that encompasses most of the important contrast functions for ICA. First of all, we can see above, that the principles of mutual information and maximum likelihood are essentially equivalent [17]. Second, as already discussed above, the infomax principle is equivalent to maximum likelihood estimation [16, 62]. On the other hand, it was discussed above how some of the cumulant-based contrasts can be considered as approximations of mutual information. Thus it can be seen that most of the multi-unit contrast function are, if not strictly equivalent, at least very closely related. However, an important reservation is necessary here: for these equivalences to be at all valid, the densities f_i used in the likelihood must be a sufficiently good approximations of the true densities of the independent components. At the minimum, we must have one bit of information on each independent component: whether it is sub- or super-Gaussian [18, 16, 40]. This information must be either available a priori or estimated from the data, see [18, 16, 40]. This situation is quite different with most contrast functions based on cumulants, and the general contrast functions which estimate directly independent components of almost any non-Gaussian distribution.

Also for the one-unit contrast functions, we have a very similar situation. Negentropy can be approximated by cumulants or by the general contrast functions, which shows that the considered contrast functions are very closely related. In fact, looking at the formulas for likelihood and mutual information in equations 2.16 and 2.18, one sees that they can be considered as sums of one-unit contrast functions plus a penalizing term that prevents the vector $\mathbf{w}_i$ from converging to the same directions. This could be called a "soft" form of decorrelation. Thus we see that almost all the contrast functions could be described by the single intuitive principle: find the most non-Gaussian

projections and use some (soft) decorrelation to make sure that different independent component are found. So, the choice of contrast function is essentially reduced to the simple choice between estimating all the independent components in parallel or just estimating a few of them (possibly one-by-one). This corresponds approximately to the choosing between symmetric and hierarchical decorrelation, which is a choice familiar in PCA learning [38]. One must also make the less important choice between cumulant based and robust contrast functions (i.e. those based on non-quadratic function), but it seems that the robust contrast functions are to be preferred in most applications.

2.5 Algorithms for ICA

2.5.1 Introduction

After choosing one of the principles of estimation for ICA, one needs a practical method for its implementation. Usually, this means that after choosing an objective function for ICA, we need to decide how to optimize it. In this section, we shall discuss the optimization method. We must to recall that the statistical properties of the ICA method depend only on the objective function used.

2.5.2 Preprocessing of the Data

Some ICA algorithms require a preliminary sphering or whitening of the data $\mathbf{x}$ and even those algorithms that do not necessarily need sphering, often converge better with sphered data. Recall that the data has also been assumed to be centered (i.e. made zero-mean).

Sphering means that the observed variable $\mathbf{x}$ is linearly transformed to a variable $\mathbf{v}$:

$$\mathbf{v} = \mathbf{Q}\mathbf{x} \tag{2.23}$$

such that the covariance matrix of $\mathbf{v}$ equals unity: $\mathrm{E}\mathbf{v}\mathbf{v}^T = \mathbf{I}$. This transformation is always possible. Indeed, it can be accomplished by classical PCA [38]. In addition to sphering, PCA may allow us to determine the number of independent components (if $m > n$). If noise level is low, the energy of $\mathbf{x}$ is essentially concentrated on the subspace spanned by the n first principal components, with n the number of independent

components in the model. Several methods exist for estimating the number of signals (here, independent components) and thus this reduction of dimension partially justifies the assumption $m = n$.
In model 1.4.3, after sphering we have:

$$\mathbf{v} = \mathbf{Bs} \tag{2.24}$$

where $\mathbf{B}$=$\mathbf{QA}$ is an orthogonal matrix, because

$$E\left\{\mathbf{vv}^T\right\} = \mathbf{B}E\left\{\mathbf{ss}^T\right\}\mathbf{B}^T = \mathbf{BB}^T = \mathbf{I} \tag{2.25}$$

Recall that we have assumed that the independent components s_i have unit variance. We have thus reduced the problem of finding an arbitrary matrix $\mathbf{A}$ in model 1.4.3 to the simpler problem of finding an orthogonal matrix $\mathbf{B}$. Once $\mathbf{B}$ is found, equation 2.24 is used to solve the independent components from the observed $\mathbf{B}$ by

$$\mathbf{y} = \hat{\mathbf{s}} = \mathbf{B}^T\mathbf{v} \tag{2.26}$$

It is also worthwhile to reflect why sphering alone does not solve the separation problem. This is because sphering is only defined up to an additional rotation: if $\mathbf{Q}_1$ is a sphering matrix, then $\mathbf{Q}_2 = \mathbf{UQ}_1$ is also a sphering matrix if and only if $\mathbf{U}$ is an orthogonal matrix. Therefore, we have to find the correct sphering matrix that equally separates the independent components. This is done by first finding any sphering matrix $\mathbf{Q}$, and later determining the appropriate orthogonal transformation from a suitable non-quadratic criterion.

2.5.3 Jutten-Hérault Algorithm

The pioneering work in [48] was inspired by NNs. Their algorithm was based on canceling the non-linear cross-correlations. The non-diagonal terms of the matrix $\mathbf{W}$ are updated according to:

$$\Delta\ \mathbf{W}_{ij} \propto g_1\left(y_i\right) g_2\left(y_j\right) \quad \forall\ i \neq j \tag{2.27}$$

where g_1 and g_2 are some odd non-linear functions and the y_i are computed at every iteration as $\mathbf{y} = (\mathbf{y} + \mathbf{W})^{-1}\mathbf{x}$. The diagonal terms $\mathbf{W}_{ii}$ are set to zero. The y_i

then give after convergence, estimates of the independent components. Unfortunately, the algorithm converges only under rather severe restrictions [48].

2.5.4 Non-Linear Decorrelation Algorithms

Further algorithms for canceling non-linear cross-correlations were introduced independently in [21, 24] and [19]. Compared to the Jutten-Hérault algorithm, these algorithms reduce the computational overhead by avoiding any matrix inversion and improve its stability. For example, the following algorithm was given in [19, 24]:

$$\Delta\ \mathbf{W} \propto \left(\mathbf{I} - g_1\left(\mathbf{y}\right) g_2\left(\mathbf{y}^T\right)\right) \mathbf{W} \tag{2.28}$$

where $\mathbf{y} = \mathbf{W}\mathbf{x}$, the non-linearities $g_1(.)$ and $g_2(.)$ are applied separately on every components of the vector $\mathbf{y}$ and the identity matrix could be replaced by any positive definite diagonal matrix. In [19], the following EASI algorithm was introduced:

$$\Delta\ \mathbf{W} \propto \left(\mathbf{I} - \mathbf{y}\mathbf{y}^T - g\left(\mathbf{y}\right)\mathbf{y}^T - \mathbf{y}g\left(\mathbf{y}^T\right)\right) \mathbf{W} \tag{2.29}$$

A principal way to choosing the non-linearities used in this learning rules is provided by the maximum likelihood (or infomax).

2.5.5 Algorithms for Maximum Likelihood or Infomax Estimation

An important class of algorithms consists of those based on maximization of network entropy (infomax) [9], which is, under some conditions, equivalent to the maximum likelihood approach. Usually these algorithms are based on (stochastic) gradient ascent of the objective function. For example, the following algorithm was derived in [9]:

$$\Delta\ \mathbf{W} \propto \left[\mathbf{W}^T\right]^{-1} - 2\ tanh(\mathbf{W}\mathbf{x})\mathbf{x}^T \tag{2.30}$$

where the tanh function is applied separately on every component of the vector $\mathbf{W}\mathbf{x}$, as above. The tanh function is used here because it is the derivative of the log-density of the "logistic" distribution [9]. This function works for estimation of most super-Gaussian (sparse) independent components; for sub-Gaussian independent components, other functions must be used. The algorithm in equation 2.30 converges,

however, very slowly, as had been noted by several researchers. The convergence may be improved by whitening the data and especially by using the natural gradient. The natural (or relative) gradient method simplifies the gradient method considerably, and makes it better conditioned. The principle of the natural gradient [4, 3] is based on the geometrical structure of the parameter space and is related to the principle of the relative gradient [19] that uses the Lie group structure of the ICA problem. In the case of basic ICA, both of these principles amount to multiplying the right-hand side of equation 2.30 by $\mathbf{W}^T\mathbf{W}$. Thus we obtain:

$$\Delta\ \mathbf{W} \propto \left(I - 2\ tanh(\mathbf{y})\mathbf{y}^T\right)\mathbf{W} \tag{2.31}$$

with $\mathbf{y} = \mathbf{W}\mathbf{x}$. After this modification, the algorithm does not need sphering. Interestingly, this algorithm is a special case of the non-linear decorrelation algorithm in equation 2.27 and is closely related to the algorithm in equation 2.28. Finally, in [63], a Newton method for maximizing the likelihood was introduced. The Newton method converges in fewer iterations, but has the drawback that a matrix inversion (at least approximate) is needed in every iteration.

2.5.6 Neural One-Unit Learning Rules

Using the principle of stochastic gradient descent, one can derive simple algorithms from the one-unit contrast functions explained above. Let us consider first whitened data. For example, taking the istantaneous gradient of the generalized contrast function in equation 2.19 with respect to $\mathbf{w}$, and taking the normalization $\|w\|^2 = 1$ into account, one obtains the following Hebbin-like learning rule:

$$\Delta\mathbf{w}_i \propto r\ g\left(\mathbf{w}^T\mathbf{x}\right);\mathbf{w} = \frac{\mathbf{w}}{\|\mathbf{w}\|} \tag{2.32}$$

where the constant may be defined, e.g. as r=EG($\mathbf{wx}$) - EG(v). The non-linearity g can thus be almost any non-linear function; the important point is to estimate the multiplicative constant r in a suitable manner [38].

2.5.7 The Tensor-Based Algorithms

A large amount of research has been done on algorithms utilizing the fourth-order cumulant tensor for estimation of ICA [14, 15]. These are typically batch algorithms

(non-adaptive), using such tensorial techniques as eigenmatrix decomposition, which is a generalization of eigenvalue decomposition for higher-order tensors. Such a decomposition can be performed using ordinary algorithms for eigenvalue decomposition of matrices, but this requires matrices of size $m^2 \times m^2$. Since such matrices is often too large, specialized Lanczos type algorithms of lower complexity have also been developed [14]. These algorithms often perform very efficiently on small dimensions. However, in large dimensions, the memory requirements may be prohibitive, because often the coefficients of the fourth-order tensor must be stored in memory, which requires $O(m^4)$ units of memory. The algorithms also tend to be quite complicated to program, requiring sophisticated matrix manipulations.

2.5.8 The FastICA Algorithm

The FastICA learning rule finds a direction, i.e. a unit vector $\mathbf{w}$ such that the projection $\mathbf{w}^T\mathbf{x}$ maximizes independence of the single estimated source y. Independence is here measured by the approximation of the negentropy given by:

$$J_G(\mathbf{w}) = E\left\{G\left(\mathbf{w}^T\mathbf{x}\right)\right\} - E\left\{G(\upsilon)\right\}]^2 \tag{2.33}$$

where $\mathbf{w}$ is an m-dimensional (weight) vector, $\mathbf{x}$ represents our mixture of signals and υ is a standardized Gaussian random variable. Maximizing J_G allows to find *one* independent component or projection pursuit direction. Maximizing the sum of n one-unit contrast functions and taking into account the constraint of decorrelation, we obtain the following optimization problem:

$$\begin{aligned} &maximize \sum_{i=1}^{n} J_G(\mathbf{w}_i) \\ &under\ constraint\ E\left\{\left(\mathbf{w}_k^T\mathbf{x}\right)\left(\mathbf{w}_j^T\mathbf{x}\right)\right\} = \delta_{jk} \quad \{k,j\} = 1,...,n \end{aligned} \tag{2.34}$$

where, on the maximum, every vector $\mathbf{w}_i$ gives one of the rows of the separating matrix. In the projection pursuit interpretation, this equation gives n projection pursuit directions that are constrained to be decorrelated. Basically, we have the following choices for the contrast function [38]:

$$G_1(u) = \frac{1}{a_1} logcosh(a_1 u) \qquad g_1(u) = tanh(a_1 u) \tag{2.35}$$

$$G_2(u) = -\frac{1}{a_2} exp\left(-a_2 u^2/2\right) \quad g_2(u) = u exp\left(-a_2 u^2/2\right) \tag{2.36}$$

$$G_3(u) = \frac{1}{4} u^4 \qquad g_3(u) = u^3 \tag{2.37}$$

where u is a generic variable, $a_1 \geq 1$, $a_2 \tilde{-} 1$ are constants and g_i is the derivative of G_i. The benefits of the different contrast functions may be summarized as follow [38]:

- G_1 is a good general purpose contrast function.
- When the independent components are highly super-Gaussian, or when robustness is very important, G_2 may be the right choice.
- If computational overhead must be reduced, then piece-wise linear approximations of G_1 and G_2 may be used.
- The use of G_3, i.e. the kurtosis, is justified on statistical grounds only for estimating sub-Gaussian independent components when there are no outliers.
- In the special case where it is important to first find the super-Gaussian components, kurtosis can be used.

Moreover, we note that multi-modality is revealed by a low kurtosis. There is an interesting relationship between this and the objective function G_1: expanding G_1 in Taylor series, setting a_1=1 and $\mathbf{u} = \mathbf{w}_T\mathbf{x}$, we obtain for

$$E\{ln\, cosh\,(\mathbf{u})\} = \frac{1}{2} E\left\{\left(\mathbf{w}^T\mathbf{x}\right)^2\right\} - \frac{1}{12} E\left\{\left(\mathbf{w}^T\mathbf{x}\right)^4\right\} + \\ + \frac{1}{45} E\left\{\left(\mathbf{w}^T\mathbf{x}\right)^6\right\} + E\left\{O\left[\left(\mathbf{w}^T\mathbf{x}\right)^8\right]\right\} \tag{2.38}$$

Applying the whitening to the data, we have in the formula that the second term is dominating and kurtosis is minimized at least approximately [58].
We remark that the algorithm requires a preliminary whitening of the data: the observed variable $\mathbf{x}$ is linearly transformed to a zero-mean variable $\mathbf{v} = \mathbf{Q}\mathbf{x}$ such that E $\mathbf{v}\mathbf{v}^T = \mathbf{I}$. Whitening can always be accomplished by e.g. Principal Component Analysis [38].

The one-unit *fixed-point* algorithm for finding a row vector $\mathbf{w}$ is [38]:

$$\mathbf{w}^* = E\left[\mathbf{v}g\left(\mathbf{w}_i^T\mathbf{v}\right)\right] - E\left[\mathbf{v}g'\left(\mathbf{w}_i^T\mathbf{v}\right)\right]\mathbf{w}_i$$
$$\mathbf{w}_i = \mathbf{w}_i^* / \left\|\mathbf{w}_i^*\right\| \tag{2.39}$$

where $g(.)$ is a suitable non-linearity, in our case $g(y) = tanh(y)$ and $g'(y)$ is its derivative with respect to y.
The algorithm of the previous equations estimates just one of the independent components. To estimate several independent components, we need to run one-unit FastICA algorithm using several units (e.g. neurons) with weight vectors $\mathbf{w}_1$,...,$\mathbf{w}_n$. To prevent different vectors from converging to the same maximum we must decorrelate the outputs $\mathbf{w}_1^T\mathbf{x}$,...,$\mathbf{w}_n^T\mathbf{x}$ after every iteration. In specific applications it may be desidered to use a symmetric decorrelation, in which vectors are not privileged over the others. This can be accomplished by the classical method involving matrix-square-roots.
If we assume that the data is whitened, we have that

$$\mathbf{W} = \mathbf{W}\left(\mathbf{W}^T\mathbf{W}\right)^{-1/2} \tag{2.40}$$

where $\mathbf{W}$ is the matrix of the vectors ($\mathbf{w}_1$,...,$\mathbf{w}_n$), and the inverse-square-root is obtained from the eigenvalue decomposition as $(\mathbf{W}^T\mathbf{W})^{-1/2}$=$\mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T$ where $\mathbf{E}$ is the eigenvector matrix and $\mathbf{D}$ is the diagonal eigenvalue one.

2.5.9 Properties of the Fixed-Point Algorithm

The fixed-point algorithm for (approximate) minimization of mutual information has a number of desirable properties [38]:

- The convergence is cubic (or at least quadratic), under the assumption of the ICA ata model. This is in contrast to gradient descendent methods, where the convergence is only linear. This means a very fast convergence, as has been confirmed by simulations and experiments on real data;

- Contrary to gradient-based algorithms, there are no step size parameters to choose. This means that the algorithm is easy to use;

- The algorithm finds directly independent components of any non-Gaussian distribution, which is in contrast to many algorithms, where some estimate of the probability distribution function has to be first available;
- The fixed-point algorithm inherits most of the advantages of neural algorithms: it is parallel, distributed, computationally simple and requires little memory space. Stochastic gradient methods seem to be preferable only if fast adaptivity in a changing environment is required;
- The statistical properties for a suitable choices of the contrast functions are superior to those of the kurtosis-based approach.

Chapter 3

Beyond Independent Component Analysis: Overcomplete Bases

In the previous chapter, we have shown the principal objective functions and optimization algorithms for the classical ICA problem. In this chapter, we focus our attention on the problem of ICA with less sensors than sources, with a particular attention to the case of a single mixture.

3.1 Introduction

The standard formulation of ICA requires at least as many sensors as sources. Lewicki and Sejnowski [55] have proposed a first generalization of ICA method for learning overcomplete representations from data that allows for more basis vectors than dimensions in the input. The goal of this method is illustrated in figure 3.1 [37]. In a two dimensional data space, the observation $\mathbf{x}$ in figure 3.1(a,b) were generated by a linear mixture of two independent random super-gaussian sources. In this space, figure 3.1 (a) shows orthogonal basis vectors (PCA) and figure 3.1 (b) shows independent basis vectors. If the two-dimensional observed data are generated by three sparse sources as shown in figure 3.1 (c,d) the complete ICA representation (c) cannot model the data adequately but the overcomplete ICA representation (d) finds three basis vectors that fit the underlying distribution of the data.

In this situation, the mixing system is not invertible: we cannot obtain the independent components by simply inverting the mixing matrix $\mathbf{A}$. Therefore, even if we knew

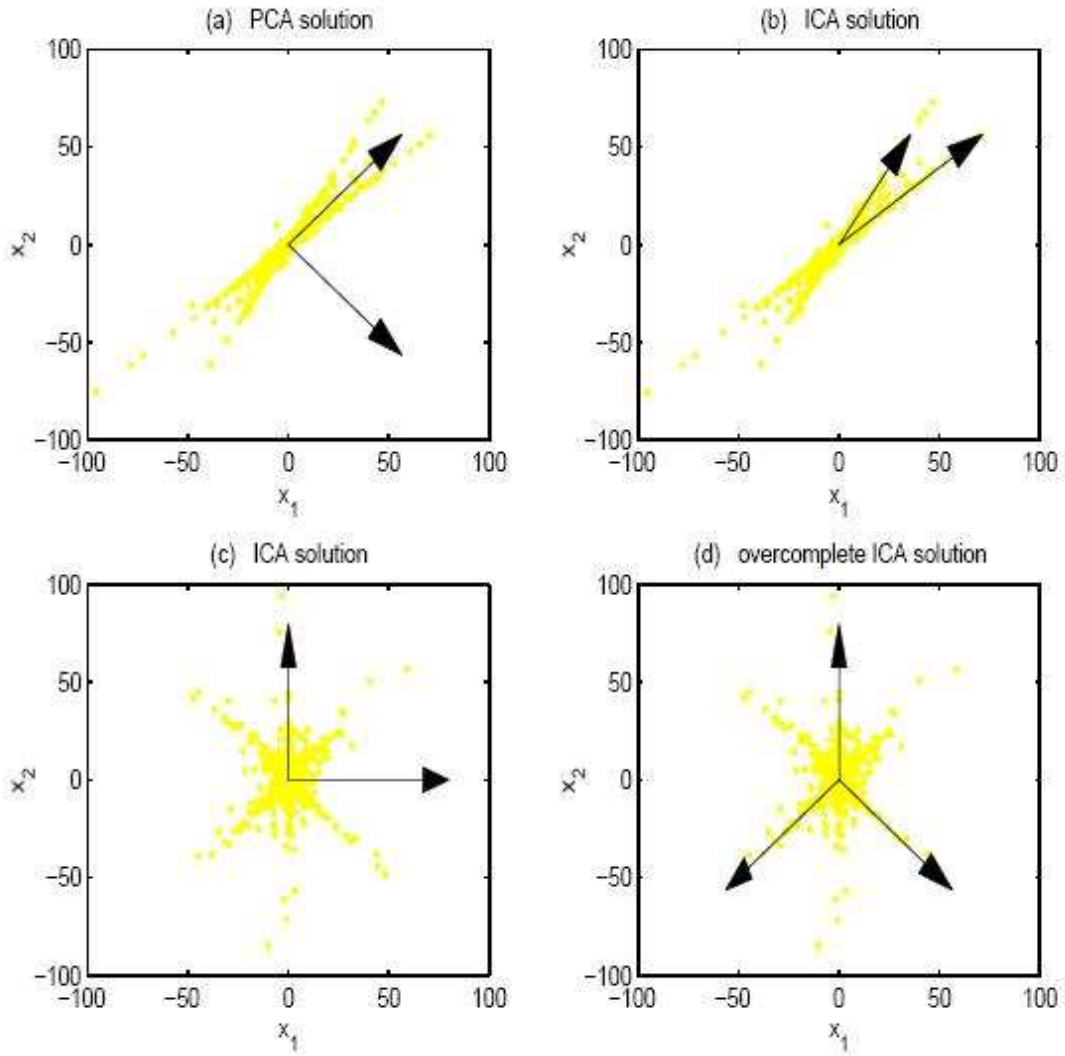


Figure 3.1: Illustration of basis vectors in a two-dimensional data space with two sparse sources (top) or three sparse sources (bottom).

the mixing matrix exactly, we could not recover the exact values of the independent components. This is because information is lost in the mixing process.
So we have two different problems. First, how to estimate the mixing matrix, and second, how to estimate the realizations of the independent components. This is in stark contrast to ordinary ICA, where these two problems are solved at the same time.

3.2 Is Source Separation Possible?

The two problems described below are called: the *identifiability* and the *separability* problems [38]. *Identifiability* describes the capability of estimating the structure of the linear model up to a scale and a permutation, while *separability* is the capability of retrieving the sources using the estimate of the mixing model. In the case of overcomplete ICA, it is still possible to identify the mixing matrix from the knowledge of $\mathbf{x}$ alone, although it is not possible to uniquely recover the sources $\mathbf{s}$. One of the possible solution to this problem is that of assuming a probability distribution for $\mathbf{s}$, one could obtain estimates of the sources by maximizing the likelihood of $p(\mathbf{x}|\mathbf{A},\mathbf{s})$. In the standard ICA formulation, we used the non-Gaussianity as a principle for the separation, in the overcomplete case non-Gaussianity is much more essential to facilitate the source separation task. For example, in the case of audio signals, we have certain time-domain statistical profile. Speech signals tend to have a Laplacian distribution, due to the many pauses that exist in the nature of speech. Musical signals tend to have a more Gaussian-like structure that might not affect the ICA algorithm in square case, but can affect the identifiability of the problem in the overcomplete case. A possible solution for signals with such statistics for overcomplete ICA is to use a linear, sparse, super-Gaussian, orthogonal transformation. A sparse transformation linearly maps the signal to a domain where most of the values are very small, i.e. concentrates the energy of the signals to certain areas. As a result the mixing matrix $\mathbf{A}$ remains unchanged by the signal transformation, so its estimation in the transform domain is equivalent to the estimation in the time-domain, although with sparser statistics. If the transform is invertible, one can perform the estimation of $\mathbf{y}$ in the transform domain. There are many candidate transform for this task, for example: the Fourier transform, the Discrete Cosine transform and the Wavelet Transform.

3.3 Estimating the source given the mixing matrix

This is a problem that does not exist in the standard formulation of ICA where $m = n$, and so you can invert the matrix $\mathbf{A}$ and get accurate estimate of your sources. In the $m \geq n$ case, the *pseudoinverse* can give accurate estimates of the sources. However, in the overcomplete case, the estimates obtained from the *pseudoinverse* are not accurate. Therefore, we have to resort to other methods to solve the problem.

3.3.1 Maximum Likelihood Estimation

One solution is to use Maximum Likelihood (ML) or Maximum A Posteriori (MAP) estimation to retrieve our sources, given the mixing matrix $\mathbf{A}$.
Imposing a source model, our sources can be retrieved by:

$$\mathbf{y} = argmax_{\mathbf{y}}\ P(\mathbf{y}|\mathbf{x},\mathbf{A}) = argmax_{\mathbf{y}}\ p_y(\mathbf{y})P(\mathbf{x}|\mathbf{A},\mathbf{y})P(\mathbf{y}) \tag{3.1}$$

Therefore, in the noiseless case the sources can be retrieved by

$$\Delta\mathbf{y} \propto -\delta\, logP(\mathbf{y})/\delta y \tag{3.2}$$

However, this gradient based algorithm is not very fast.

3.3.2 Linear Programming

Usually we employ sparse linear transform to enhance the quality of separation. Therefore, a Laplacian model for the sources $p(y) \propto\ exp^{-|y|}$ can be applied. A good starting point for the algorithm can always be the pseudoinverse solution. Lewicki [55] proved that source estimation assuming Laplacian priors, can be reduced to minimizing the *L1*-norm of the estimated sources.

$$min_{\mathbf{y}}\ \|\mathbf{y}\|_1 = min_{y_i} \sum_i |y_i| = min_{\mathbf{y}}[1\ 1...1]\,|\mathbf{y}| \tag{3.3}$$
$$subject\ to\ \mathbf{x} = \mathbf{W}\mathbf{y}$$

This can be transformed and solved as a linear programming problem. However, solving a linear programming problem for every time sample can be quite computationally

demanding and very slow. This can be quite important when you are updating the mixing matrix as well and you want to find an estimate of the sources for each estimate of **A**. In that case, we aim for a solution that can be fast and accurate.

3.4 Estimating the mixing matrix given the sources

3.4.1 Clustering Approach

Hyvärinen's Approach

Hyvärinen [34] in his analysis shows that maximizing the $\log p(\mathbf{A},\mathbf{s})$ is not an approximation but it is equivalent to the log-likelihood that Lewicki tries to maximize in [34]. Moreover, Hyvärinen forms a very efficient *clustering algorithm* for superGaussian components. In order to perform separation, he assumes that the sources are very sparse. Therefore, for sparse data you can claim that at most only one component is active at each sample. In other words, we attribute each point of the scatter plot to one source only. This is a *competitive winner-take-all* mechanism.
The step of the method are:

1. Initialize $\mathbf{A} = [\mathbf{a}_1,\mathbf{a}_2,...,\mathbf{a}_n]$.

2. Collect the points that are close to the directions represented by $\mathbf{a}_i$.
   For all $\mathbf{a}_i$ find the set of points S_i of $\mathbf{x}$ that:

$$\left|\mathbf{a}_i^T\mathbf{x}(n)\right| \geq \left|\mathbf{a}_j^T\mathbf{x}(n)\right|, \quad \forall\ j \neq i \tag{3.4}$$

3. Update

$$\mathbf{a}_i \leftarrow \sum_{n\in S_i} \mathbf{x}(n)(\mathbf{a}_i^T\mathbf{x}(n)) \tag{3.5}$$

$$\mathbf{a}_i \leftarrow \mathbf{a}_i / \left\|\mathbf{a}_i\right\|, \quad \forall\ i = 1,2,...,n \tag{3.6}$$

4. Repeat 2,3 until convergence.

As we can see, this is a clustering approach, as we force the direction of the mixing matrix to align along the concentration of the points in the scatter plot.

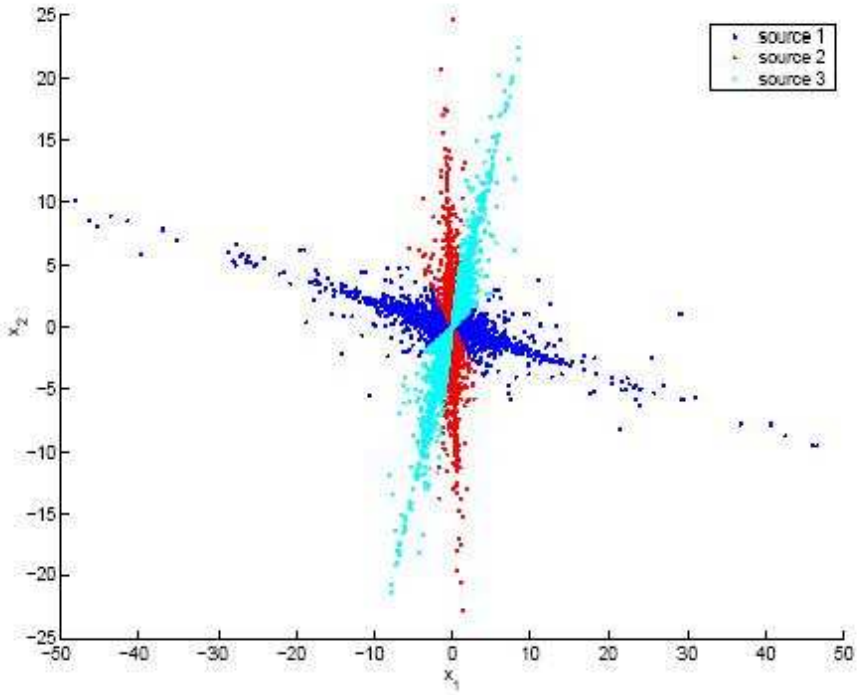


Figure 3.2: Illustration of clustering algorithm applied on 2 sensors - 3 sources scenario.

In figure 3.2, we show an example of clustering algorithm applied on 2 sensors - 3 sources scenario.

To estimate the sources in this case, all we have to do is construct the vectors $\mathbf{x}_{S_i}(\mathrm{t})$ that contain all the vectors from $\mathbf{x}(\mathrm{t})$ corresponding to each S_i. Then the estimates are given by:

$$\mathbf{y}_i = \mathbf{a}_i^T \mathbf{x}_{S_i} \tag{3.7}$$

3.4.2 Bayesian Approaches

Maximizing joint likelihood

In [55], Lewicki described a Bayesian approach to overcomplete ICA. He also explored the general case with additive noise n as described in equation 1.7.
Assuming that the noise is Gaussian and isotropic with covariance matrix $\mathbf{C}_n = \sigma\ {}_n^2\mathbf{I}$, it is possible to write:

$$log\ p(\mathbf{x}|\mathbf{A},\mathbf{s}) \propto -\frac{1}{2\sigma_n^2}(\mathbf{x} - \mathbf{A}\mathbf{s})^2 \tag{3.8}$$

Now, we have to deal with two problems as stated before:

- estimate $\mathbf{A}$;
- estimate $\mathbf{y}$.

We have discussed so far various methods for getting an estimate of the sources, given an estimate of $\mathbf{A}$. Now, Lewicki thought of maximizing the following:

$$max_{\mathbf{A}}\, p(\mathbf{x}|\mathbf{A}) = max_{\mathbf{A}} \int p(\mathbf{y})p(\mathbf{x}|\mathbf{A},\mathbf{y})d\mathbf{y} \tag{3.9}$$

After approximating $p(\mathbf{x}|\mathbf{A})$, with a Gaussian around $\mathbf{y}$ and a mathematical analysis, Lewicki derives a gradient algorithm that resembles the natural gradient.

$$\Delta\mathbf{A} \propto -\mathbf{A}(\phi(\mathbf{y})\mathbf{y}^T + \mathbf{I}) \tag{3.10}$$

where $\phi(\mathbf{y})$ represents the activation function. Assuming sparse priors, Lewicki proposed $\phi(\mathbf{y}) = \tanh(\mathbf{y})$. Lewicki claims that this approach can work for sources captured in the time-domain, however it is bound to have performance in a sparser domain. The algorithm can be summarised as follows:

1. randomly initialize A;
2. initialize source estimates $\mathbf{y}$ either with the pseudoinverse or with zero signals;
3. given the estimated $\mathbf{y}$, get a new estimate for $\mathbf{A}$:

$$\mathbf{A} \leftarrow \mathbf{A} - \eta\mathbf{A}(\phi(\mathbf{y})\mathbf{y}^T + \mathbf{I}) \tag{3.11}$$

   where η is the learning rate;
4. given the new estimate for $\mathbf{A}$, find a new estimate for $\mathbf{y}$ either by solving the linear programming problem for every sample n, or by other methods;
5. repeat steps 3,4 until convergence.

As this is a gradient algorithm, its convergence depends highly on the choice of learning rate and on signal scaling.

Mixtures of Gaussians - Attias' approach

Attias [6] proposed to model the sources as a *Mixture of Gaussians* (MoG) and used an *Expectation-Maximization* (EM) algorithm to estimate the parameters of the model. A MoG is defined as:

$$p(s_i) = \sum_{k=1}^{K} \pi_{ik} N_{s_i}(\mu_{ik}, \sigma_{ik}^2) \tag{3.12}$$

where K defines the number of Gaussians used, μ_{ik} and σ_{ik} denote the mean and standard deviation of the k^{th} Gaussian and $\pi_{ik} \in [0,1]$ the weight of each Gaussian, with the constraint that $\sum_{k=1}^{K} \pi_{ik} = 1$. To model the joint density function $p(\mathbf{s})$, we issue a vector $\mathbf{q}(t) = [q_1(t), q_2(t), ..., q_n(t)]$. Each $q_k(t)$ can take a discrete value from 1 to K and represents the state of the mixture of the k^{th} source at time t. the joint density function $p(\mathbf{s})$ is itself a MoG in the following form:

$$p(\mathbf{s}) = \prod_{i=1}^{N} p(s_i) = \sum_{q_1} ... \sum_{q_N} \pi_{1,q_1} ... \pi_{L,q_N} \prod_{i=1}^{N} N_{s_i}(\mu_{i,q_i}, \sigma_{i,q_i}^2) \tag{3.13}$$

Assuming additive Gaussian noise of zero mean and covariance J, it is possible to exploit the Gaussian structure to express $p(\mathbf{x}|\mathbf{A})$.
Attias [6] shows that:

$$p(\mathbf{x}|\mathbf{A},\mathbf{J}) = \sum_{q_1=1}^{K} ... \sum_{q_N=1}^{K} \pi_{1,q_1} ... \pi_{N,q_N} \times N_x(\mathbf{a}_1\mu_{1,q_1} + ... + \mathbf{a}_1\mu_{N,q_N}, J + \mathbf{a}_1\mathbf{a}_1^T\sigma_{1,q_1}^2 + ... + \mathbf{a}_N\mathbf{a}_N^T\sigma_{1,q_N}^N) \tag{3.14}$$

where $\mathbf{A} = [a_1, a_2, ..., a_n]$. In order to estimate the parameters of this model, Attias chose to minimize the Kullback-Leibler distance between the model sensor density $p(\mathbf{x}|\mathbf{A},\mathbf{J})$ and the observed one $p_o(\mathbf{x})$. He developed an EM algorithm to train the parameters of the model. Again, the whole training procedure is divided into two steps that are repeated for each iteration:

1. adapt the parameters of the model;

2. estimate the sources.

More in detail, for the first step we have:

$$\mathbf{A} = E\left\{\mathbf{x}\mathbf{y}^T\right\}\left(E\left\{\mathbf{x}\mathbf{x}^T\right\}\right)^{-1} \tag{3.15}$$

$$\mathbf{J} = E\left\{\mathbf{x}\mathbf{x}^T\right\} - E\left\{\mathbf{x}\mathbf{y}^T\right\}\mathbf{A}^T \tag{3.16}$$

$$\mu_{i,q_i} = \frac{E\left\{p(q_i|y_i)y_i\right\}}{E\left\{p(q_i|y_i)\right\}} \tag{3.17}$$

$$\sigma^2_{i,q_i} = \frac{E\left\{p(q_i|y_i)y_i^2\right\}}{E\left\{p(q_i|y_i)\right\}} - \mu^2_{i,q_i} \tag{3.18}$$

$$\pi_{i,q_i} = E\left\{p(q_i|y_i)\right\} \tag{3.19}$$

$$p(q_i|y_i) = \frac{\pi_{i,q_i}p(y_i)}{\sum_{j=1}^{N}\pi_{j,q_j}p(y_j)} \tag{3.20}$$

While for the second step, Attias proposed a MAP-estimator, maximizing the source posterior $\mathrm{p}(\mathbf{y}|\mathbf{x})$. More specifically,

$$\mathbf{y} = argmax_{\mathbf{y}}\ log\, p(\mathbf{x}|\mathbf{y}) + \sum_{i=1}^{N} log\, p(\mathbf{y}_i) \Rightarrow \tag{3.21}$$

$$\Delta\mathbf{y} = \eta\mathbf{A}^T\mathbf{J}^{-1}(\mathbf{x} + \mathbf{A}\mathbf{y}) - \eta\phi(\mathbf{y}) \tag{3.22}$$

where η is the learning rate and $\phi(\mathbf{y}) = \partial \log \mathrm{p}(\mathbf{y})/\partial\, \mathbf{y}$, incorporating the source model. All the Bayesian approaches tend to give complete and more general solutions. However they tend to be very slow in convergence, compared to clustering approaches.

3.5 An harder case: separation of independent components from a single mixture

Until now in this chapter we have presented some standard algorithm used for the ICA problem in the case of overcomplete basis, in particular when we have more than one mixture. This last case is a challenge problem still open. In fact analyzing the performance of the cited algorithm, we can see that they fail to solve the problem in the case of a single mixture.

In literature, some works have been proposed for this case, but usually they use some *a priori* knowledge about the source like the approach of T.W.Lee [44, 45]. In the next section, we will present some of these algorithm.

3.5.1 A probabilistic approach to single channel blind signal separation

This technique, presented by T. W. Lee and G.-J. Jang [44, 45] for extracting individual sound sources from an additive mixture of different signals, has as a central idea to exploit the inherent time structure of sources by learning *a priori* sets of basis filters in time domain that encode the sources in a statistically efficient manner. Sets of basis functions are learned a priori from the training data set and these sets are used to separate the unknown test sound sources. The algorithm recovers the original auditory streams in a number of gradient-ascent adaptation steps maximizing the log-likelihood of the separated signals, calculated using the basis functions and the probability density function (pdf) of their coefficients - the output of the ICA basis filters. The object function not only makes use of the ICA basis functions as a strong prior for the source characteristics, but also their associated coefficient pdf's modeled by generalized Gaussian distributions [44, 45, 43]. The algorithm first involves the learning of the time-domain basis functions of the sound sources that we are interested in the seaparating from a given training database. This corresponds to the prior information necessary to successfully separate the signals. The authors assume a generative models in the observed single channel mixture as well as in the original sources. The model is depicted in figure 3.3 [45].

In order to formulate the problem, the authors assume that the observed signal y^t is an addition of P independent source signals

$$y^t = \lambda_1 x_1^t + \lambda_2 x_2^t + \ldots + \lambda_p x_p^t \tag{3.23}$$

where x_i^t is the $t-th$ sampled value of the $i-th$ source signal and λ_i is the gain of each source which is fixed over time. So from this model, it is possible to observe that at every $t \in [1,T]$ the observed instance is assumed to be a weighted sum of different sources. In their approach, the authors regard only the case of $P = 2$, that is the situation of two different signals mixed and observed in a single sensor. For each individual source signals, the authors adopt a decomposition based approach by expressing a fixed-length segment drawn from a time varying signal as a linear superposition of a number of elementary patterns, called basis functions, with scalar multiplies, as explained in figure 3.3 (**B**). Continuous samples of length N, with $N <<$

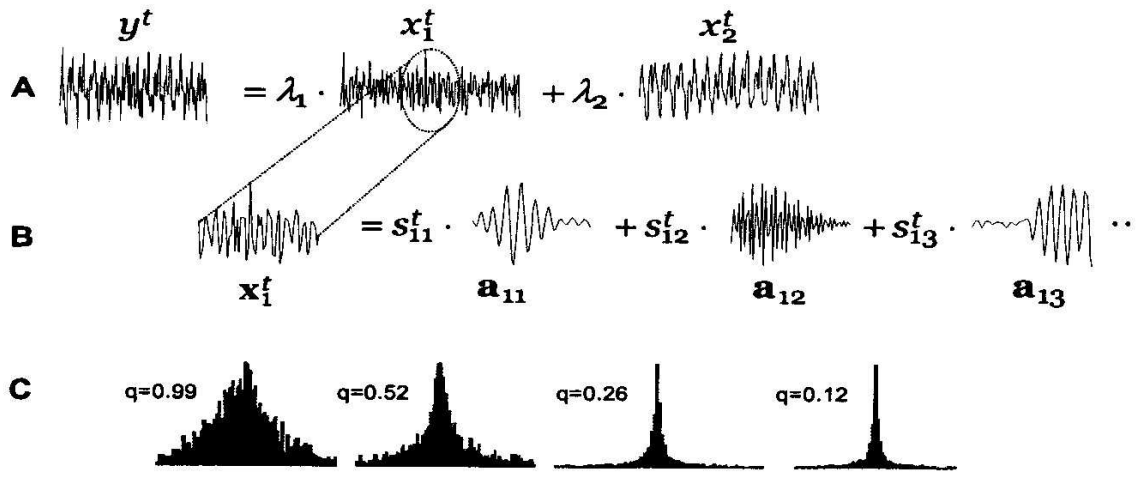


Figure 3.3: Generative models for the observed mixture and original source signal. From top (**A**): a single channel observation is generated by a weighted sum of two source signals with different characheristics. (**B**): individual source signals are generated by weighted (s_{ik}^t) linear superposition of basis functions (a_{ik}). (**C**): Examples of actual coefficient distributions.

T are chopped out of a source. The constructed column vector is then expressed as a linear combination of the basis functions such that

$$x_i^t = \sum_{k=1}^{M} \mathbf{a}_{ik} s_{ik}^t = \mathbf{A}\mathbf{s}_i^t \tag{3.24}$$

where M is the number of basis functions, $\mathbf{a}_{ik}$ is the $k-th$ basis function of the $i-th$ source in the form of N-dimensional column vector, s_{ik}^t its coefficient (weight). The authors assume that $M = N$ and $\mathbf{A}$ has full rank so that the transform between x_i^t and s_i^t be reversible in both directions. The inverse of the basis matrix, $\mathbf{W}_i = \mathbf{A}_i^{-1}$, refers to the ICA filters that generate the coefficient vector: $s_i^t = \mathbf{W}_i x_i^t$. The purpose of this decomposition is to model the multivariate distribution of x_i^t in a statistically efficient manner. The ICA learning algorithm is equivalent to searching for the linear transformation that make the components as statistically independent as possible, as well as maximizing the marginal densities of the transformed coordinates for the given training data [62],

$$\mathbf{W}_i^* = \ arg\,max_{\mathbf{W}_i} \prod_t Pr(x_i^t|\mathbf{W}_i) = \ arg\,max_{\mathbf{W}_i} \prod_t \prod_k Pr(s_{ik}^t) \tag{3.25}$$

where $Pr(a)$ denotes the probability of the value of a variable a. Independence between the components and over time samples factorizes the joint probabilities of the coefficients into the product of marginal ones. The authors use a generalized Gaussian prior [44, 45] to estimate these marginal probabilities. With the generalized Guassian ICA learning algorithm [45], the basis function and their individual parameters set are obtained beforehand and used as prior information for the source separation algorithm. This is essentially a *maximum a posteriori* (MAP) estimation in a number of adaptation steps on the source signals to maximize the data likelihood.
The major disvantage of this method is the necessity of a training data set, capable to modelize the basis function needed, if we don't have a training data set we can't use this method. So we can say that the *a priori* knowledge needed is impossible to recovery if we have a single observation of our mixture and no idea on the sources.

3.5.2 Different approaches

A really different approach to the separation of musical signal from single mixture is given by the Computational Auditory Stream Analysis (CASA) community. Any biological or artificial hearing system must extract individual acoustic objects or *streams* in order to do successfull localization, denoising and recognition. Bregman [12] called this process *auditory scene analysis*. Source separation or *computational auditory scene analysis* (CASA) is the practical realization of this problem via computer analysis of microphone recordings. The CASA community have focused on both multiple and single microphone source separation problems. Usually CASA approaches use almost exclusively hand designed systems which include substantial knowledge of the human auditory system and its psychophysical characteristics [64]. Recently, there was an approach that tried to bring together the representations of CASA and methods which learn from data such as ICA. In his paper [64], Roweis presents a technique called *refiltering* which recovers sources by a nonstationary reweighting ("masking") of frequency sub-bands from a single recording and argue for the application of statistical algorithms to learning this masking function. He uses a simple factorial HMM system which learns on recordings of single speakers and can then separate mixtures using only one observation signal by computing the masking function and then refiltering. As it is possible to note from this brief description, also in this case we need to learn some

"basis functions" or "filters" before to make the separation from a single mixture, so we need a training set.

3.6 Recent developments and conclusions

In this chapter, we have described some recent work about the case of independent component analysis in the case of overcomplete basis, focusing our attention to the case of a single mixture. The method described suffers of different disvantage: slow convergence, slow capabilities of approximations, too many a priori knowledge.
When passing to the case of a single mixture, we can say that at the moment there are very few algorithms that can work directly on a single mixture and in general this algorithm need to learn some a priori parameter from a bigger training data set. So they are unable to separate directly given only an observation mixtuere. This is still an open problem. A first temptative to solve this problem is given in [42], we will describe largely this approach in the next section.

Chapter 4

ICA on Single Mixture: a Projection Method

In the previous chapter, we described the stadard method developed for the overcomplete ICA scenario. In this chapter, we focus our attention on the problem of ICA on a single mixture, we propose a method of projection related to the dynamical system theory.

4.1 Introduction

In [42], the authors developed a methodology for the extraction of multisource brain activity using only single channel recordings of electromagnetic (EM) brain signals. At the hearth of the method is dynamical embedding (DE), where first an appropriate embedding matrix is constructed out of a series of delay vectors from the measured signal. The embedding matrix contains the information we require, but in a mixed form which therefore needs to be deconstructed. In particular, the authors demonstrated how one form of ICA performed on the embedding matrix can deconstruct the single channel recording into its underlying informative components.
In this chapter, we introduce the dynamical systems theory and the methodologies for constructing appropriate embedding matrix, starting from a single channel observations. We introduce, also, a slightly different methodology from [42] for the projection of the mixture.

4.2 Deterministic Chaos

The apparent contradiction (or paradox) contained in the term "deterministic chaos" has intrigued for long years also people not directly involved with science.
Deterministic math models are usually associated to the idea of regular, predictable fenomena, which repeat their behavior in time, while the term "chaotic" usually is referred to situation characterized by completely absence of rules and by unpredictability. The discovery of the deterministic chaos breaks this dichotomy, because it shows how deterministic math models (which are deprived of each element of randomness in their describing equations) can create extremely complex trends, which are unpredictable under many aspects, so to result almost indistinguishable from sequences of events, created by random processes.

4.3 Signals, Dynamical Systems and Chaos

Chaos comprises a class of signals intermediate between regular sinusoidal or quasiperiodic motions and unpredictable, truly stochastic behavior.
It has long been seen as a form of "noise", because the tools for its analysis were couched in a language tuned to linear processes [1].
In the analysis of signals from physical systems, usually it is impossible to assume that the system is linear, instead we assume from the outset that a dynamical system in the form of a differential equation or a discrete - time evolution rule is responsible for the observations.
Chaos occurs as a feature of orbits $\mathbf{x}$(t) arising from nonlinear evolution rules which are systems of differential equations

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{F}(\mathbf{x}(t)) \tag{4.1}$$

with three or more degrees of freedom $\mathbf{x}(t) = [x_1(t), x_2(t), ..., x_d(t)]$ or invertible discrete time maps[1]

$$\mathbf{x}(t+1) = \mathbf{F}(\mathbf{x}(t)) \tag{4.2}$$

[1]Non Invertible maps in one dimension can show chaos as in the example of the logistic map $x \rightarrow$ r $x(1\text{-}x)$.

with two or more degrees of freedom [1]. Degrees of freedom in systems characterized by ordinary differential equations means the number of required first order autonomous ordinary differential equations.
In discrete time systems, which are described by maps $x(t) \rightarrow \mathbf{F}(\mathbf{x}(t))=x(t+1)$, the number of degrees of freedom is the same as the number of the components in the state vector $\mathbf{x}(t)$. The requirement for a minimum size of state space to realize chaos is geometric. For differential equations in the plane (d=2) it has been known for a long time that only fixed points (time independent solutions) or limit cycles (periodic orbits) are possible. Chaos, as a property of the orbits $\mathbf{x}(t)$, manifest itself as complex time traces with continuous, broadband Fourier spectra, nonperiodic motion and exponential sensitivity to small changes in the orbit.
As a class of observable signals $\mathbf{x}(t)$, chaos lies logically between:

1. the well studied domain of predictable, regular, or quasi-periodic signals which have been the mainstay of signal processors for decades, and

2. the totally irregular **stochastic** signals we call "noise" and which are completely unpredictable.

With conventional **linear** tools such as Fourier transforms, chaos looks like "noise", but chaos has structure in an appropriate state or phase space.
That structure means there are numerous potential engineering applications of sources of chaotic time series which can take advantage of the structure to predict and control those sources.
One important insight into dynamical systems is the role played by *information theory*. There is an intuitive notion that a dynamical system that has chaotic behavior is precisely a realization of Shannon's concept of an ergodic information source [1].

4.4 Observed Chaos

From the point of view of extracting quantitative information from observations of chaotic systems, the characteristic feature just outlined in the previous section, pose an interesting challenge to the observer. First of all, it is typical to observe only one

or at best a few of the dynamical variables which govern the behavior of the system of interest.

How are we to go from scalar to univariate observations to the multivariate state or phase space which is required for chaotic motions to occur in the first place?

To address this we focus our attention on discrete time maps. This is really no restriction as in some sense all analysis of physical systems takes place in discrete time: we never sample anything continuously. If we sample a scalar signal s(t) at time intervals τ_s starting at some time t_0, then our data is actually of the form s(n) = s(t_0+n τ_s), and the evolution we observe takes us from s(k) to s(k+1).

We can represent continuous flows

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{F}(\mathbf{x}(t)) \tag{4.3}$$

as finitely sampled evolution

$$\mathbf{x}(t_0 + (n+1)\tau_s) \approx \mathbf{x}(t_0 + n\tau_s) + \tau_s \mathbf{F}(\mathbf{x}(t_0 + n\tau_s)) \tag{4.4}$$

So the observations take

$$\begin{aligned} s(t_0 + k\tau_s) \rightarrow s(t_0 + (k+1)\tau_s), \\ s(k) \rightarrow s(k+1) \end{aligned} \tag{4.5}$$

4.5 Reconstructing Phase Space or State Space

The answer to the question how to go from scalar observation s(k) = s(t_0+kτ_s) to multivariate phase space is contained in the geometric theorem called the embedding theorem attributed to Takens and Mañé [1].

Suppose we have a dynamical system x(t) $\rightarrow$ $\mathbf{F}$($\mathbf{x}$(t))=x(t+1), where $\mathbf{x}$(t) phase space is multidimensional. The theorem tells us that if we are able to observe a single scalar quantity h(.), of some vector function of the dynamical variables $\mathbf{g}$($\mathbf{x}$(n)), then the geometric structure of the multivariate dynamics can be **unfolded** from this set of scalar measurements h($\mathbf{g}$($\mathbf{x}$(n))) in a space made out of new vectors with components consisting of h(.) applied to powers of $\mathbf{g}$($\mathbf{x}$(n)). These vectors

$$\mathbf{y}(n) = [h(\mathbf{x}(n)), h(\mathbf{g}^{\tau_1}(\mathbf{x}(n))), h(\mathbf{g}^{\tau_2}(\mathbf{x}(n))), ..., h(\mathbf{g}^{\tau_{d-1}}(\mathbf{x}(n)))] \tag{4.6}$$

define motion in a d-dimensional Euclidian space.
With quite general conditions of smoothness on the functions $h(.)$ and $\mathbf{g}(\mathbf{x})$ [], it is shown that if d is large enough, then many important properties of the unknown multivariate signal $\mathbf{x}$(n) at the source of the observed chaos are reproduced without ambiguity in the new space of vectors $\mathbf{y}$(n).
In particular, it is shown that the sequential order of the points $\mathbf{y}$(n) $\rightarrow$ $\mathbf{y}$(n+1), namely, the evolution in time, follows that of the unknown dynamics $\mathbf{x}$(n) $\rightarrow$ $\mathbf{x}$(n+1), assures the deterministic behavior of the substitute representation of this dynamics $\mathbf{y}$(n) $\rightarrow$ $\mathbf{y}$(n+1). The integer dimension of the original space need not be the same as the integer dimension of the reconstructed space.
The vector $\mathbf{y}$(n) is designed to assure that errors in the sequential order which might occur during the projection from the evolution in the original $\mathbf{x}$(n) space down to the scalar space h($\mathbf{g}$($\mathbf{x}$(n))) are undone. Such errors result if two points quite far apart in the original space were projected near each other along the axis of scalar observations. This false neighborliness of observations in h($\mathbf{g}$($\mathbf{x}$(n))) can arise from projection from a higher dimensional space. It has nothing to do with closeness due to dynamics. Further, such an error would be mistaken for some kind of "random" behavior as the deterministic sequence of phase space locations along a true orbit would be interrupted by false neighbors resulting from the projection.
To implement the general theorem any smooth choice for $h(.)$ and $\mathbf{g}(\mathbf{x})$ is possible []. We focus our attention to a choice that is easy to utilize directly from observed data. One uses for the general scalar function $h(.)$ the observed scalar variable $\mathbf{s}$(n)

$$h(\mathbf{x}(n)) = s(n) \tag{4.7}$$

and for the general function $\mathbf{g}(\mathbf{x})$, we choose the operation which takes some initial vector $\mathbf{x}$ to that vector one time delay τ_s later so the τ_k^{th} power of $\mathbf{g}(\mathbf{x})$ is

$$g^{\tau_k}(\mathbf{x}(n)) = \mathbf{x}(n + \tau_k) = \mathbf{x}(t_0 + (n + \tau_k)\tau_s) \tag{4.8}$$

then the components of $\mathbf{y}$(n) take the form:

$$\mathbf{y}(n) = [s(n), s(n + \tau_1), s(n + \tau_2), ..., s(n + \tau_{d-1})] \tag{4.9}$$

If we make the further useful choice $\tau_k = k\tau$, that is, time lags which are integer multiples of a common lag τ, then the data vectors $\mathbf{y}$(n) are:

$$\mathbf{y}(n) = [s(n), s(n+\tau), s(n+2\tau), ..., s(n+(d-1)\tau)] \tag{4.10}$$

composed simply of time lags of the observation at time n $\times\ \tau_s$. These $\mathbf{y}$(n) replace the scalar data measurements s(n) with data vectors in an Euclidian d-dimensional space in which the invariant aspects of the sequence of points $\mathbf{x}$(n) are captured with no loss of information about the properties of the original system. The new space is related to the original space of the $\mathbf{x}$(n) by smooth, differentiable transformations.

The basic idea of this construction of a new state space is that if one has an orbit - a time ordered sequence of points in some multivariate space observed at time differences τ_s - seen projected onto a single axis $h(.)$ or s(n) on which the measurements happen to be made, then the orbit, which we presume came from an autonomous set of equations, may have overlaps with itself in the variables s(n) - by virtue of the projection, not from the dynamics. We know there is no overlap of the orbit with itself in the true set of state variables by the uniqueness theorems about the solutions of autonomous equations. Unfortunately, we don't know these true state variables, having observed only s(n). If we can unfold the orbit by providing independent coordinates for a multi-dimensional space made out of the observations, then we can undo the overlaps coming from the projection and recover orbits which are not ambiguous.

The reconstruction theorem recognizes that even in the case where the motion is along a one-dimensional curve, it is possible for the orbit to overlap in points when one uses two-dimensional space to view it. If one goes to a three-dimensional space [s(n),s(n+τ),s(n+2τ)], then any such remaining points of overlap are undone. The theorem notes that if the motion lies on a set of dimension d_A, which could be fractional, then choosing the integer dimension d of the unfolding space so $d{>}d_A$ is sufficient to undo all overlaps and make the orbit unambiguous.

It is important to note that once one has enough coordinates to unfold any overlaps due to projection, further coordinates are not needed: they serve no purpose in revealing the properties of the dynamics. The embedding theorem [70] works in principle for any value of τ once the dimension is large enough as long as one has an infinite amount of noise free data. This is never going to happen to anyone. This means some thought must be given as to how one may choose both the time delay τ and the embedding

dimension d when one is presented with real, finite length and possibly contaminated data. In the next sections, we will describe some methods for the choice of the time delay and the embedding dimension.

4.6 Choosing Time Delays

The statement of the embedding theorem [70] that any time lag will be accetable is not useful for extracting information from the data. If we choose τ too small, then the coordinates $x(n+j\tau)$ and $x(n+(j+1)\tau)$ will be so close to each other in numerical value that we cannot distinguish from each other. From any practical point of view, they have not provided us with two independent coordinates. Similarly, if τ is too large, then $x(n+j\tau)$ and $x(n+(j+1)\tau)$ are completely independent of each other in statistical sense and the projection of an orbit on the attractor is onto two totally unrelated directions. The origin of this statistical independence is the ubiquitous instability in chaotic systems, which results in any snall numerical or measurement error's being amplified exponentially in time. A criterion for an intermediate choice is called for, and it cannot come from the embedding theorem itself or considerations based on it, since the theorem works for almost any value of τ. Now, we introduce two possible methods for estimating τ.

4.6.1 Cross Correlation

One's first thought might be to consider the values of $x(n)$ as chosen from some unknown distribution. Then computing the *linear autocorrelation function* [1]:

$$C_L(\tau) = \frac{\frac{1}{N}\sum_{m=1}^{N}[x(m+\tau)-\bar{x}\,][x(m)-\bar{x}\,]}{\frac{1}{N}\sum_{m=1}^{N}[x(m)-\bar{x}\,]^2} \tag{4.11}$$

where

$$\bar{x} = \frac{1}{N}\sum_{m=1}^{N} x(m) \tag{4.12}$$

and looking for that time lag where $C_L(\tau)$ first passes through zero, would give us a good hint of a choice for τ.

Indeed, this does give a good hint. It tells us, however, about the independence of the coordinates only in a linear fashion. To see this, recall that if we want to know whether

two measurements $x(n)$ and $x(n+\tau)$ depend linearly on each other on the average over the observations, we find that their connection, in a least-squares sense, is through the correlation matrix just given.
That is, if we assume that the values of $x(n)$ and $x(n+\tau)$ are connected by

$$[x(n+\tau) - \bar{x}\,] = C_L(\tau)\,[x(n) - \bar{x}\,] \tag{4.13}$$

then minimizing

$$\sum_{n=1}^{N} \{x(n+\tau) - \bar{x} - C_L(\tau)\,[x(n) - \bar{x}\,]\}^2 \tag{4.14}$$

with respect to $C_L(\tau)$, immediately leads to the definition of $C_L(\tau)$ above.
Choosing τ to be the first zero of $C_L(\tau)$ would then, on average over the observations, make $x(n)$ and $x(n+\tau)$ linearly independent. What this may have to do with their nonlinear dependence or their utility as coordinates for a nonlinear system is not addressed by all this. Since we are looking for a *prescription* for choosing τ and this prescription must come from considerations beyond those in the embedding theorem, linear independence of coordinates may serve, but we prefer another point of view, one that stresses an important aspect of chaotic behavior - namely the viewpoint of information theory [27] - and leads to a nonlinear notion of independence.

4.6.2 Average Mutual Information

The second method that we introduce for choosing the time delay is based on the average mutual information [27]. The mutual information between measurement a_i drawn from a set A=$\{a_i\}$ and b_j drawn from a set B= $\{b_j\}$ is the amountlearned by the measurement of a_i about the measurement of b_j. In bits, it is

$$log_2 \left[\frac{P_{AB}(a_i, b_j)}{P_A(a_i)P_B(b_j)}\right] \tag{4.15}$$

where $P_{AB}(a,b)$ is the joint probability density for measurements A and B. $P_A(a)$ and $P_B(b)$ are the individual probability densities for the measurements of A and B. If the measurements of a value from A is completely independent from a measurement of a value from B, then $P_{AB}(a,b)$ factorizes: $P_{AB}(a,b) = P_A(a)P_B(b)$ and the amount of information between the measurements, the mutual information, is zero, as it should

be. The average over all measurements of this information statistic, called the average mutual information between A and B measurements, is:

$$I_{AB} = \sum_{a_i,b_j} P_{AB}(a_i, b_j)\, log_2 \left[\frac{P_{AB}(a_i, b_j)}{P_A(a_i) P_B(b_j)} \right] \tag{4.16}$$

To place this abstract definition in the context of observations from a physical system $x_(i)$, we think of the sets of measurements x_i as the A set and of the measurement a time lag τ later, $x_{i+\tau}$, as the B set. The average mutual information between observations at i and $i+\tau$, namely , the average amount of information about $x_{i+\tau}$ we have when we make an observation x_i, is then

$$I(\tau) = \sum_{i=1}^{N} P(x_i, x_{i+\tau})\, log_2 \left[\frac{P(x_i, x_{t+\tau})}{P(x_i) P(x_{i+\tau})} \right] \tag{4.17}$$

and I(τ) ≥ 0.

The average mutual information can be considered a kind of generalization to the nonlinear world from the correlation function in the linear world. It is the average over the data or equivalently the attractor of a special statistic, namely the mutual information, while the correlation function is the average over a quadratic polynomial statistic.

Now we have to decide what property of I(τ) we should select, in order to establish which among the various values of $I(\tau)$ we should use in making our data vectors y_i. If τ is too small, the measurements $x(n)$ and $x(n+\tau)$ tells us so much about one another that we need not make both measurements. If τ is large, then $I(\tau)$ will approach zero and nothing connects $x(n)$ and $x(n+\tau)$, so this is not useful.

Fraser and Swinney [27] suggest as a *prescription* that we choose that τ_m where the first minimum of I(τ) occurs as a useful selection of time lag τ. The lag τ_m is selected as a time lag where the measurements are somewhat independent, but not statistically independent.

Recognizing that this is a prescription, one may well ask what to suggest if the average mutual information ha no minimum. This occurs when one is dealing with maps, as the $I(\tau)$ curve from $x(n)$ data taken from the Hénon map [1].

This does not mean that $I(\tau)$ loses its role as a good grounds for selection of τ, but only that the first minimum criterion needs to be replaced by something representing good sense. Without much grounds beyond intuition, we use $\tau = 1$ or 2 if we know

the data comes from a map, or choose τ such that $I(\tau)/I(0) \approx \frac{1}{5}$. This is clearly an attempt to choose a useful τ in which some nonlinear decorrelation is at work, but not too much. Since this is prescriptive, one may compare it to the prescription used in linear dynamics of choosing a time lag τ such that $C_L(\tau) = 0$ for the first time.
Recognizing, as we have stressed, that the choice of τ is prescriptive, we agree with the caution that "we do not believe that there exists a unique optimal choice of time lag". Nonetheless, it is useful to have a general rule of thumb as a guide to a delay τ that is workable; seeking the optimum is likely to be quite unrewarding.

4.7 Choosing the Embedding Dimension

The goal of the reconstruction theorem [70] is to provide a Euclidean space R^d large enough so that the set of points of dimension d_A can be unfolded without ambiguity. This means that if two points of the set lie close to each other in some dimension d they should do so because it is a property of the set of points, not of the small value of d in which the set is being viewed.
The simplest example is that of a sine wave $s(t) = A\ sin(t)$. Seen in $d = 1$ (the $s(t)$ space), as in figure 4.1, this oscillates between $\pm A$. Two points on this line which are

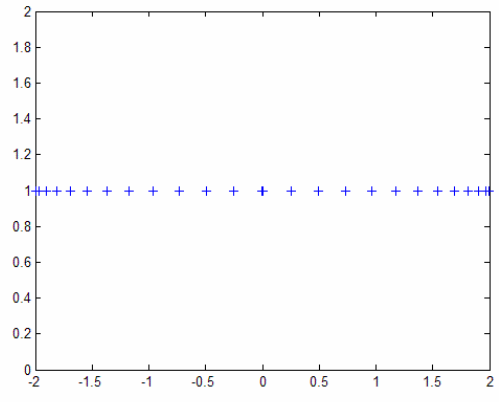


Figure 4.1: The phase space structure of a sine wave seen in one dimension $x(t)$ where $x(t) = 2sin(t)$.

close in the sense of Euclidean or other distance may have quite different values of $\dot{s}(t)$. So two "close" points in $d = 1$ may be moving in opposite directions along the single

spatial axis chosen for viewing the dynamics.
Seen in a two dimensional space $[s(t), s(t + T\tau_s)]$, as in figure 4.2, the ambiguity of velocity of the points is resolved and the sine wave is seen to be motion on a figure topologically equivalent to a circle. It is generically an ellipse whose shape depends on

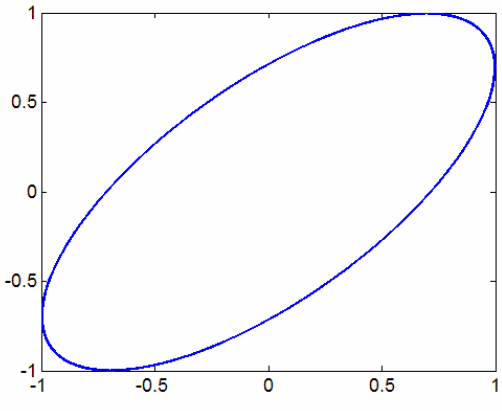


Figure 4.2: The phase space structure of a sine wave seen in two dimensions $[x(n), x(n+2)]$ where $x(t) = 1sin(t)$.

the value of T. The overlap of orbit points due to projection onto the one - dimensional axis is undone by the creation of the two - dimensional space.
If we procced further and look at the sine wave in three dimensions, as in figure 4.3, no further unfolding occurs and we see the sine wave as another ellipse.
It is clear that once we have unfolded without ambiguity the geometric figure on which the orbit moves, no further unfolding will occur. When all ambiguities are resolved, one says that the space R^d provides an embedding of the attractor.
An equivalent way to look at the embedding theorem is to think of the attractor as comprised of orbits from a system of very high dimension. The attractor, which has finite d_A, lies in a very small part of the whole phase space and we can hope to provide a projection of the whole space down to a subspace in which the attractor can be faithfully captured. The embedding theorem provides a *sufficient* condition from geometrical considerations alone for choosing a dimension d_E large enough so that the projection is good - i.e. without orbit crossings of dimension zero, one, two, etc.
If we work with a dimension d_E larger than necessary, two problems will arise:

1. many of the computations, needed for extracting interesting properties from the

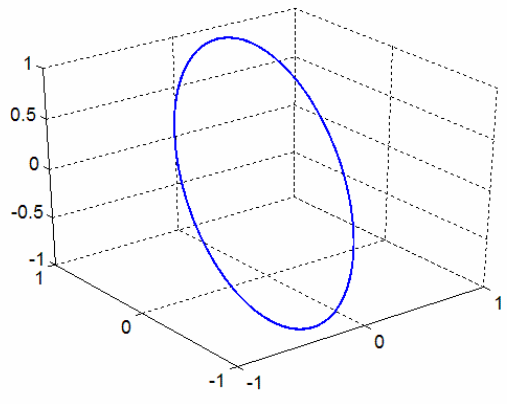


Figure 4.3: The phase space structure of a sine wave seen in three dimensions $[x(n), x(n+2), x(n+4)]$ where $x(t) = 1sin(t)$.

data, require searches and other operation in R^d whose computational cost rises exponentially with d;

2. in the presence of "noise" or other high-dimensional contamination of our observations, the "extra" dimensions are not populated by dynamics, already captured by a smaller dimension, but entirely by the contaminating signal.

In too large an embedding space one is unnecessarily spending time working around aspects of a bad representation of the observations which are solely filled with "noise". This realization has motivated the search for analysis tools that will identify a *necessary* embedding dimension from the data itself. In the next section we will describe some methods for this analysis.

4.7.1 Singular Value Analysis

If our measurements $y(n)$ are composed of the signal from the dynamical system we wish to study plus some contamination from other systems, then in absence of specific information about the contamination it is plausible to assume it to be rather high dimensional and to assume that it will fill more or less uniformly any few dimensional space we choose for our considerations.
Let us call the embedding dimension necessary to unfold the dynamics we seek d_N. If

we work in $d_E > d_N$, then in an heuristic sense $d_E - d_N$ dimensions of the space are being populated by contamination alone. If we think of the observations embedded in d_E as composed of a true signal $y_T(n)$ plus some contamination $\mathbf{c}$: $\mathbf{y}(n) = \mathbf{y}_T(n) + \mathbf{c}(n)$ then the $d_E \times d_E$ sample covariance matrix:

$$COV = \frac{1}{N}\sum_{n=1}^{N}[y(n) - \bar{y}][y(n) - \bar{y}]^T \tag{4.18}$$

with $\bar{y} = \frac{1}{N}\sum_{n=1}^{N} y(n)$,
will, again in an heuristic sense, have d_N eigenvalues arising from the variation of the (slightly contaminated) real signal about its mean and $d_E - d_N$ eigenvalues which represent the "noise". If the contamination is quite high dimensional, it seems plausible to think of it filling these extra $d_E - d_N$ dimensions in some uniform manner, so perhaps one could expect the unwelcome $d_E - d_N$ eigenvalues, representing the power in the extra dimensions, to be nearly equal. If this were the case, then by looking at the eigenvalues or equivalently the singular values of COV, we might hope to find a "noise floor" at which the eigenvalue spectrum turned over and became flat. There are d_E eigenvalues and the one where the floor is reached may be taken as d_N.
This analysis can also be carried out *locally* [], which means that the covariance matrix is over a neighborhood of the N_B nearest neighbors $\mathbf{y}^{(r)}(n)$ of any given data point $\mathbf{y}(n)$:

$$COV(n) = \frac{1}{N_B}\sum_{r=1}^{N_B}[y^{(r)}(n) - \bar{y}(n)][y^{(r)}(n) - \bar{y}(n)]^T \tag{4.19}$$

with $\bar{y} = \frac{1}{N_B}\sum_{r=1}^{N_B} y^{(r)}(n)$.
The global singular - value analysis has the attractive feature of being easy to implement, but it has the downside of being hard to interpret. It gives a linear hin as to the number of active degrees of freedom, but it can be misleading because it does not distinguish two process with nearly the same Fourier spectrum.

4.7.2 False Nearest Neighbors

The False Nearest Neighbors Method [53] for determining d_N comes from asking, directly of the data, the basic question addressed in the embedding theorem. When has one eliminated false crossings of the orbit with itself which arose by virtue of having

projected the attractor into a too low dimensional space?
Answer to this question have been discussed in various ways. Each of the ways has addressed the problem of determining when points in dimension d are neighbors of one another by virtue of the projection into too low a dimension.
By examining this question in dimension one, then dimension two, etc. until there are no incorrect or false neighbors remaining, one should be able to establish, from geometrical considerations alone, a value for the necessary embedding dimension $d_E = d_N$. We describe the implementation of Kennel et al [53].
In dimension d each vector

$$\mathbf{y}(k) = [x(k), x(k+\tau), ..., x(k+(d-1)\tau)] \tag{4.20}$$

has a nearest neighbor $\mathbf{y}^{NN}(k)$ with nearness in the sense of some distance function. Euclidean distance is natural and works well. The Euclidean distance in dimension d between $\mathbf{y}(k)$ and $\mathbf{y}^{NN}(k)$, that we denote with $R_d(k)$

(4.21)

$$\begin{aligned} R_d(k)^2 = \left[x(k) - x^{NN}(k)\right]^2 + \left[x(k+\tau) - x^{NN}(k+\tau)\right]^2 + ... \\ + \left[x(k+(d-1)\tau) - x^{NN}(k+(d-1)\tau)\right]^2 \end{aligned} \tag{4.22}$$

$R_d(k)$ is presumably small when one has a lot of data and for a data set with N entries, this distance is more or less of order $1/N_{1/d}$. In dimension $d+1$ this nearest neighbor distance is changed due ti the $(d+1)^{st}$ coordinates $x(k+d\tau)$ e $x^{NN}(k+d\tau)$ to

$$R_{d+1}(k)^2 = R_d(k)^2 + \left[x(k+d\tau) - x^{NN}(k+d\tau)\right]^2 \tag{4.23}$$

If $R_{d+1}(k)$ is large, we can presume it is because the near neighborliness of the two points being compared is ude to the projection from some higher dimensional attractor down to dimension d. By going from dimension d to dimension $d+1$, we have "unprojected" these two points away from each other. Some threshold size R_T is required to decide when neighbors are false. Then if

$$\frac{\left[x(k+d\tau) - x^{NN}(k+d\tau)\right]^2}{R_d(k)} > R_t \tag{4.24}$$

the nearest neighbors at time point k are declared false.
The criterion stated so far for false nearest neighbors has a subtle defect. If one applies

it to data from a very high dimensional random number generator, it indicates that this set of observations can be embedded in a small dimension. If one increase the number of points analyzed, the apparent embedding dimension rises. The problem is that when one tries to populate "uniformly" (as "noise" will try to do) an object in d dimensions with a fixed number of points, the points must move further and further apart as d increases because most of the volume of the object us at large distances. If we had an infinite quantity of data, there would be no problem, but with finite quantities of data eventually all points have "near neighbors" that do not move apart very much as dimension is increased.

4.7.3 Cao's Method

The method due to Cao [13] overcomes the shortcomings of this basic methods and in particular the problem of threshold selection of the false neighbor method. Infact similar to the idea of the false neighbor method [53], we define:

$$a(i,m) = \frac{\left\| y_i(m+1) - y_{n(i,m)}(m+1) \right\|}{\left\| y_i(m) - y_{n(i,m)}(m) \right\|}, \quad for\ i = 1, 2, ..., N-(m-1)\tau \qquad (4.25)$$

where $\|.\|$ is some measurement of the Euclidian distance, usually the maximum norm, y_i is the i-th reconstructed vector with embedding dimension m+1, n(i,m) (1$\leq$ n(i,m) $\leq$ N-$m\tau$) is an integer such that $y_{n(i,m)}$(m) is the nearest neighbor of y_i(m) in the m-dimensional reconstructed phase space in the sense of distance $\|.\|$ we defined above. Notes that n(i,m) depends on i and m, and the n(i,m) in the numerator in equation 4.25 is tha same as that in the denominator.

If m is qualified as an embedding dimension by the embedding theorem [70], then any two points which stay close in the m-dimensional reconstructed space will be close in the *(m+1)*-dimensional reconstructed space. Such a pair of points are called true neighbors, otherwise they are called false neighbors. Perfect embedding means that no false neighbors exist. This is the idea of the false neighbor method in [53], where the authors diagnosed a false neighbor by seeing whether their (slightly different) version of a(i,m) is larger than some given threshold value. The problem is how to choose this threshold value. To avoid this problem, Cao [13] in his method define the following

quantity, i.e. the mean value of all a(i,m),

$$E(m) = \frac{1}{N - m\tau} \sum_{i=1}^{N-m\tau} a(i, m) \tag{4.26}$$

$E(m)$ is dependent only on the dimension m and the lag τ. To investigate its variation from m to $m+1$, we define

$$E1(m) = \frac{E(m+1)}{E(m)} \tag{4.27}$$

We found that E1(m) stops changing when m is greater than some value m_0 if the time series comes from an attractor. So m_0+1 is the minimum embedding dimension we look for. In [13], it is defined another quantity which is useful to distinguish deterministic signals from stochastic signals. Let

$$E^*(m) = \frac{1}{N - m\tau} \sum_{i=1}^{N-m\tau} \left| x_{i+m\tau} - x_{n(i,m)+m\tau} \right| \tag{4.28}$$

From this it is possible to define

$$E2(m) = \frac{E^*(m+1)}{E^*(m)} \tag{4.29}$$

The introduction of E2(m) is justified by the fact that for time series data from a random set of numbers, E1(m), in principle, will never attain a saturation value as m increases. But in practical computations, it is difficult to resolve whether the E1(m) is slowly increasing or has stopped changing if m is sufficiently large. To solve this problem, it is possible to consider E2(m). For random data, since the future values are independent of the past values, E2(m) will be equal to one for any m. However for deterministic data, E2(m) is certainly related to m, as a result, it cannot be a constant for all m.

It is recommended calculating both E1(m) and E2(m) for determining the minimum embedding dimension of a scalar time series and to distinguish deterministic data from random data.

4.8 Choosing T and d_E

The dertemination of the appropriate phase space in which to analyze chaotic signals is one of the first tasks, and certainly a primary task, for all who wish to work with

observed data in the absence of detailed knowledge of the system dynamics.
To determine the time lag to be used in an embedding, one may always wish to use something nonlinear, such as average mutual information, but the data may mitigate against that. If one has sampled a map, achieved stroboscopically or taken as a Poincaré section, there is typically no minimum in the average mutual information function. The reason is quite simple: the time between samples τ_s is so long that the orbit has become decorrelated, in an information - theoretic sense.
For solving this problem there are two opportunity, the first, if it is possible, is to resample the data. The second is to turn to the autocorrelation function of the time series to find at least an estimate of what one can reliably use for a time delay in state-space reconstruction. While the criterion is linear, it may not be totally misleading to use the first zero crossing of the autocorrelation function as a useful time lag. When the average mutual information does have a first minimum, it is usually more or less the same order, in units of τ_S, as the first zero crossing of the autocorrelation, so one is not likely to be terribly misled by this tactic.
Once a time delay has been agreed upon, the embedding dimension is the next order of business. In [1], the authors state that is better to work with algorithms that are geometric rather then derivative from the data. Computing correlation functions $C_q(r)$ not only requires a large data set, it also degrades rapidly when the data are contaminated. If one wishes to know whether to use dimension d or d+1, then geometric methods will allow a way to start the selection. In any case, robustness seems to come with methods that do not require precise determination of distances between points on the strange attractor.

4.9 ICA on a single mixture by projection

As mentioned in the introduction of this chapter, in [42] the authors developed a methodology for the extraction of multisource brain activity using only single channel recordings of electromagnetic (EM) brain signals. At the hearth of the method is dynamical embedding, where first an appropriate embedding matrix is constructed out of a series of delay vectors from the measured signal. The approach considered a SVD to accomplish phase space reconstruction and a ICA based approach to separate the

signals.

In our case we use the methods introduced to analyze the phase space and the FatICA algorithm [38] to separate the signals. The FastICA is a fixed-point algorithm developed by Hyvärinen to perform the BSS using the negentropy information [38].

In this case the approach is composed by two steps.

In the first step we determine the embedding dimension using the Cao's method and the time lag using the average mutual information to obtain the time delayed mixtures.

In the second step, obtained the matrix of time delayed mixtures as shown in equation 4.30

$$\begin{matrix} x_1, & x_2, & \dots, & x_{N-(m-1)\tau} \\ x_{1+\tau}, & x_{2+\tau}, & \dots, & x_{N-(m-1)\tau+\tau} \\ \vdots & \vdots & \vdots & \vdots \\ x_{1+(m-1)\tau}, & x_{2+(m-1)\tau}, & \dots, & x_N \end{matrix} \tag{4.30}$$

we apply the FastICA approach on this.

4.10 Conclusions

In this chapter, we first introduced some methodologies for the analysis of the embedding dimension of a time series. We described the most used method for the recovering of the time delay and the embedding dimension of the time series. We gave the detail of this method and how it is possible to use them in the case of the independent component analysis on single channel. We must note that the projection given from the application of the dynamical system theory gives us a matrix of vectors, where each vector is a shifted version of the original signal, where the time delay and the embedding dimension determine that shift.

Applying FastICA on that vector gives us the possibility to overcome the single channel mixture.

We made several experiments using that method applying it on Musical and Gravitational data. Detail of the experiments are shown in the nexts chapters.

Chapter 5

ICA on Single Mixture: a Non Linear Principal Component Analysis method.

In the previous chapter, we described the methodologies for the analysis of a time series. We explained the detail of some methods for the extraction from the data of two parameters: a time delay and an embedding dimension. In this chapter, we explain a new model for the separation of independent component analysis on single mixture based on the integration of a Non Linear Principal Component Analysis neural network, with the parameters found by the time series analysis.

5.1 Introduction

Principal Component Analysis (PCA) is a well-known, widely used statistical technique. Essentially, the same basic technique is used in several areas under different names, such as Karhunen - Loeve transform or expansion, Hotelling transform and signal subspace or eigenstructure approach.

In pattern recognition, PCA is used in various forms for optimal feature extraction and data compression [46]. In image processing, PCA defines the Hotelling or KL transform, that is optimal in image data compression. In signal processing, a useful characterization of signals is to assume that they roughly lie in the signal subspace de-

fined by PCA. Several modern methods of signal modeling, spectrum estimation and array processing are based on this concept.

5.2 Basic Mathemathics

Let $\mathbf{x}$ be an L-dimensional data vector coming from some statistical distribution centralize to zero: $E\{\mathbf{x}\} = 0$. The **i-**th principal component $\mathbf{x}^T\mathbf{c}(i)$ of $\mathbf{x}$ is defined by the normalized eigenvector $\mathbf{c}(i)$ of the data covariance matrix $\mathbf{C} = E\{\mathbf{x}\mathbf{x}^T\}$ associated with the **i-**th largest eigenvalue $\lambda(i)$. The subspace spanned by the principal eigenvectors $\mathbf{c}(1), ..., \mathbf{c}(M)$ $(M < L)$ is called the PCA subspace (of dimensionality M).
PCA networks are neural realizations of PCA in which th weight vectors $\mathbf{w}(i)$ of the neurons or the weight matrix $\mathbf{W} = [\mathbf{w}(1), ..., \mathbf{w}(M)]$ converge to the principal eigenvectors $\mathbf{c}(i)$ or to the PCA subspace during the learning phase.
It is well known that standard PCA emerges as the optimal solution to several different information representation problems. These include:

1. maximization of linearly transforme variances $E\{[\mathbf{w}(i)^T\mathbf{x}]^2\}$ or outputs of a linear network under orthonormality constraints ($\mathbf{W}\mathbf{W}^T = \mathbf{I}$);

2. minimization of the mean-square representation error $E\{\|\mathbf{x} - \hat{\mathbf{x}}\|^2\}$, when the input data $\mathbf{x}$ are approximated using a lower dimensional linear subspace $\hat{\mathbf{x}} = \mathbf{W}\mathbf{W}^T\mathbf{x}$;

3. uncorrelatedness of outputs $\mathbf{w}(i)^T\mathbf{x}$ of different neurons after orthonormal transform ($\mathbf{W}\mathbf{W}^T = \mathbf{I}$);

4. minimization of representation entropy.

Derivation of the optimal PCA solutions with the required assumptions and constraint conditions can be found in several paper [57].
In the next section, we briefly consider the relative merits and shortcomings of linear and nonlinear PCA networks and algorithms. Various robust and nonlinear extensions of neural PCA are introduced by generalizing each of the above mentioned quadratic optimization criteria, which lead to standard PCA solution [51]. Such an approach

gives a sound mathematical foundation to the generalizations and helps to understand the properties of the corresponding learning algorithms. The main attention is devoted to the first two criterions, for which we derive several new learning algorithms.
Another typical approach to nonlinear PCA has been just to insert a nonlinearity somewhere in a PCA network and see what happens, or to propose some other heuristic modification. The result of such heuristic algorithms are more difficult to interpret. A third approach is to start from some fixed neural network structure and study what kind of algorithms can be realized using it. Sometimes this approach lead to the same learning algorithms that are obtained from suitable optimization criteria.

5.3 Linear and Non Linear Neural PCA

It is now well known that relatively simple, neurobiologically justified Hebbian-type learning rules can provide PCA. This, togheter with the usefulness and many applications of PCA, has prompted a lot of interest in various realizations of PCA [59]. However, PCA networks and learning algorithms have some limitations that diminish their attractiveness:

1. Standard PCA networks are able to realize only linear input-output mappings.

2. The eigenvectors needed in standard PCA can be computed efficiently using well-known numerical methods. Gradient type neural PCA learning algorithms converge relatively slowly and achieving a good accuracy requires an ecessive number of iterations in large problems.

3. Principal Components are defined solely by the data covariances (or correlations). These second-order statistics characterize completely only Gaussian data and stationary, linear processing operations.

4. PCA networks cannot usually separate independent subsignals from their linear mixture.

If a PCA-type network contains nonlinearities, the situation becomes much more favorable for a neural realization.

First, the input-output mapping becomes generally nonlinear, which is a major argument for using neural networks. Nonlinear processing of the data is often more efficient, and the properties of standard linear methods have been explored thoroughly.
Second, neural algorithms become much more competitive or may be the only possibility for heuristic learning principles. In optimizing nonquadratic criteria, one must resort to iterative algorithms anyway, because efficient closed form solutions are usually not available.
The third motivation of using nonlinearities is that they introduce in an implicit way higher-order statistics into the computations. This can be seen by expanding the nonlinearities into their Taylor series. Higher order statistics, defined by cumulants and higher than second moments are needed for a good characherization of non-Gaussian data. There exist several important problems that cannot adequately be solved using merely second-order statistics.
Fourth, the outputs of standard PCA networks are usually at most mutually uncorrelated but not independent, which would be more desiderable in many cases. In Karhunen and Joutensalo [50], the authors have demonstrated that adding nonlinearities to a PCA network increases the independence of the outputs, so that the original signals can sometimes be roughly separated by their mixture. Naturally, nonlinear PCA type networks have some drawbacks compared to the linear ones. The mathematical analysis of the learning algorithms is often inherently difficult, making the properties of the networks less well understood. The nonlinear learning algorithms are more complicated and may sometimes be caught more easily in local minima. Adding nonlinearities to a neural network does not help automatically or in all the problems. For some nonquadratic criteria the final input - output mapping is still linear, because the nonlinearities appear in the learning rule only.
Another important characterization of the nonlinear PCA is that the learning algoriths are divided into symmetric and hierarchic, in a way quite similar to those for standard PCA networks. In standard PCA learning algorithms, some kind of hierarchy or differentiation is necessary between the learning rules of different neurons to get the prinicipal components or eigenvectors themselves. The completely symmetric algorithms yield PCA subspace and some linear combinations of principal components only. It seems that in nonlinear PCA networks hierarchy is not so important, because

nonlinearities break the complete symmetry during learning and the outputs of symmetric networks become more unique as in the linear case [59, 50].
The learning algorithms derived considering generalizations of the optimization problems leading to standard PCA can be divided into two classes in another way. We distinguish between the so-called *robust* PCA algorithms [51, 23] and *nonlinear* PCA algorithms. We define robust PCA so that the criterion to be optimized grows less than quadratically and the constraint conditions are the same as for the standard PCA solution, which emerges from the respective quadratic criterion. Typically, the weight vectors of the neurons are required to be mutually orthonormal. Robust PCA problems usually lead to mildly nonlinear algorithms, in which the nonlinearities appear at selected places only. More specifically, at least some of the outputs of the neurons are still their linear responses $y(i) = \mathbf{x}^T\mathbf{w}(i)$, where $\mathbf{w}(i)$ is the weight vector of the ith neuron. In the nonlinear PCA algorithms all the outputs $g\,[y(i)]$ of the neurons are nonlinear functions of the response.

The structure for the nonlinear PCA network is shown in figure 5.1 for the symmetric

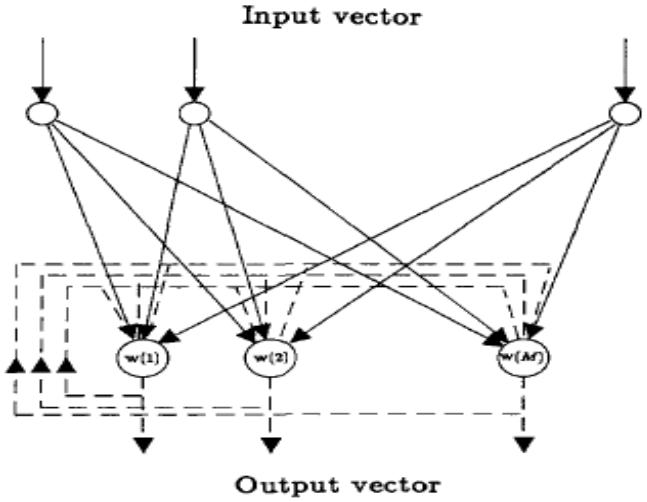


Figure 5.1: Architecture of the symmetric network for NLPCA. Feedback connections (dashed lines) are needed in the learning phase only.

case and in figure 5.2 for the standard hierarchic arrangement. The network contains input and output layers only. After learning, the feedback connections between outputs and inputs shown by dashed lines in the figures are not needed and the network becomes purely feedforward. The same structure can be used for all the algorithms,

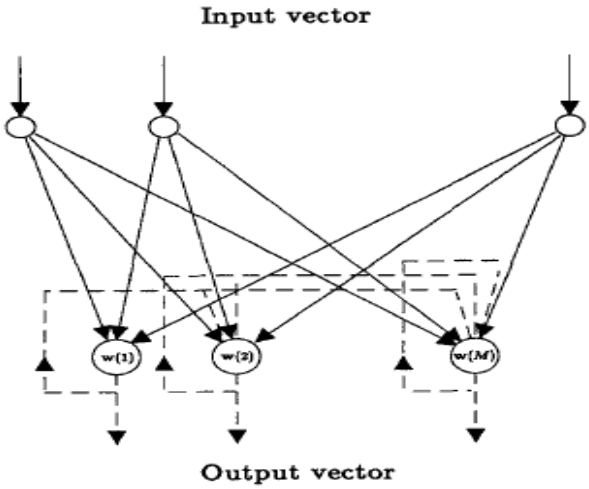


Figure 5.2: Architecture of the hierarchic network for NLPCA. Feedback connections (dashed lines) are needed in the learning phase only.

but details of the realization vary.

5.4 Generalization of variance maximization

The standard quadratic problem leading to a PCA solution is one of how to maximize the output variances $E\{y(i)^2\} = E\{[\mathbf{w}(i)^T\mathbf{x}]^2\} = \mathbf{w}(i)^T\mathbf{C}\mathbf{w}(i)$ of the linear network under orthonormality constraints.

The number of neurons M is assumed to be less than or equal to the dimension L of the data vectors $\mathbf{x}$. The maximization problem is not well defined unless the nonrandom L-dimensional weight vectors $\mathbf{w}(i)$ of the neurons are constrained somehow. In lack of prior knowledge, orthonormality constraints are the most natural, because they measure the variances along maximally different directions.

Normally, the ith weight vector $\mathbf{w}(i)$ is constrained so that it must have unit norm and be orthogonal to the weight vector $\mathbf{w}(j)$, $j = 1, \ldots, i-1$ of the previous neurons. These constraints take the mathematical form $\mathbf{w}(i)^T\mathbf{w}(j) = \delta_{ij},\ \ j \leq i$, where the Kronecker delta $\delta_{ij} = 1,\ for\ i = j$ and $0\ for\ i \neq j$. The optimal $\mathbf{w}(i)$ is then the ith principal eigenvector $\mathbf{c}(i)$ of $\mathbf{C}$ and the outputs of the PCA network become the principal component of the data vectors. The PCA network and the learning algorithms are in

this case hierarchic. In the following, we refer to this constraint set and case as the standard hierarchic case.

The respective variance maximization problem can be solved for symmetric orthornormality constraints $\mathbf{w}(i)^T\mathbf{w}(j) = \delta_{ij},\ j \leq i$, as well. It is convenient to define the $L \times M$ weight matrix $\mathbf{W} = [\mathbf{w}(1), \ldots, \mathbf{w}(M)]$, for which columns are the weight vectors of the M neurons. The symmetric orthonormality constraints then become $\mathbf{W}^T\mathbf{W} = \mathbf{I}$, where $\mathbf{I}$ is the unit matrix. The optimal solution is now given by any orthonormal basis spanning the PCA and is thus not unique. This version of the variance maximization problem leads to PCA subspace networks and learning rules. We refer to this case and constraint set as the standard symmetric case.

Consider now generalization of the variance maximization problem for robust PCA. Instead of using the standard mean - square value, we can maximize a more general expecatation $E\{f[\mathbf{x}^T\mathbf{w}(i)]\}$ of the response $\mathbf{x}^T\mathbf{w}(i)$ of the ith neuron. The function $f(t)$ is assumed to be a valid cost function that grows less than quadratically, at least for large values of t. More specifically, we assume that $f(t)$ is even, nonnegative, continuously differentiable almost everywhere and $f(t) \leq t^2/2$ for large values of $|t|$. Furthermore, its only minimum is attained at $t = 0$ and $f(t_1) \leq f(t_2)$ if $|t_1| < |t_2|$. Some of these assumptions are not absolutely necessary. Examples of such a function are $f(t) = ln\, cosh(t)$ and $f(t) = |t|$ [51].

The criterion to be maximized is then for each neuron weight vector $\mathbf{w}(i)$, $i = 1, \ldots, M$ of the form

$$J_1\left[\mathbf{w}(i)\right] = E\left\{f\left[\mathbf{x}^T\mathbf{w}(i)\right]\right\} + \sum_{j=1}^{I(i)} \lambda_{ij}\left[\mathbf{w}(i)^T\mathbf{w}(j) - \delta_{ij}\right] \tag{5.1}$$

Here the summation imposes via the Lagrange multipliers $\lambda_{ij} = \lambda_{ji}$ the necessary orthonormality constraints $\mathbf{w}(i)^T\mathbf{w}(j) = \delta_{ij}$. Both the hierarchic and symmetric problems can be discussed under the same general criterion 5.1. In the standard symmetric case, the upper bound of the summation index is $I(i) = M$ for all $i = 1, \ldots, M$. In the standard hierarchic case $I(i) = i$; the optimal weight vector of the ith neuron defines then the robust counterpart of the ith principal eigenvector $\mathbf{c}(i)$. One advantage in using hierarchic networks is that the order of the neurons could be permuted.

However the two basic cases described above are the most relevant ones and we concentrate on them in the following.

The gradient of $J_1[\mathbf{w}(i)]$ with respecto to $\mathbf{w}(i)$ is

$$\mathbf{h}(i) = \frac{\partial J_1(\mathbf{w}(i))}{\partial \mathbf{w}(i)} = \tag{5.2}$$

$$= E\left\{\mathbf{x} g\left[\mathbf{x}^T \mathbf{w}(i)\right]\right\} + 2\lambda_{ij}\mathbf{w}(i) + \sum_{j=1, j\neq i}^{I(i)} \lambda_{ij}\mathbf{w}(j)$$

where $g(\cdot)$ is the derivative $df(\cdot)/dt$ of $f(\cdot)$.

At the optimum, the gradients must vanish for $i = 1, \ldots, M$. Differentiation with respect to the Lagrange multipliers yields the orthonormality constraints

$$\mathbf{w}(i)^T \mathbf{w}(j) = \delta_{ij}, \quad j = 1, \ldots, I(i) \tag{5.3}$$

which must also be satisfied at the optimum. The optimal values of the Lagrange multipliers can be determined by multiplying equation 5.3 by $\mathbf{w}(j)^T$, $j = 1, \ldots, I(i)$, from the left, and equating the result to zero. Taking into account the equation 5.3, this yields to $\lambda_{ij} = -\mathbf{w}(j)^T E\left\{\mathbf{x} g\left[\mathbf{x}^T \mathbf{w}(i)\right]\right\}$ for $i \neq j$ and $\lambda_{ii} = -\frac{1}{2}\mathbf{w}(i)^T E\left\{\mathbf{x} g\left[\mathbf{x}^T \mathbf{w}(i)\right]\right\}$. Inserting these values into equation 5.3, we get

$$\mathbf{h}(i) = \left[\mathbf{I} - \sum_{j=1}^{I(i)} \mathbf{w}(j)\mathbf{w}(j)^T\right] E\left\{\mathbf{x} g\left[\mathbf{x}^T \mathbf{w}(i)\right]\right\} \tag{5.4}$$

A practical stochastic gradient algorithm for maximizing equation 5.1 is now obtained by inserting the estimate $\mathbf{h}_k(i)$ of the gradient vector in equation 5.4 at step k into the update formula

$$\mathbf{w}_{k+1}(i) = \mathbf{w}_k(i) + \mu_k \mathbf{h}_k(i) \tag{5.5}$$

Here the μ_k is the gain parameter.

In the practice, we use the standard instantaneous gradient estimates. They are obtained simply by omitting the expectations and using instead of them the instantaneous values of the quantities in question.

The final algorithm thus becomes

$$\mathbf{w}_{k+1}(i) = \mathbf{w}_k(i) + \mu_k \left[\mathbf{I} - \sum_{j=1}^{I(i)} \mathbf{w}_k(j)\mathbf{w}_k(j)^T\right] \mathbf{x}_k g\left[\mathbf{x}_k^T \mathbf{w}_k(i)\right] \tag{5.6}$$

The assumptions made earlier on the cost function $f(\cdot)$ imply that its derivative $g(\cdot)$ appearing in equation 5.6 should be an odd, nondecreasing (often monotonically growing) function. For stability reason, it is at least necessary to assume that $g(t) \leq 0$, for

$t < 0$ and $g(t) \geq 0$, for $t > 0$ [59].

Defining the instantaneous representation error vector

$$\mathbf{e}_k(i) = \mathbf{x}_k - \sum_{j=1}^{I(i)} \left[\mathbf{x}_k^T \mathbf{w}_k(j)\right] \mathbf{w}_k(j) = \mathbf{x}_k - \sum_{j=1}^{I(i)} y_k(j)\mathbf{w}_k(j) \tag{5.7}$$

the algorithm in equation 5.6 can be written in a simpler form

$$\mathbf{w}_{k+1}(i) = \mathbf{w}_k(i) + \mu_k g\left[y_k(i)\right] \mathbf{e}_k(i) \tag{5.8}$$

From equation 5.7 and equation 5.8, one can easily see that no matrix multiplications are needed in the actual realization.

In the symmetric case $I(i) = M$, for $i = 1, \ldots, M$, the error vector $\mathbf{e}_k(i)$ becomes the same $\mathbf{e}_k$ for all the neurons. Then equation 5.6 can be expressed compactly in the matrix form

$$\mathbf{W}_{k+1}(i) = \mathbf{W}_k + \mu_k \left[\mathbf{I} - \mathbf{W}_k \mathbf{W}_k^T\right] \mathbf{x}_k g\left[\mathbf{x}_k^T \mathbf{W}_k\right] = \mathbf{W}_k + \mu_k \mathbf{e}_k g(\mathbf{y}_k^T) \tag{5.9}$$

where $\mathbf{y}_k = \mathbf{W}_k^T \mathbf{x}_k$ is the instantaneous response vector. The function $g(\cdot)$ is applied separately to each component of its argument vector. The algorithm in equation 5.9 coincides with the well-known Oja's PCA subspace rule [22, 57, 51] in the linear special case $g(t) = t$.

Otherwise, equation 5.9 defines a robust generalization of Oja's rule that was first proposed quite heuristically at the end of the paper by Oja et al. [59].

In the standard hierachic case $I(i) = i$, so equation 5.9 can be written in the matrix form

$$\mathbf{W}_{k+1}(i) = \mathbf{W}_k + \mu_k \left\{\mathbf{x}_k g\left(\mathbf{y}_k^T\right) - \mathbf{W}_k \mathbf{UT}\left[\mathbf{y}_k g(\mathbf{y}_k^T)\right]\right\} \tag{5.10}$$

where the upper triangular operator $\mathbf{UT}$ sets the elements of its argument matrix to zero below the diagonal. In the linear special case $g(t) = t$, equation 5.10 coincides exactly with the well-known GHA algorithm [22, 57, 51] proposed originally by Sanger [66, 65]. Otherwise, equation 5.10 defines a robust generalization of the GHA algorithm. Another, more practical formulation of equation 5.10 is obtained by noting that the error vector in equation 5.7 can be expressed in the standard hierarchic case recursively as $\mathbf{e}_k(i) = \mathbf{e}_k(i-1) - y_k(i)\mathbf{w}_k(i)$, with $\mathbf{e}_k(0) = \mathbf{x}_k$. This show that robust GHA can be implemented locally in a similar manner as standard GHA [65].

5.5 Independent component analysis using non linear PCA network

In this section, we will describe how it is possible to obtain the standard ICA problem starting from a non linear PCA network.
We can consider a single mixture data vector as

$$\mathbf{x}_k = \left(x\left[k\right], x\left[k+1\right], \ldots, x\left[k+L-1\right]\right)^T \tag{5.11}$$

formed of L successive samples. We note that L is the number of the neural network (NN) inputs. We suppose to find the p principal eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_p$ corresponding to the p largest eigenvalues $\lambda_1 \geq \ldots \geq \lambda_p$ (number of outputs in the NN). In other words, we have

$$\mathbf{R}_{xx}\mathbf{u}_i = \lambda_i \mathbf{u}_i \tag{5.12}$$

The autocorrelation matrix on the data vectors $\mathbf{x}_k$ of equation 5.11 is

$$\mathbf{R}_{xx} = \frac{1}{K}\sum_{k=1}^{K-1} \mathbf{x}_k \mathbf{x}_k^H \tag{5.13}$$

Now, inserting equation 4.11 into equation 5.12 yelds

$$\lambda_i \mathbf{u}_i \approx \frac{1}{K}\sum_{k=1}^{K-1} \left(\mathbf{x}_k^H \mathbf{u}_i\right) \mathbf{x}_k \tag{5.14}$$

Thus, the true eigenvectors are approximately some linear combinations of the data vectors $\mathbf{x}_k$ [49]. So it is possible to write equation 5.14 as

$$\mathbf{v}_i = \sum_{k=1}^{K-1} g_{ik} x_k \tag{5.15}$$

where $i = 0, \ldots, p-1$. In matrix form, we can also write

$$\mathbf{V} = \mathbf{G}\mathbf{X} \tag{5.16}$$

where $\mathbf{V}$ is a $p \times (L-1)$ matrix, $\mathbf{G}$ is a $p \times K$ mixing matrix and $\mathbf{X}$ is the data vector. In details we have that

$$\mathbf{v}_i = \mathbf{g}_i \times \mathbf{x}_1 \tag{5.17}$$

$$\begin{aligned} &\mathbf{x}_2 \\ &\dots \\ &\mathbf{x}_K \end{aligned} \tag{5.18}$$

where we have that in our case $\mathbf{x}_j = (as_j^1 + bs_j^2)$ and where the source signals are $s_j^n = (s^n[j], s^n[j+1], \dots, s^n[j+L-1])^T$. We can note that also in this case and it is clearer from equation 5.16, we obtain the standard ICA problem.

5.6 Use of the embedding dimension

It is important to note that in the approach proposed in this work, we made an integration between the described model of non linear PCA network and the embedding dimension.

In fact, in standard ICA, each source can be separated and reconstructed in the observation domain through the operation

$$\mathbf{x}_{s_i} = A_{(:,i)} W_{(i,:)} \mathbf{x} \tag{5.19}$$

where $\mathbf{x}_{s_i}$ is the i-th source in the observation domain.

With a single channel of data, we can apply the same formula to data blocks giving

$$\mathbf{x}_{s_i}(nN-k+1) = A_{(:,i)} \sum_{j=1}^{N} W_{(i,j)} \mathbf{x}(nN-j+1) \tag{5.20}$$

However the resulting source estimates are highly dependent on the block alignment. In our case, we choose the shift of the observation data in according to the embedding dimension of the mixture. This choice is made in order to emphasize the independence between the signal embedded in the mixture and to avoid the problem of the dependence between blocks. As explained in the previous chapter, in fact, with the study of the embedding dimension of a mixture we want to identify quantities that are unchangend when initial conditions on an orbit are altered or when, anywhere along the orbit, pertubation are encountered.

5.7 Characterization of the algorithm proposed

As described in the previous section, in a robust nonlinear PCA NN of fundamental importance is the choise of the parameter of the network such as the number of input nuerons, the number of output neurons and the initial choice of the weight matrix. Our idea was that of using the embedding parameter for the definition of the network model and the initial data guess. In detail, the proposed approach can be divided into the following steps:

- Preprocessing: we first calculate and subtract the average pattern to obtain a zero mean process.

- Neural computing: we calculate the weights vector $\mathbf{w}_i$, for $i = 1, ..., m$, by using equations in Step 4 of the Algorithm 1.
  The fundamental learning parameters are:
  i) the number of output neurons m, which is equal to the embedding dimension and it is the number of principal eigenvectors that we need and the time lag τ needed to build the input patterns;
  ii) the number of input neurons q;
  iii) the initial random weight matrix $\mathbf{W}$ of $m \times q$ dimension;
  iv) α, the nonlinear learning function parameter;
  v) the learning rate μ_k and the ϵ tolerance.

The general algorithm is described in Algorithm 1.

5.8 A case of study: an Armonic Oscillator, the Mackey Glass time series and random Gaussian noise

In the experiment we consider a mixture of a three different signals. The first source is a simple harmonic oscillator with frequency of 15 Hz. The second is obtained by sampling the Mackey-Glass time delay differential equation [1]. This time series is chaotic, and so there is no clearly defined period. The third signal is a random Gaussian noise (see Fig. 5.3). The mixture that we analyze is plotted in Fig. 5.4. We note that for the single harmonic oscillator we have the time lag $\tau = 2$ and the embedding dimension $m = 2$, while for the Mackey-Glass $\tau = 17$ and $m = 3$. Applying the phase reconstruct

Algorithm 1 Embedded Robust PCA Algorithm

1: Initialize m to the embedding dimension calculated in the first step. Initialize the weight matrix $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_m]$ with small random values. Initialize the learning threshold ϵ, the learning rate μ_k (that generally is exponential decrescent and depends from the epoch key)and the α parameter.
Reset epoch counter $k = 1$ and pattern counter $n = 1$.

2: Input the $n - th$ pattern

$$\mathbf{x}_n = [x(n), x(n+\tau), \ldots., x(n+(m-1)\tau)]$$

where m is the number of input components and τ is the time lag.

3: Calculate the output for each neuron $y_i = \mathbf{w}_i^T \mathbf{x}_n$, $\forall i = 1, \ldots, m$.

4: Modify the weights using the following equation

$$\mathbf{w}_i(k+1) = \mathbf{w}_i(k) + \mu_k g(y_i(k))\mathbf{e}_i(k)$$

where

$$\mathbf{e}_i(k) = \mathbf{x}_n - \sum_{j=1}^{I(i)} y_j(k)\mathbf{w}_j(k)$$

and

$$\mathbf{w}_i(k+1) = \frac{\mathbf{w}_i(k+1)}{\|\mathbf{w}_i(k+1)\|}$$

where $g(.)$ is the derivative of the cost function $f(.)$. In the hierarchical case we have $I(i) = i$. In the symmetric case $I(i) = m$, the error vector $\mathbf{e}_i(k)$ becomes the same $\mathbf{e}_i$ for all the neurons.

5: $n = n + 1$.

6: UNTIL $n \leq m$ GO TO 2

7: Convergence test:
IF $C_T = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(w_{ij} - w_{ij}^{old})^2 < \epsilon$
THEN GO TO 8
ELSE
Make orthonormalization:

$$\mathbf{W} = (\mathbf{W}\mathbf{W}^T)^{\frac{1}{2}}\mathbf{W}$$

$$\mathbf{W}^{old} = \mathbf{W}$$

8: $k = k + 1$; GO TO 2.

9: END

approach to the mixture, we obtain $\tau = 1$ and $m = 8$ and they are the parameters that we use to reconstruct the signals and to determine the PCA NN architecture. In Fig. 5.5a and in Fig. 5.5b we show the separated signals obtained by using the FastICA based approach and the Robust PCA approach, respectively. However in the case of the

Robust PCA we have to note that the separation is clearer than the other. This can be shown in Fig. 5.6a and 5.6b where we show and compare the source and the estimated signals. In this case we also calculate the correlation coefficients between the signals. In the case of the harmonic oscillator the ICA based approach has a correlation of 67% while the Robust PCA based approach of 95%. In the case of the Mackey-Glass we have a correlation of 28% for the ICA based approach while the Robust PCA method of 83%.

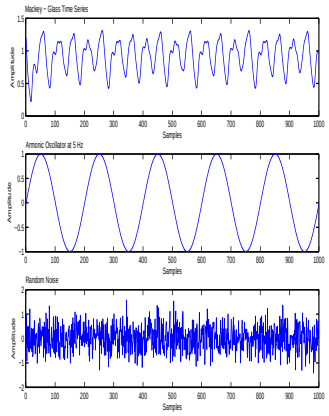


Figure 5.3: Source signals: Single harmonic oscillator (up); Mackey-Glass time series (middle); random Gaussian noise (down).

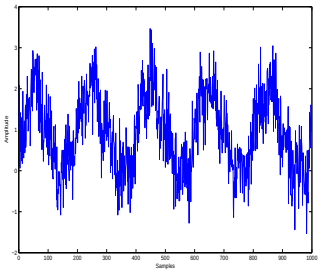
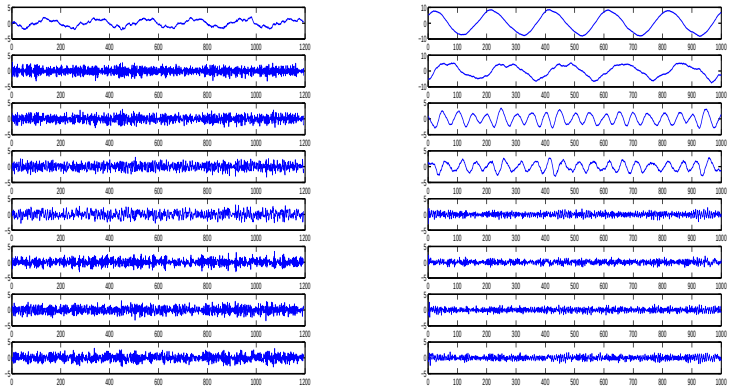
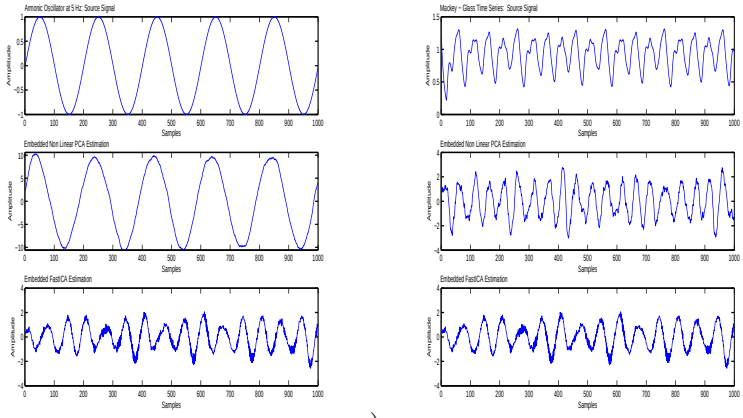


Figure 5.4: Mixture of Mackey Glass time series, single armonic oscillator and random Gaussian noise.



a) b)

Figure 5.5: Separated signals: a) FastICA based algorithm; b) Robust PCA based approach.



a) b)

Figure 5.6: Separated signals: a) Harmonic oscillator estimation: source signal (up), Robust PCA NN based approach estimation (middle), FastICA based approach (down) ; b)Mackey-Glass estimation.

Chapter 6

Applications on Data Coming from Virgo Interferometer

In the previous chapter, we presented a model for the separation of single channel independent components. In this chapter, we show an application of this model to the case of data coming from Virgo interferometer, for the detection of gravitational wave signals.

6.1 Introduction

Gravitational Wave (hereafter GW) detection is certainly one of the most challenging goals for today physics: a very strong proof in favor of the Einstein General Relativity description of phenomena related to the dynamics of gravitation and the opening of a completely new channel of information on astrophysical objects [67]. The VIRGO/LIGO/GEO/TAMA ([2], [68], [71], [69]) network of ground-based kilometer-scale laser interferometer gravitational wave detectors will be the key to open up that new astronomical channel of information in the frequency band 10 Hz to 10 KHz.
Virgo[1] project is an international project (Italian - French), that has as goal the direct detection of the gravitational wave, come out by astrophysical sources, by means of interferometric techniques. Virgo antenna, an interferometer located in Cascina (PI), is listening all gravitational signals coming from all the universe. These signals must be detected from a ground of noise registered by the interferometer. Detecting gravitational wave is a really complex problem, because they are unknown signals with a minimal amplitude (about 10^{-23}). There are two possible application of the separation

[1] http://www.virgo.infn.it

techniques to Virgo data:

- identification of noise source
- identification of gravitational wave signal in noise ground

The first kind of application is the identification of noise source that interacts with the interferometer output. Because of the minimal amplitude of the gravitational wave signal, it is necessary to detect and isolate in the output of the interferometer all the possible sources of external noise (i.e. for example environmental noise). For detecting such noise, on Virgo site has been installed environmental sensors of different nature, for example: seismometer, magnetometer, temperature sensors, pressure sensors and so on.
Data analysis from all these sensors contribute to characterize and identify noise sources inherent of the site, such as noise due to the motor of the various machine present on the site, air movement due to the conditioning or other.
As second kind of application, we can think to the possibility of using blind source separation technique for the detection of the gravitational signal in the ground of noise.

6.2 Detecting gravitational wave signals

As we stated above, the gravitational wave signals have minimal amplitude, but even if these interferometers seem to be sensitive enough for the detection of these sources, nevertheless the problem of GW signal analysis is still in progress, concerning an adequate choice of the data analysis techniques in connection with the shape of the expected signal, the noise of the detector and the available computing power. For this task, many efforts have been made for the development of special data analysis techniques for the enhancement of the signal-to-noise ratio of these GW signals and the most credited algorithm is the matched-filtering technique. This technique, as it is well known [31] [60], requires the correlation of the output of a detector with a template of the expected signal (matched filter). But, although very simple in principle, the application of such algorithm requires a practically exact theoretical knowledge of the shape of the expected signal as function of the unknown parameters which describe the coalescing binary and, then, the correlation of the detector output with several

thousands of templates and these two requirements are very difficult to satisfy for a certain kind of signals coming from coalescing binary signals. The shape of the GW signal can be obtained by computing the gravitational radiation field generated by a system of two point-masses moving on a practically circular orbit. The large number of templates necessary for data analysis using matched-filtering technique poses problems due to the great computing power needed to perform this task on-line. In fact, as a consequence of the large band of these detectors (some kHz), sampling rates of the order of $20\,kHz$ are used, resulting in a huge amount of data/day to be analyzed on-line (of the order of 10 GByte/day). Of course, the analysis of such a large amount of information could be made off-line, but it would be better to select on-line all the data frames which may contain a GW signal. The computational cost depends on the number of parameters considered in the approximation of the phase, on the accuracy of the sampling of the likelihood function (connected with the ability to recover weak signals) and on the actual frequency band to be considered, taking into account the VIRGO sensitivity.

6.3 Whitening

For working with Virgo data it is necessary a preprocessing step for whitening the data. Let $x(t)$ be a wide-sense stationary, continuous-time random process, with mean μ, covariance function:

$$K_x(\tau) \equiv \mathbb{E}\{(x(t_1) - \mu)(x(t_2) - \mu)\}, \quad \tau = t_1 - t_2 \ , \tag{6.1}$$

and power spectral density:

$$S_x(\omega) \equiv \mathcal{F}\left\{K_x(\tau)\right\} = \int_{-\infty}^{\infty} K_x(\tau) \exp(-j\omega\tau) d\tau \ , \tag{6.2}$$

where $\mathcal{F}$ is the Fourier transform.

We can *whiten* the process $x(t)$ by defining a suitable filter $H_w(\omega)$ which transforms the process into a white noise process $w(t)$, whose power spectral density is constant.

Since $K_x(\tau)$ is Hermitian symmetric and positive semi-definite by construction, it follows that $S_x(\omega)$ is real, and can be factored as:

$$S_x(\omega) \equiv |H(\omega)|^2 = H(\omega)H^*(\omega), \tag{6.3}$$

where the star "*" operator denotes complex conjugation. It is possible to show that such representation is possible if and only if $S_x(\omega)$ satisfies the Paley-Wiener condition:

$$\int_{-\infty}^{\infty} \frac{\log S_x(\omega)}{1+\omega^2} d\omega < \infty \ . \tag{6.4}$$

To build the whitening filter, we have to choose a suitable parametric form for $H(\omega)$, which is then adapted to the data. The most common choice is to consider a rational function in zero-pole form:

$$H(\omega) = \frac{\sum_k^N c_k - i\omega}{\sum_k^D d_k - i\omega} \ . \tag{6.5}$$

Choosing the minimum-phase $H(\omega)$ (so that its poles and zeros are on the left half ω plane), the whitening filter will then be stable:

$$H_w(\omega) = \frac{1}{H(\omega)} \ . \tag{6.6}$$

Formally, the whitening operation can then be written:

$$w(t) = \mathcal{F}^{-1}\left\{H_w(\omega)\right\} * \left(x(t) - \mu\right) \ , \tag{6.7}$$

where the star "$*$" denotes convolution. We can show that $w(t)$ is a white process, by showing that its power spectral density is constant:

$$S_w(\omega) \equiv \mathcal{F}\left\{\mathbb{E}\left\{w(t_1)w(t_2)\right\}\right\} = H_w(\omega)S_x(\omega)H_w^*(\omega) = \frac{S_x(\omega)}{S_x(\omega)} = 1 \ . \tag{6.8}$$

In a practical implementation, we will deal with discrete-time processes, however the basic principles are always the same. In particular, we will choose a pole-only function which implies an autoregressive (AR) model of the data. In the following, we have used the maximum entropy (or Burg) algorithm [30] to fit the model coefficients to the data. To assess the model order, we have used the cross-validation criterion [11], and selected the order which gives the highest spectral flatness measure:

$$f = \frac{\exp\left(\frac{1}{2\pi}\int_{-\pi}^{\pi} \log S_w(\omega) d\omega\right)}{\frac{1}{2\pi}\int_{-\pi}^{\pi} S_w(\omega) d\omega} \tag{6.9}$$

6.4 Simulation results for detection

In this section we describe some result of application of the non linear PCA approach to the detection of chirp signal in Virgo noise at different signal to noise ratio (hereinafter

SNR). The result proposed are really interesting because of the small SNR and the possibility to recognize the presence of a signal without knowing nothing about the sources. As a first example, we consider a mixture composed of Virgo noise and a chirp signal with an SNR of 10. In figure 6.1, we show the mixture which we use in the simulation, while in figure 6.2, we show the source signals used to form the mixture. We want to stress that a SNR of 10 is really small, in fact if we compare the source noise with the mixture it is impossible to note where has been injected the signal, that is further observable in figure 6.3 where we superimpose at the noise the chirp signal at SNR 10.

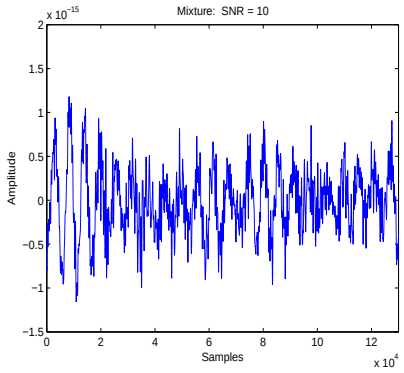


Figure 6.1: Mixture with SNR 10.

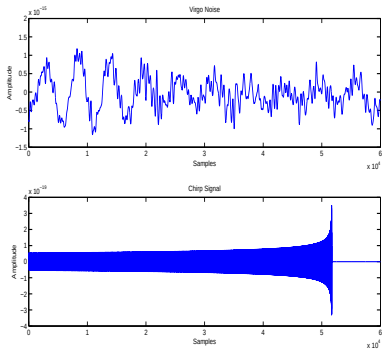


Figure 6.2: Source signal: (up) Virgo noise, (down) chirp signal.

On the mixture presented in figure 6.1, we apply first a whitening process as described in the previous section, getting the mixture 6.4, and then we apply the NLPCA approach to separate the components. After that we was able to recognize clearly the presence of the signal in the noise as it is shown in the figure 6.5(a-b).

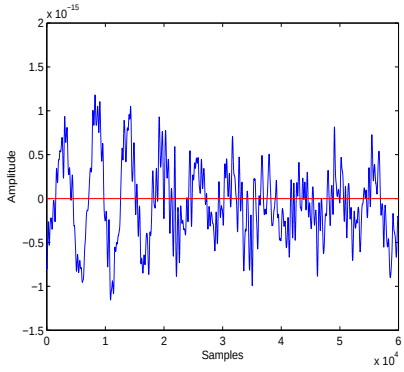


Figure 6.3: Comparison between source signals at SNR 10, in blue Virgo noise and in red chirp signal.

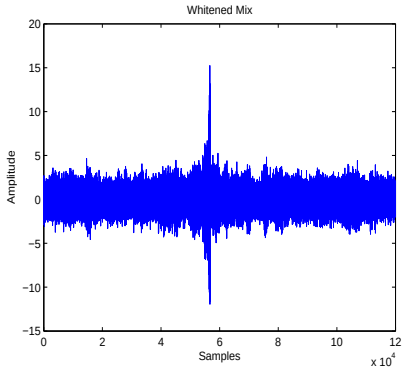
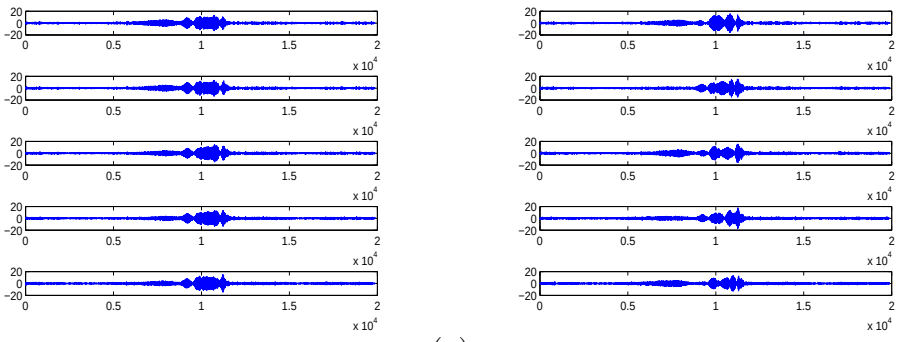


Figure 6.4: Whitened Mixture with SNR 10.



(a) (b)

Figure 6.5: Separated components from mixture at snr 10.

For better underline the results it is useful to show a spectrogram (time-frequency plot) of one of the separated components (the others have similar spectra). As it is possible to note from figure 6.6, it is clearly recognizable the chirp.

In the others simulation proposed, we show how it is still possible to obtain the

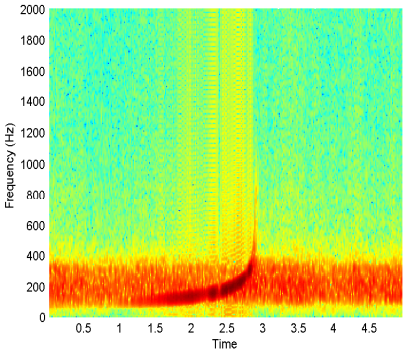


Figure 6.6: Spectro of the first independent component.

separation also if the SNR decrease.

Now, we consider a mixture composed of Virgo noise and a chirp signal with an SNR of 5. First of all, we show the mixture which we use in simulation (figure 6.7), then we show the source signals (figure 6.8), in order to underline the great difference in amplitude among the signal and the consequently difficult of the problem. As in the previous simulation described, it is important to stress that a SNR of 5 is really small, in fact if we compare the source noise with the mixture it is impossible to note where has been injected the signal. We can further observe in figure 6.9 this important feature, where we superimpose at the noise the chirp signal at SNR 5.

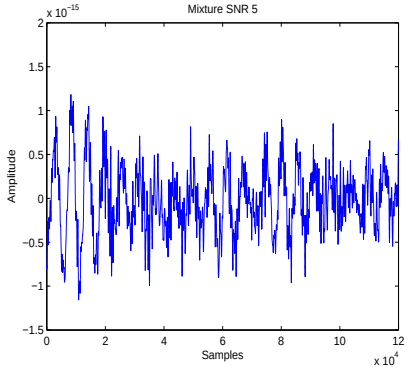


Figure 6.7: Mixture with SNR 5.

On the mixture, we apply the same process described for the first simulation: a first step of whitening of the signal in order to obtain the mixture in figure 6.10; and a second step in which we apply the NLPCA approach to separate the components. The result of this computation is shown in figures 6.11(a-b), as it is possible to note we clearly identify the gravitational signal and its position in the chunk of noise.

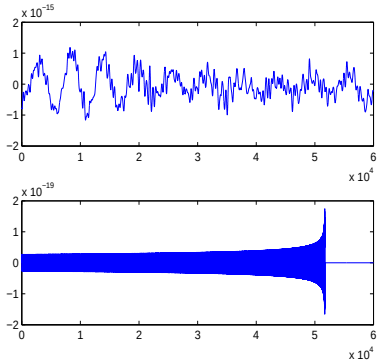


Figure 6.8: Source signal: (up) Virgo noise, (down) chirp signal.

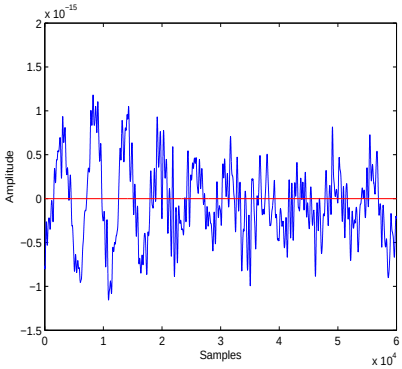


Figure 6.9: Comparison between source signals at SNR 5, in blue Virgo noise and in red chirp signal.

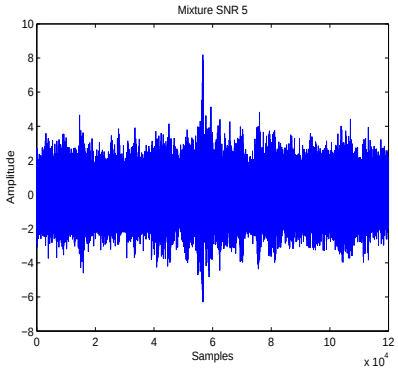
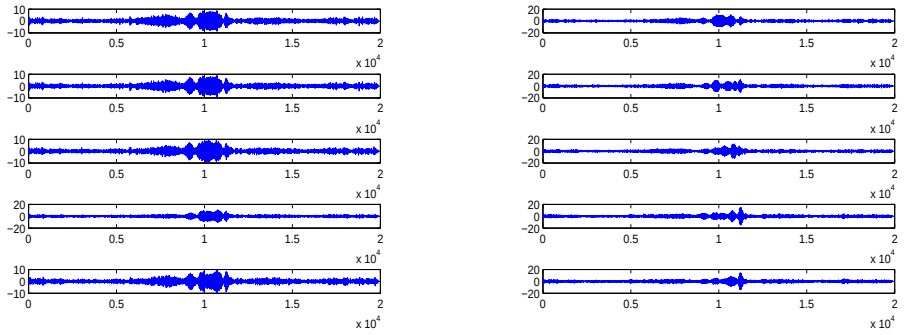


Figure 6.10: Whitened Mixture with SNR 5.

In figure 6.12, we show a spectrogram (time-frequency plot) of one of the separated components (the others have similar spectra), as it is possible to note from figure 6.12, it is clearly recognizable the chirp.

Continuing to decrease the SNR between the chirp signal and the noise to 1, we work



(a) (b)

Figure 6.11: Separated components from mixture at snr 5.

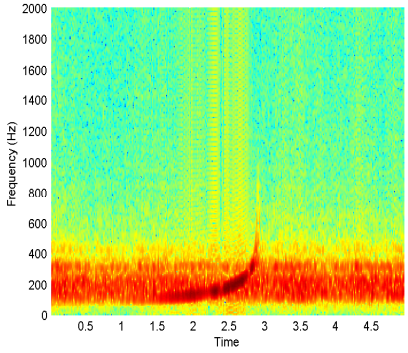


Figure 6.12: Spectro of the first independent component.

on the mixture shown in figure 6.13. For better understand the increasing difficulty of the problem, we show in figure 6.14 the source signals used to form the mixture and in figure 6.15 we show the superposition of the noise signal with the chirp signal with SNR 1. As it is possible to note from this images, the comparison between the noise and the signal really underline the problem of different amplitude and the difficult to detect the chirp signal in the noise.

On the mixture presented in figure 6.13, we apply first the whitening process, getting the mixture 6.16, and then we apply the NLPCA approach to separate the components 6.17(a-b).

In this last case, separation is not so good, in fact in time domain (see figure 6.17), it is impossible to recognize the chirp wave form, but if we have a look at the spectrogram (time-frequency plot) of one of the separated components in figure 6.18, it is still possible to note, also if not clearly as in the previous case, the chirp wave form.

From the simulation presented in this section, we can say that NLPCA is a good

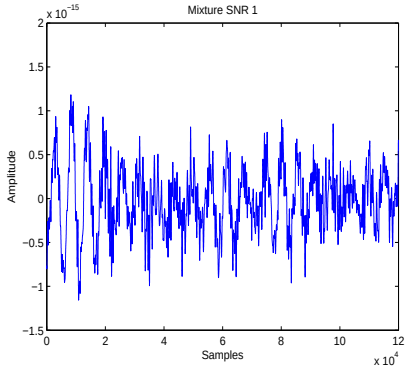


Figure 6.13: Mixture with SNR 1.

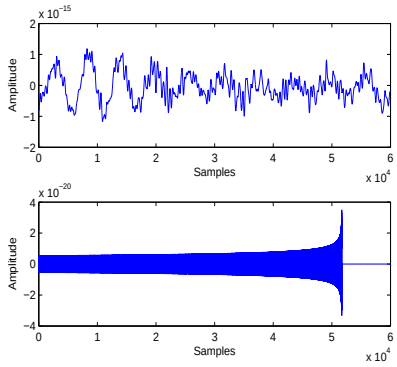


Figure 6.14: Source signal: (up) Virgo noise, (down) chirp signal.

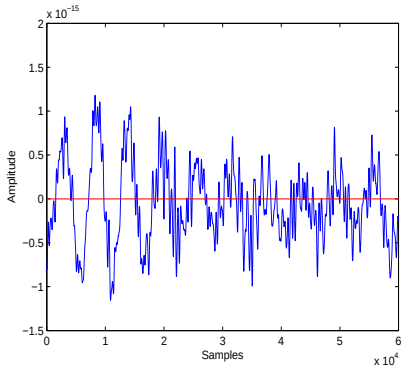


Figure 6.15: Comparison between source signals at SNR 1, in blue Virgo noise and in red chirp signal.

method for detecting a gravitational wave signal in the background noise of an interferometer. In this simulation, we have shown three particular case with different SNR, starting from 10 to 1. It is important to note that in gravitational wave detection an SNR of 10 is really a good starting point for detection. We can say that in this first

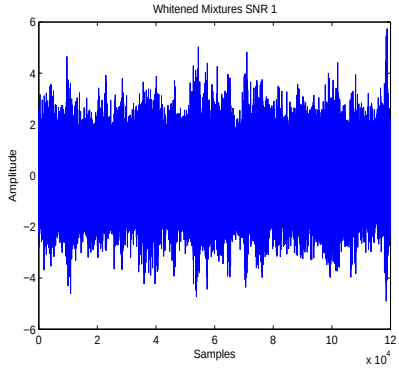
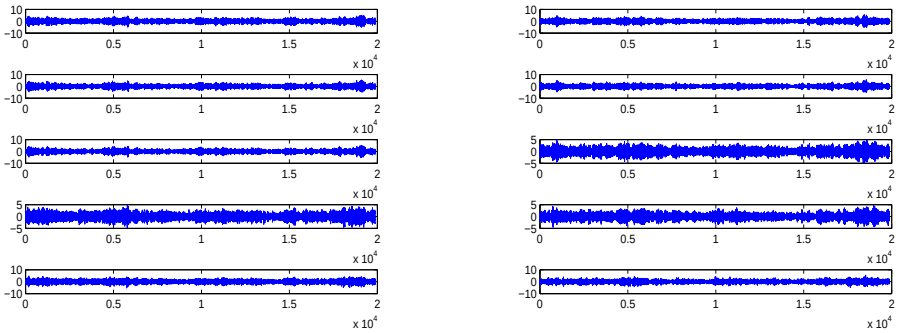


Figure 6.16: Whitened Mixture with SNR 1.



(a) (b)

Figure 6.17: Separated components from mixture at snr 1.

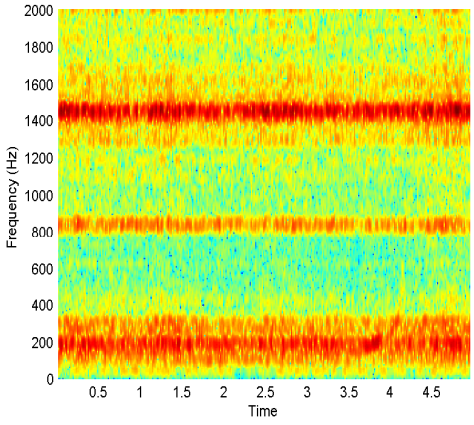


Figure 6.18: Spectro of the first independent component.

kind of application, we use our method for determine a chunk of data in which it is possible to find a gravitational signal. This is an important use of the method, because it is important to have a pre-analysis of the data. We must consider that the interferometer collects data all day long, so considering its sampling frequency for seconds, we

got an huge quantity of data. It's important to have a method for choose among this huge quantity a chunk of data in which it is possible to have a signal.

6.5 Chirp Wave Form Reconstruction

In this second kind of application, we can use our method for reconstructing the gravitational wave signal. Until now, we have used the NLPCA in according with a whitening process for detecting a chunk of data in which it is possible to have a chirp signal. Now considering an higher SNR, we can show how it is possible to use NLPCA for the reconstruction of the signal.
In this simulation we construct a mixture using noise coming from Virgo interferometer and an amplitude and frequency modulated chirp signal, the source signal are represented in figure 6.19. This modulated chirp signal is a variant to the standard chirp used in the previous section in which it is assumed that the generating mass have a spin, in this way to the characteristic frequency in time increasing of a chirp, we also have a difference in amplitude. We choose these two signals in order to get a signal similar to the one produced by coalescing binaries stars [67] and also for trying the method on a more difficult environment. In this simulation, we also show a comparison between the NLPCA method described in chapter 5 and the embedded FastICA method described at the end of chapter 4. We made this comparison, for evaluating two different method based on similar concept. Both the methods, in fact, work on a single mixture and use the embedding dimension as parameter of the method, but they are very different in the separation method and in the modelling of the neural network for the separation.
The mixture on which we work is represented in figure 6.20. First of all we analyze the embedding dimension of the mixture obtaining $\tau = 45$ and $m = 5$. These are the parameters that we use in both the approaches.

In figure 6.21(a,b), we show the results obtained from the two method: Embedded FastICA (figure a) and NLPCA (figure b). As it is possible to note as a first view on this two figure, both the methods obtain a good detection of the gravitational signal, but the NLPCA method can also reconstruct the signal waveform without any knowledge of the signal.
We also give a quantitative measure of the goodness of the separation using a correlation

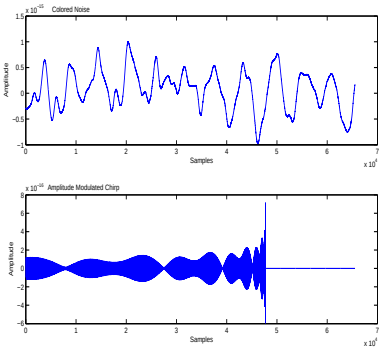


Figure 6.19: Source signals: Interferometric noise simulation (up); Amplitude modulate chirp signal (down).

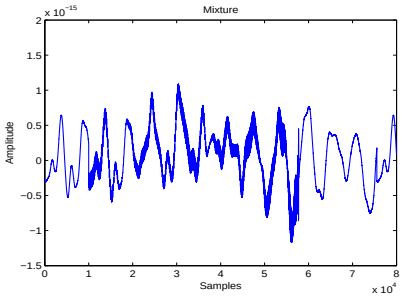
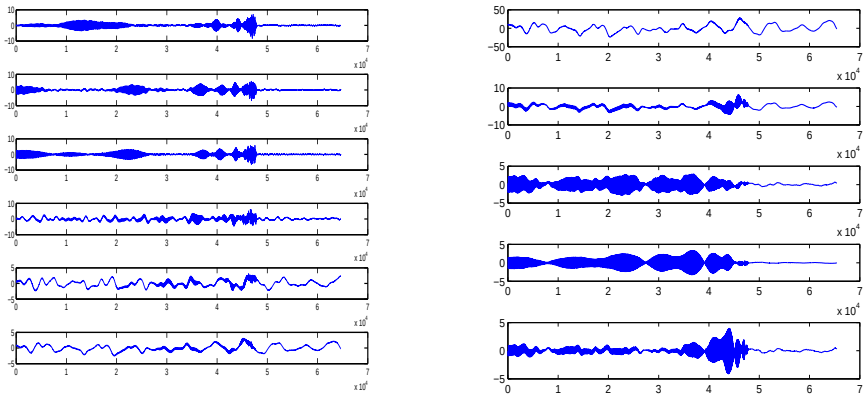


Figure 6.20: Signal Mixture.

measure, calculated between the extracted signals and the source signal. We use this kind of quality measure because we can't use the standard measures used in ICA world. The standard Amari's performance index, the measures usually used in literature [20], work on the separation matrix, but with these method we can't estimate that, so we need a measure that acts directly on the signals and its forms, avoiding to consider the amplitude: the correlation is a good candidate for that purpose.

So in the case of the simulation proposed, the correlation percentage for the chirp signal with the signals extracted by NLPCA is in mean of 70% while for the Embedded FastICA approach we have a mean of 43%.

Then, we choose the best representative signal for each method and we compare these with the source signal in figure 6.22). In this image, it is really evident the better performance of the NLPCA method in the reconstruction of the signal.



a) b)

Figure 6.21: Separated signals: a) FastICA based algorithm; b) Robust PCA based approach.

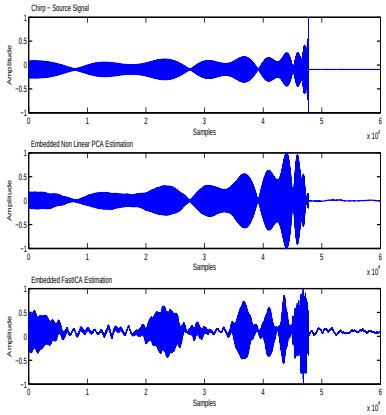


Figure 6.22: Comparison of the original modulated chirp signal (top) with the Embedded Non Linear PCA approach (middle) and FastICA approach (down).

6.6 Conclusions

In this chapter, we have shown an important field of application of the proposed approach. We want to stress that we are working on realistic data, the chunk of noise used in the simulation is really a small chunk of data taken from Virgo interferometer data. The gravitational source signals are the most realistic one, because they come from the theoretic study on this subject.

It is important to note that until now we don't have a sperimental proof of the existence of these waves and so we can only trust the gravitational wave theory for what regards the wave's form.

So, it is important to note that using a method that doesn't need to know in advantage the form of the signal as a target, is a real improvement in the gravitational wave detection theory. The simulations proposed in this chapter show an approach that permit to detect a gravitational wave signals without any knowledge about the signal itself. In fact, we use the source signals only for an evaluation of the performance of the methods.

We presented two kind of simulation: a first one with the only aim of detecting the signal and a second one with the purpose to detect and reconstruct the signal in its form.

In the simulations based on the detection of the signal, we got really good performance also at SNR really low and we want to stress without knowledge of the source.

It's also important to note that we can't make a comparison with the technique of the matched filter because of the intrinsic difference of the methods. In rough words, in the case of the matched filter, the signal is detected after a matching of the mixture data with a collection of possible target signal. This collection is composed of a forecasting of the wave form for the gravitational source signal, varying the mass and the position of the stars. But how can we be sure to have covered all the possible cases? And what happens if the signals emitted are not equal to the target signal? With matched filter, the answer to these question is that we cannot detect the signal or if you want that we have a very low probability to detect the signals.

In this chapter we have shown a method that overcome the knowledge of the wave form and so it can be used in every situation, maybe as a preprocessing analysis for individuating a chunk of data in which it is possible to have a gravitational signal.

The other kind of simulations proposed aims to detect and also to reconstruct the signal in its form. This is a very important field of application because it permits to recovery the signal in its form analyzing in detail its characteristics.

Chapter 7

Applications on Music Mixture

In the previous chapter, we have described a first kind of application of the proposed model on real data coming from Virgo interferometer. In this chapter, we show another important sector of application: music. We present several experiments and simulation about signals coming from music instruments.

7.1 Introduction

In these last years, music and computer science have met in a variety of way. The introduction of music in this field has opened really challenging scenarios for the researchers, in particular for the recognition of the speak, for the synthesization of digital music, for the creation of new algorithm of compression and so on. In this scenario, the recognition of a music instruments track from a mixture of different instruments tracks is an open problem with an high importance.
Until now, techniques like Independent Component Analysis have been used principally for the speak recognition task. This is because speak signals have an highly super Gaussian distribution. Several works have been proposed for this purpose.
In this chapter, we will show some applications of the NLPCA method described in algorithm 1 to the problem of separation of source signals from a single mixture in the case of music signal.
We start from simple simulation, in which given an instrument track playing a single note, we try the separation of the harmonics of the note using the proposed NLPCA method. After these first simulations, we pass to examine mixtures composed by dif-

ferents instruments. We try the separation of the single music sources composing the mixture. It's important to note that the quality of the obtained separation is so high that in certain situation we can extract from the estimated components the music transcription, in order to compare it with the one obtained from the original source.
As a major difficulty, we present a simulation in which we separate from a single mixture two different kind of signals: a music instruments track and a male voice, mixed with white noise.

7.2 Short introduction to Mathematical Armonies

Music is a periodic variation in air pressure

$$P = Asin(2\pi ft) \tag{7.1}$$

where A is the amplitude, t the time, f the frequency and P is the pressure in decibels or Pascal (see figure 7.1)

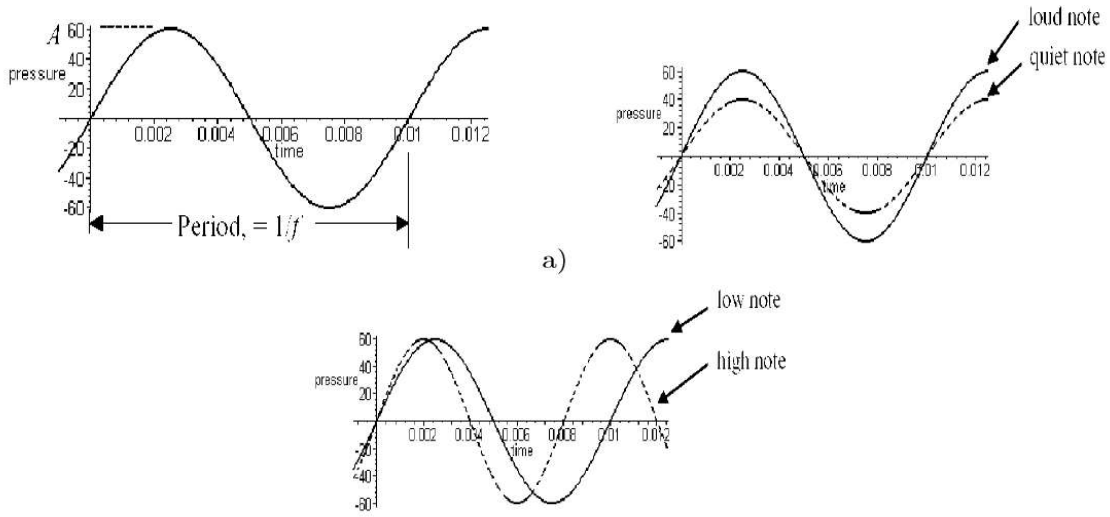


Figure 7.1: Sound Feature

Sound has two characteristics:

- *Volume*, that is the amplitude A in Pascals or decibels
- *Pitch*, that is the frequency f in Hertz (Hz)

In figure 7.2, we show some frequency range of various instruments. If we consider a

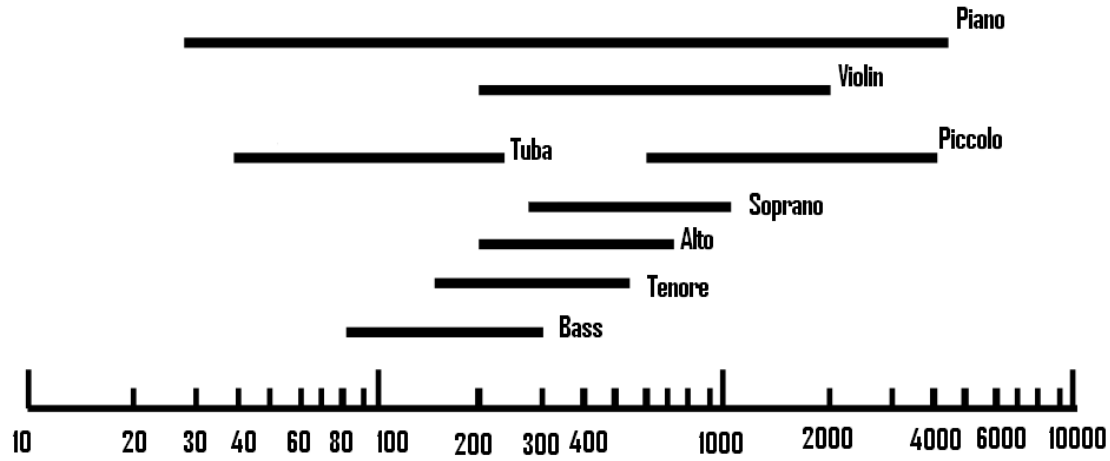


Figure 7.2: Frequency ranges of various instruments, in Hz. Audible frequencies range from 20 Hz to 20000 Hz

vibrating string, we can show that the frequency is expressed by

$$f = \frac{1}{2\,length}\sqrt{\frac{tension}{thickness}} \tag{7.2}$$

In this way, we say that the frequencies of octaves form a geometric sequence (figure 7.3). We note also that a string vibrates in many modes, called harmonics (figure 7.4)

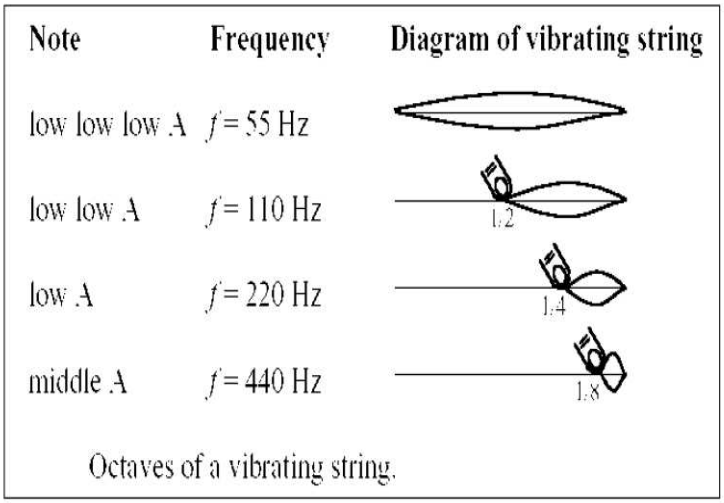


Figure 7.3: Frequency diagram of octaves.

and the frequencies of the harmonics form an arithmetic sequence.

In figure 7.5, we show an example of a keyboard. There are two accepted musical

Note	Frequency	Harmonic	Diagram of string
low low low A	$f = 55$ Hz	fundamental	
low low A	$f = 110$ Hz	second	
low E	$f = 165$ Hz	third	
low A	$f = 220$ Hz	fourth	
middle C#	$f = 275$ Hz	fifth	
middle E	$f = 330$ Hz	sixth	
approx. middle G	$f = 385$ Hz	seventh	
middle A	$f = 440$ Hz	eighth	

Figure 7.4: Frequency diagram of music harmonics.

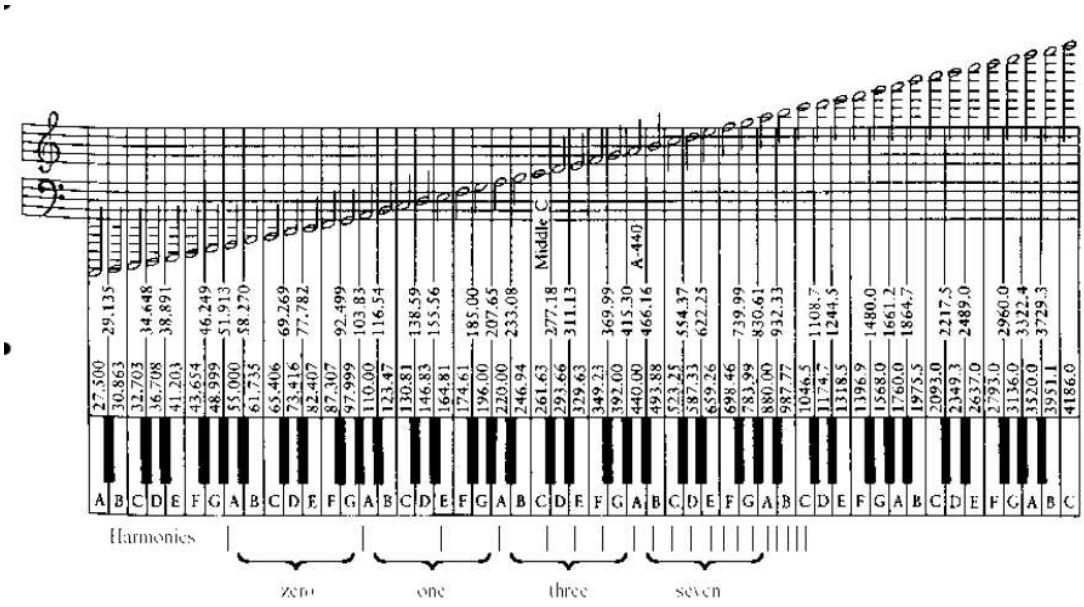


Figure 7.5: Examples of harmonics and octave in the case of a piano.

pitch standards, the so-called American Standard pitch, which takes A in the fourth piano octave (A4) to have a frequency of 440 Hz, and the older International pitch standard, which takes A4 to have a frequency of 435 Hz. Both of these pitch standards define what are called "equal tempered chromatic scales". Mathematically, this means that each successive pitch is related to the previous by a factor of the twelfth root of 3.

$$\sqrt[12]{2} = 1.05946309436 \tag{7.3}$$

That is, the ratio between the frequencies of any two successive pitches in either standard is 1.05946309436. There are twelve half-tones (black and white keys on a piano), or steps in an octave. Since the pitch (frequency) of each successive step is related to the previous pitch by the twelfth root of 2, the twelfth step above a given pitch is exactly twice the initial pitch (i.e., an octave corresponds to a doubling of a pitch). The frequency of intermediate notes, or pitches, can be found simply by multiplying (or dividing) a given starting pitch by as many factors of the twelfth root of 2 as there are steps up to (or down to) the desired pitch. For example, the G above A4 (that is G5) in the American Standard has a frequency of $440 \times (\sqrt[12]{2})^1 = 440 \times 1.78179743628 = 783.99Hz$ (approximately). Likewise in the International standard, G5 has a frequency of 775.08 Hz(approximately). G#5 is another factor of the 12th root of 2 above these, or 830.61 and 821.17 Hz, respectively.
Note when counting steps that there is a single half-tone (step) between B and C, and between E and F. In figure 7.6, 7.7 and 7.8, we show some fundamental frequencies.
The frequencies of 440 Hz of the note LA corresponds to the fundamental frequency and it is associated to the diapason. The notes of the superior tone are multiple of the fundamental frequency. For example, we consider the note La with a fundamental frequency of 55 Hz, this note has the following harmonics:

- I harmonic: $f = 55 * 2 = 110Hz$
- II harmonic: $f = 55 * 4 = 220Hz$
- III harmonic: $f = 55 * 8 = 440Hz$
- IV harmonic: $f = 55 * 16 = 880Hz$
- V harmonic: $f = 55 * 32 = 1760Hz$

In general, n harmonic: $f = c * 2^n$.

7.3 Simulation on the separation of harmonics

The first kind of simulation on music data made is on the separation of harmonics: given a single mixture of a music instrument sounding a note, we try to separate the different harmonics of this note. We made several experiments on different kind of

Note	Symbol	Frequency
DO2	C2	66 Hz
DO♯2	C♯2	70 Hz
RE2	D2	74 Hz
RE♯2	D♯2	78 Hz
MI2	E2	83 Hz
FA2	F2	88 Hz
FA♯2	F♯2	93 Hz
SOL2	G2	98 Hz
SOL♯2	G♯2	104 Hz
LA2	A2	110 Hz
LA♯2	A♯2	117 Hz
SI2	B2	124 Hz
DO3	C3	131 Hz
DO♯3	C♯3	139 Hz
RE3	D3	147 Hz
RE♯3	D♯3	156 Hz
MI3	E3	165 Hz
FA3	F3	175 Hz
FA♯3	F♯3	185 Hz
SOL3	G3	196 Hz
SOL♯3	G♯3	208 Hz
LA3	A3	220 Hz
LA♯3	A♯3	233 Hz
SI3	B3	247 Hz

Figure 7.6: Examples of frequencies of the notes: table 1.

music instruments and different notes. From the results obtained, we can state that with the proposed approach it is possible to separate the harmonics. Let us show

DO4	C4	262 Hz
DO♯4	C♯4	277 Hz
RE4	D4	294 Hz
RE♯4	D♯4	311 Hz
MI4	E4	330 Hz
FA4	F4	349 Hz
FA♯4	F♯4	370 Hz
SOL4	G4	392 Hz
SOL♯4	G♯4	415 Hz
LA4	A4	440 Hz
LA♯4	A♯4	466 Hz
SI4	B4	494 Hz
DO5	C5	523 Hz
DO♯5	C♯5	554 Hz
RE5	D5	587 Hz
RE♯5	D♯5	622 Hz
MI5	E5	659 Hz
FA5	F5	698 Hz
FA♯5	F♯5	740 Hz
SOL5	G5	784 Hz
SOL♯5	G♯5	831 Hz
LA5	A5	880 Hz
LA♯5	A♯5	932 Hz
SI5	B5	988 Hz
DO6	C6	1046 Hz

Figure 7.7: Examples of frequencies of the notes: table 2.

some figures explaining the obtained results. The first set of figures is relative to some experiments made on a single mixture of a flute sounding C4 note. In figure 7.9, we

DO♯6	C♯6	1109 Hz
RE6	D6	1175 Hz
RE♯6	D♯6	1245 Hz
MI6	E6	1319 Hz
FA6	F6	1397 Hz
FA♯6	F♯6	1480 Hz
SOL6	G6	1568 Hz
SOL♯6	G♯6	1661 Hz
LA6	A6	1760 Hz
LA♯6	A♯6	1865 Hz
SI6	B6	1976 Hz
DO7	C7	2093 Hz
DO♯7	C♯7	2217 Hz
RE7	D7	2349 Hz
RE♯7	D♯7	2489 Hz
MI7	E7	2637 Hz
FA7	F7	2794 Hz
FA♯7	F♯7	2960 Hz
SOL7	G7	3136 Hz
SOL♯7	G♯7	3322 Hz
LA7	A7	3520 Hz
LA♯7	A♯ 7	3729 Hz
SI7	B7	3951 Hz
DO8	C8	4186 Hz

Figure 7.8: Examples of frequencies of the notes: table 3.

show the source signal of the mixture used and in figure 7.10 the spectrogram of this signal in order to underline the frequency of the note and its harmonics.

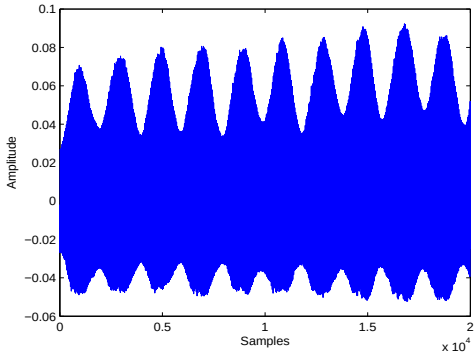


Figure 7.9: Harmonics separation on flute C4 note: source signal.

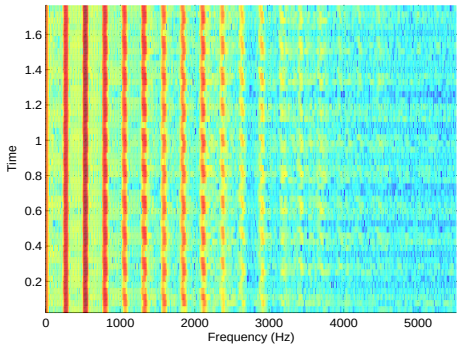


Figure 7.10: Harmonics separation on flute C4 note: spectrogram of the source signal.

The results of the separation of the harmonics are visible in the time domain (see figure 7.11), in the frequency domain (see figure 7.12) and in the frequency-time domain (see figure 7.13).
As it is possible to note from these figure, we got a good separation of the different harmonics starting from a single mixture of the original signal.

The second set of figures is relative to some experiments made on a single mixture of a piano sounding G6 note. As before the first figure presented (7.14) represents the source signal, while in figure 7.15, we show the spectrogram of this signal for better evidentiate the time-frequency contribute of the note and its harmonics.

After the application of the NLPCA method, we can see the results of the separation in the time domain (see figure 7.16), in the frequency domain (see figure 7.17) and in the frequency-time domain (see figure 7.18).
As it is possible to note from these figure, the separation of the harmonics is really good. In fact we can distinguish clearly the contribute of each harmonic to the single

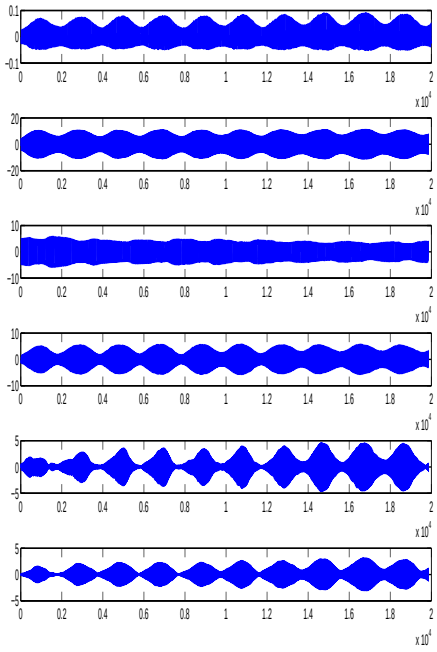


Figure 7.11: Harmonics separation on flute C4 note: separation view in time domain. The first plot from above is the source signal, while the other are the separation of the harmonics.

estimated component.

As third example of separation of harmonics, let us consider a trumpet playing the C4 note. For better understand the results obtained, first of all we present the source signal in figure 7.19 and its spectrogram in figure 7.20.

We show the results in different context: in the time domain (see figure 7.21), in the frequency domain (see figure 7.22) and in the frequency-time domain (see figure 7.23).

Once again, we can note a good separation of the harmonics.

As last simulation for this section we consider a violin playing C5 note. As in the previous case, in figure 7.24, we show the source signal of the mixture used and in figure 7.25 the spectrogram of this signal in order to underline the frequency of the note and its harmonics.

We show the results in three different context: in the time domain (see figure 7.26), in the frequency domain (see figure 7.27) and in the frequency-time domain (see figure 7.28). These different representations are useful for better understand the performance

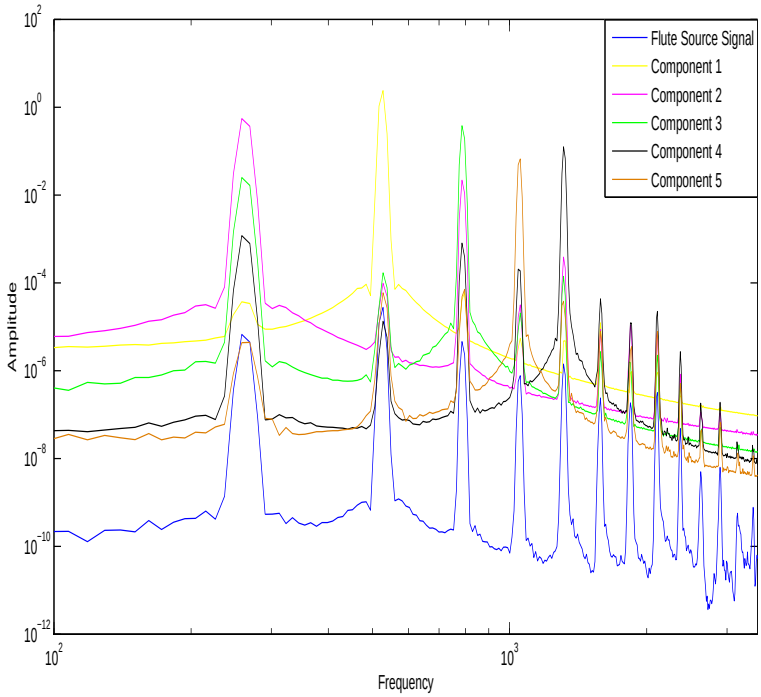


Figure 7.12: Harmonics separation on flute C4 note: separation view in frequency domain. The blue curve is the source signal, while the other are the separation of the harmonics.

of the separation. In fact, the frequency domain and the time-frequency domain are more representetive for these results. Analyzing these figures, we stress the high performance in the separation obtained by the NLPCA method.

As it is possible to note from the simulations proposed in this section, the NLPCA introduced in this work is a very powerful method for the separation of the harmonics from single note. This is a very interesting result also because as the variety of the simulations can show, it doesn't depend from the type of the instruments used or from the note played.
We must stress that we made several experiments on that topic varying instruments and note and in all the case we got a good separation. Here we have presented only the most representative ones.

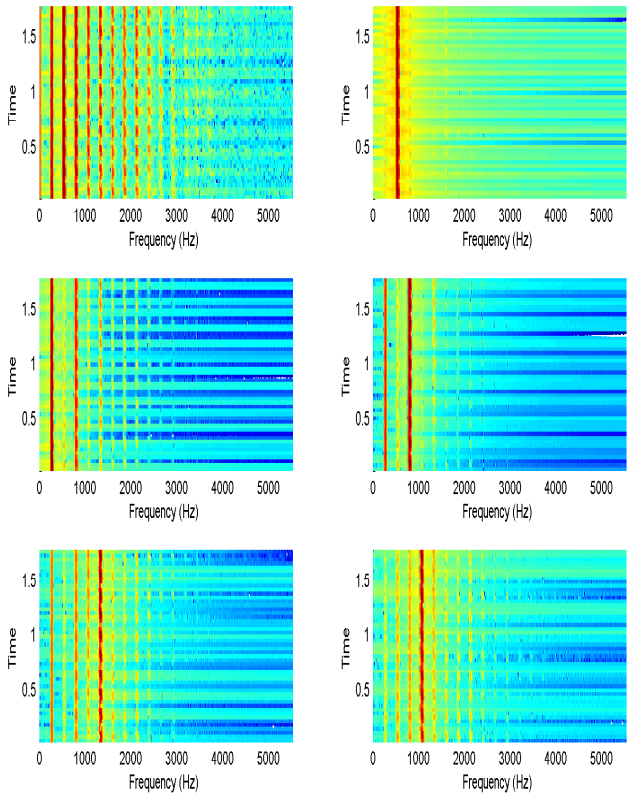


Figure 7.13: Harmonics separation on flute C4 note: separation view in frequency-time domain. The first plot in the left corner is the source signal, while the other are the separation of the harmonics.

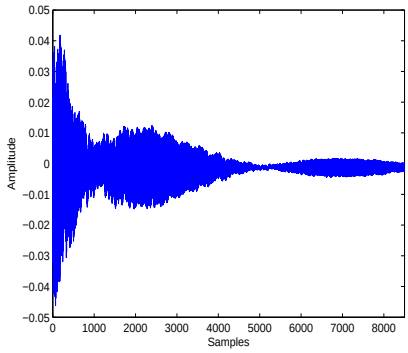


Figure 7.14: Harmonics separation on piano G6 note: source signal.

7.4 Experimental results

In the second part of simulations, we focused our attention on the separation of music signals and we made several experiments using single mixtures composed by three

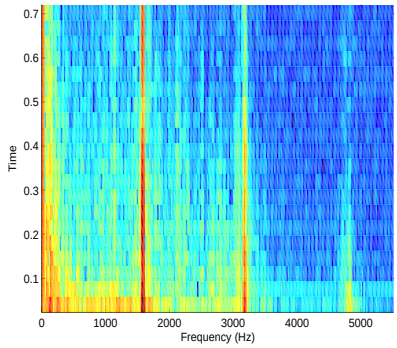


Figure 7.15: Harmonics separation on piano G6 note: spectrogram of the source signal.

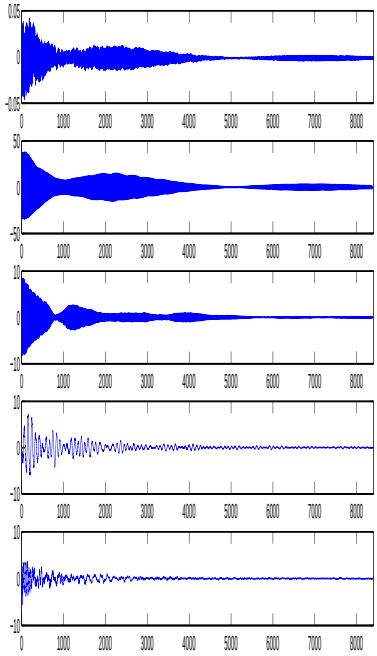


Figure 7.16: Harmonics separation on piano G6 note: separation view in time domain. The first plot from above is the source signal, while the other are the separation of the harmonics.

different kind of musical instruments. The samples are chosen among the following musical instruments: cello, viola, piano, guitar, oboe, gong, violin, castanets, xylophone, etc.

We use known signals, for better understanding the quality of the results, because we can compare the estimated signals with the source signals. We compare our model with the one based on that described in Section 4.9, also using a performance index

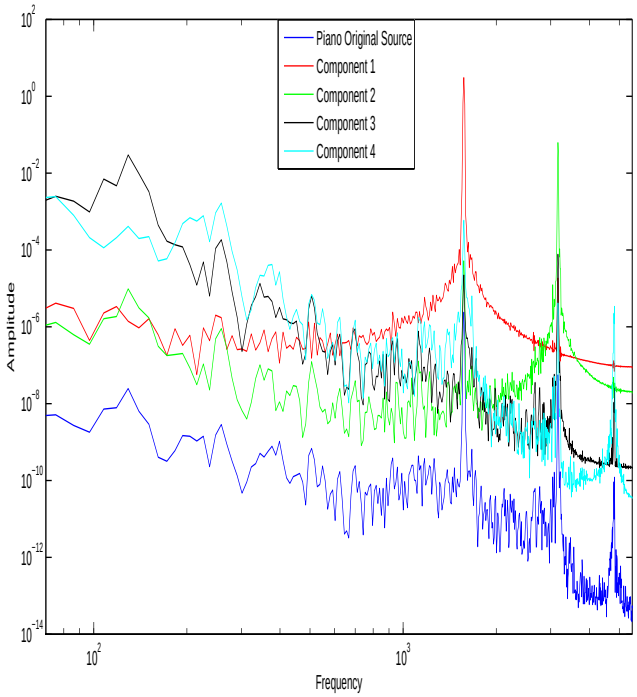


Figure 7.17: Harmonics separation on piano G6 note note: separation view in frequency domain. The blue curve is the source signal, while the other are the separation of the harmonics.

based on the correlation. We note that in our case we are unable to use the standard Amari's performance index since in that case a demixing matrix is needed [20]. We stress that in all the experiments that we made, we obtained a good separation of the single signals and this is also confirmed by the high correlation between the estimated and the source signals, that generally is from 50% to 94%.

In the first experiment we present the result obtained by analyzing a mixture composed by these instruments: oboe, cello and gong. With our approach, we obtain a good separation of the single signals of the mixture, with a correlation of 94% for the oboe, 85% for the cello and 50% for the gong, while with the ICA approach we got a correlation of 75% for the oboe, 70% for the cello and 45% for the gong, respectively.

To clarify the separation performances, in figure 7.29, we also show the single mixture on which we applied the proposed approach and in figure 7.30 (a,b,c), we show the original source signal (top), the NLPCA approach signal estimation (middle) and the Embedded FastICA Approach Estimation (down) for the three source signal, respectively.

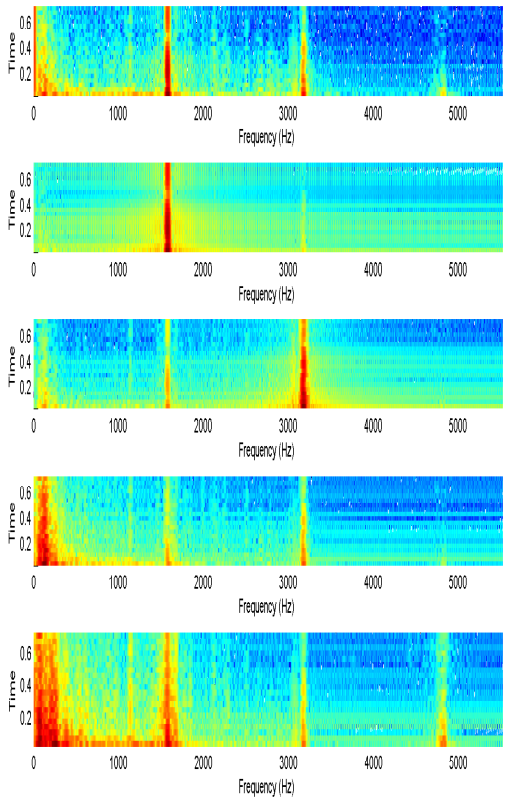


Figure 7.18: Harmonics separation on piano G6 note: separation view in frequency-time domain. The first plot in the left corner is the source signal, while the other are the separation of the harmonics.

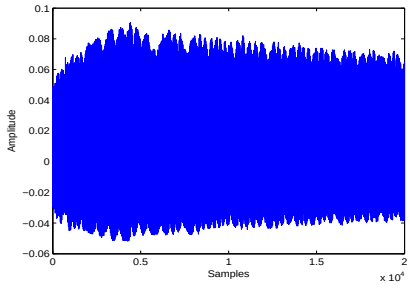


Figure 7.19: Harmonics separation on trumpet C4 note: source signal.

As a second experiment presented, we work on a mixture composed by these instruments: castanets, xylophone and viola. With our approach, we obtain the separation of the single signals from the mixture, with a correlation of 50% for the castanets, 83% for the viola and 50% for the xylophone, while with the Embedded FastICA approach

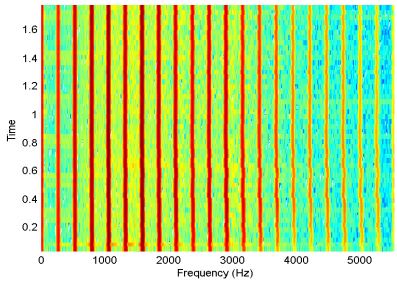


Figure 7.20: Harmonics separation on trumpet C4 note: spectrogram of the source signal.

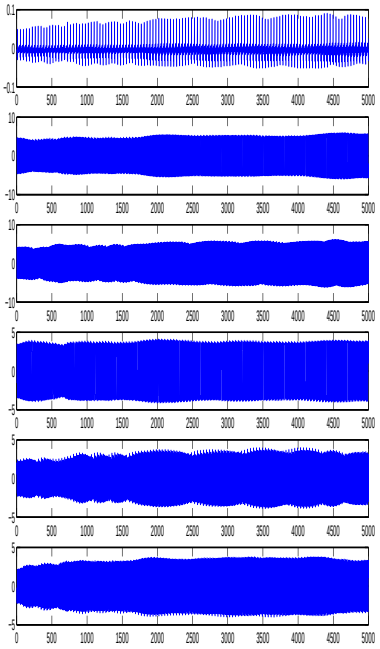


Figure 7.21: Harmonics separation on trumpet C4 note: separation view in time domain. The first plot from above is the source signal, while the other are the separation of the harmonics.

we got a correlation of 20% for the castanets, 51% for the viola and 15% for the xylophone.

In figure 7.31, we show the single mixture on which we applied the proposed approach, in figure 7.32 (a,b,c), we show the original source signal (top), the NLPCA approach signal estimation (middle) and the Embedded FastICA Approach Estimation (down) for the three source signal.

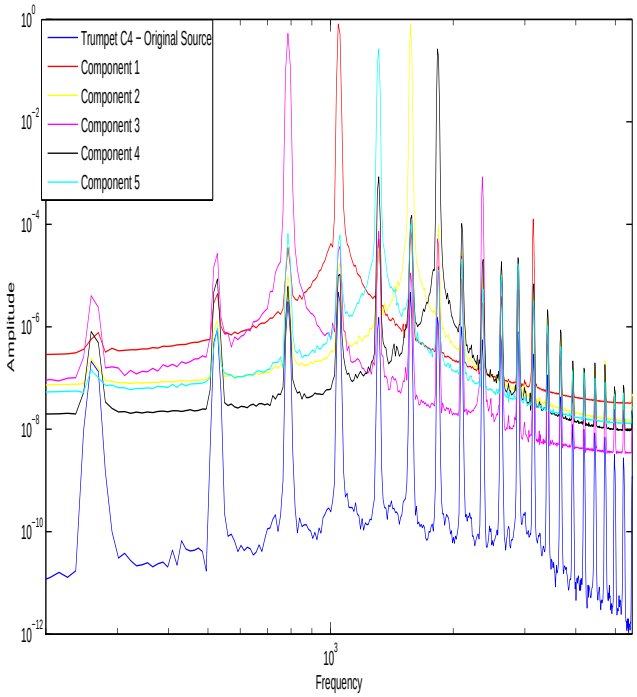


Figure 7.22: Harmonics separation on trumpet C4 note note: separation view in frequency domain. The blue curve is the source signal, while the other are the separation of the harmonics.

As it is possible to note an important feature of the simulation made is that we use different kind of instruments in the composition of the mixture. In the third experiment proposed in fact, we analyze a mixture composed by these instruments: castanets, bells and viola.

The obtained results are: a correlation of 50% for the castanets, 83% for the viola and 55% for the bells with the proposed approach, while with the Embedded FastICA approach we have a correlation of 5% for the castanets, 62% for the viola and 18% for the bells.

For better understand the results in figure 7.34 (a,b,c), we show the original source signal (top), the NLPCA approach signal estimation (middle) and the Embedded FastICA Approach Estimation (down) for the three source signal.

As a fourth experiment, we present the result obtained by analyzing a mixture composed by these instruments: guitar, oboe and viola. By using the NLPCA approach we have a correlation of 50% for the guitar, 80% for the viola and 85% for the oboe, while with the Embedded FastICA approach we get a correlation of 32% for the guitar,

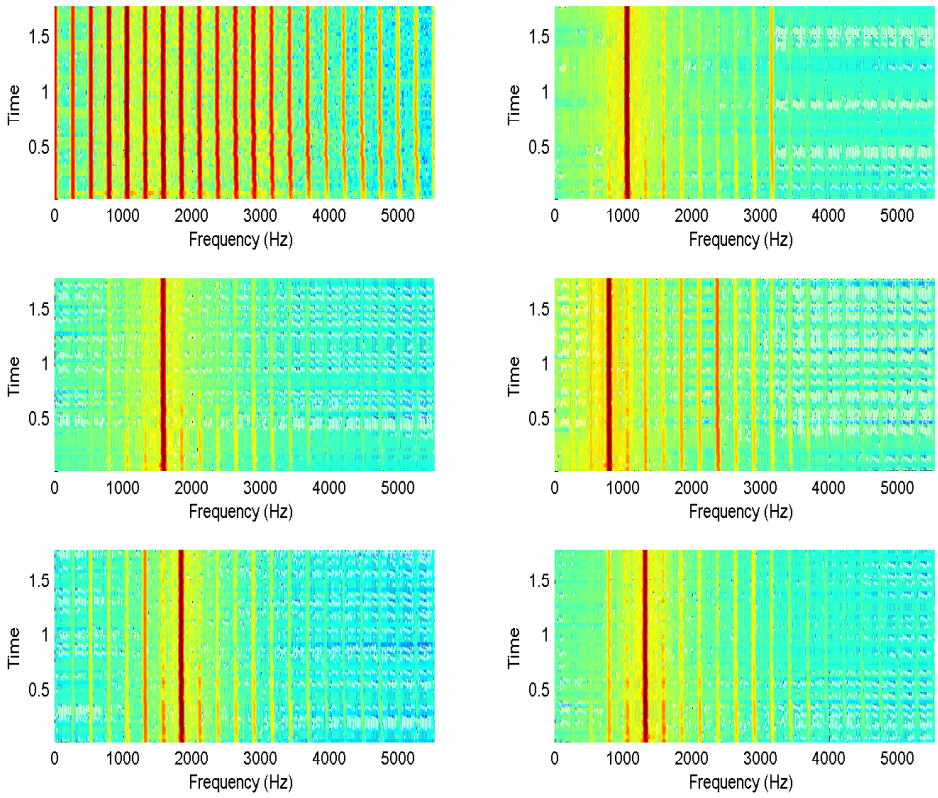


Figure 7.23: Harmonics separation on trumpet C4 note: separation view in frequency-time domain. The first plot in the left corner is the source signal, while the other are the separation of the harmonics.

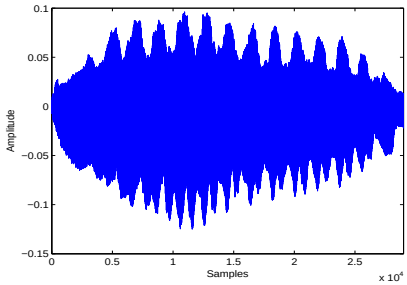


Figure 7.24: Harmonics separation on violin C5 note: source signal.

66% for the viola and 75% for the oboe.

Also in this case to clarify the result in figure 7.35, we show the single mixture on which we applied the proposed approach, in figure 7.36 (a,b,c), we show the original source signal (top), the NLPCA approach signal estimation (middle) and the Embedded

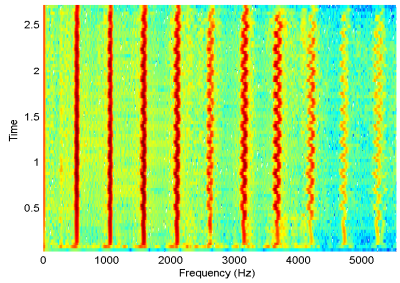


Figure 7.25: Harmonics separation on violin C5 note: spectrogram of the source signal.

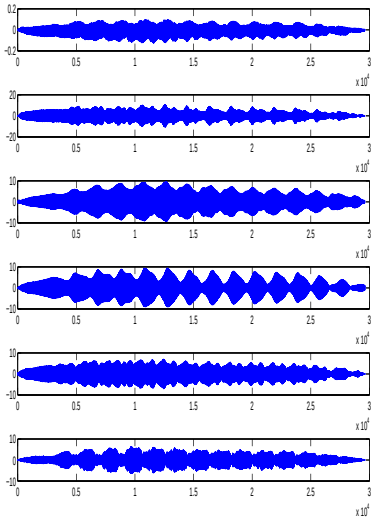


Figure 7.26: Harmonics separation on violin C5 note: separation view in time domain. The first plot from above is the source signal, while the other are the separation of the harmonics.

FastICA approach estimation (down) for the three source signal.

Going on with the differentiation of the instruments, we present the case in which the mixture is composed by: oboe, bell and corn. With the approach proposed in this work, we obtain a good separation of the single signals of the mixture, with a correlation of 50% for the corn, 90% for the oboe and 55% for the bell, while with the Embedded FastICA approach we got a correlation of 40% for the corn, 62% for the oboe and 30% for the bell.

In figure 7.37, we show the single mixture on which we applied the proposed approach, while in figure 7.38 (a,b,c), we show a comparison in the time domain among the original source signal (top), the NLPCA approach signal estimation (middle) and the

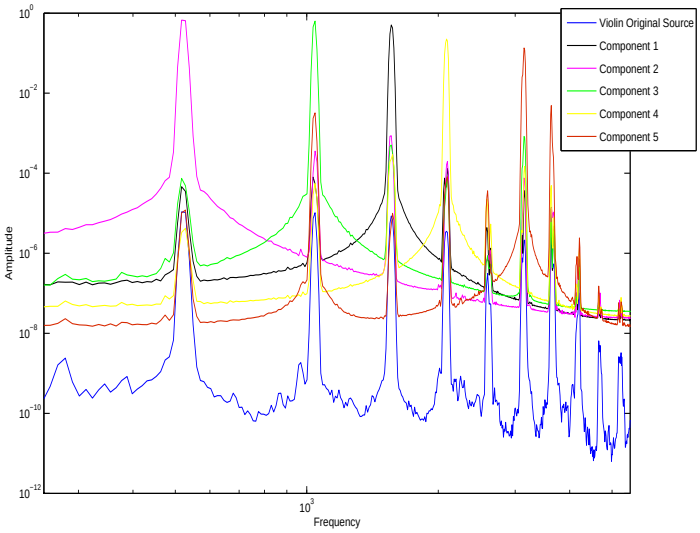


Figure 7.27: Harmonics separation on violin C5 note note: separation view in frequency domain. The blue curve is the source signal, while the other are the separation of the harmonics.

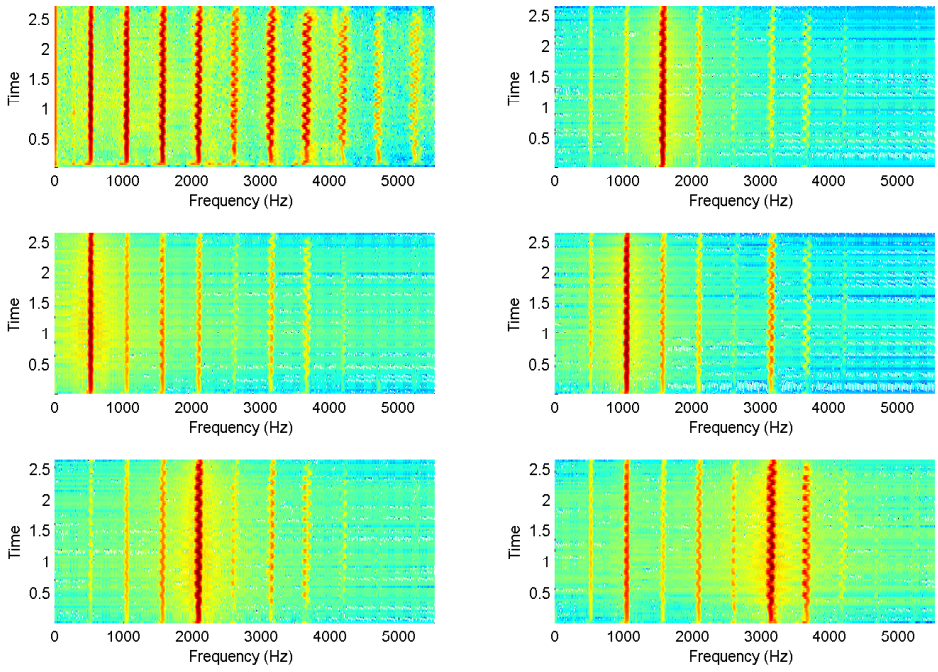


Figure 7.28: Harmonics separation on violin C5 note: separation view in frequency-time domain. The first plot in the left corner is the source signal, while the other are the separation of the harmonics.

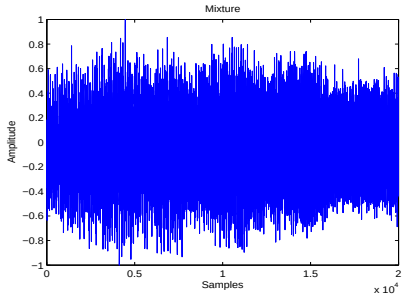


Figure 7.29: The single music mixture in simulation 1.

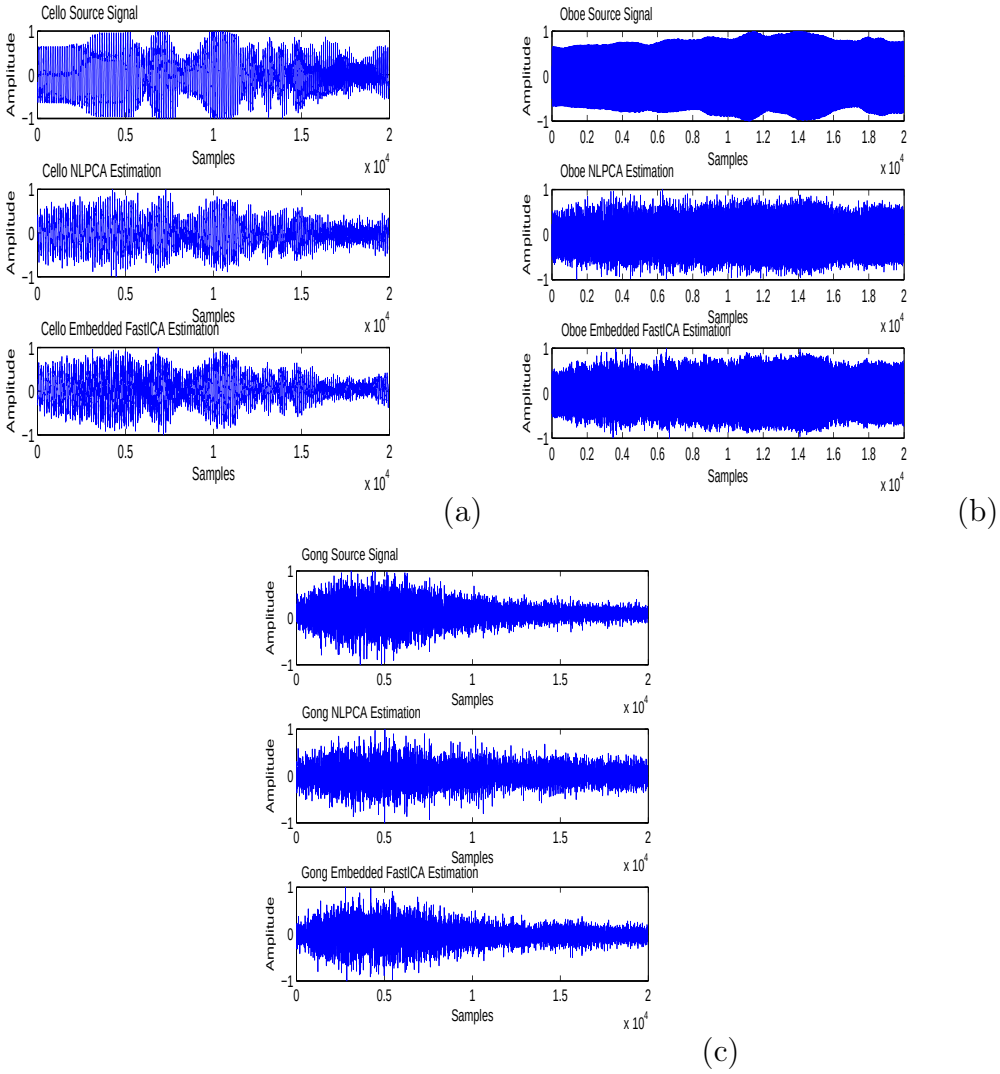


Figure 7.30: The comparison among original source signal (top), NLPCA Estimation (middle), Embedded FastICA Estimation (down): (a) cello signal, (b) oboe signal,(c) gong signal.

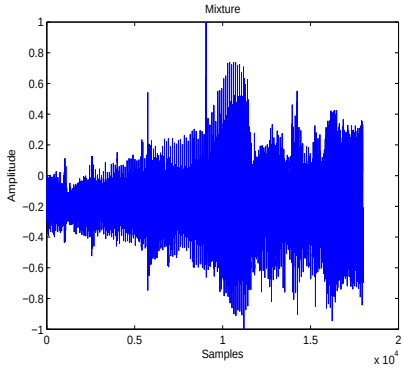


Figure 7.31: The single mixture in experiment 2.

Embedded FastICA Approach Estimation (down) for the three source signal.

As a summary of results, we report a table where we indicate the correlation coefficient in the experiments proposed by the NLPCA approach and the FastICA approach.

We also stress that another important result is the extraction of the single instrument's score and its musical transcription from the separated signals. Even thought we are still working on this problem, here we present some results where we obtain a better performance. In fact, for example considering the signals of experiment 1, in Fig. 7.39 (a) we compare the original cello score (up), with the cello score extracted by the separated signal (down) and in Fig. 7.39 (b), we compare the original oboe score (up), with the oboe score extracted by the separated signal (down). We observe that in both the cases there is a good agreement between the scores.
We can conclude that with our method we can perform an high quality separation of music signals from a single mixture and that by using the separated signals we can transcript in a simple way instrument scores.

7.5 A different kind of experiment: separation of a voice from a music instruments

In this section, we describe a different kind of simulation in which we consider signals of different nature. In particular in this experiment, we consider a mixture of two recorded signals and one Gaussian noise (Fig. 7.40 - down). The first recorded signal is the recording of a male voice that contains the seven digits (7.40 - top) and the second is a single flute note (G6) (7.40 - middle). The mixture that we analyze is

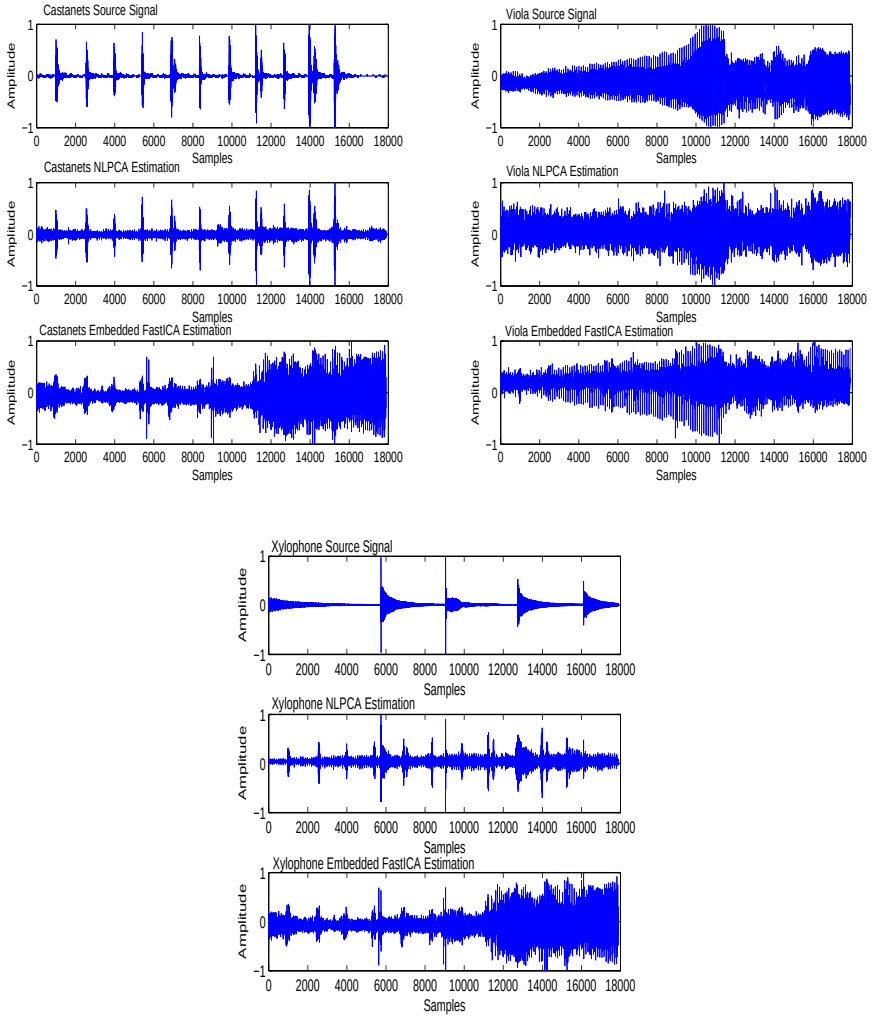


Figure 7.32: The comparison among original source signal (top), NLPCA Estimation (middle), Embedded FastICA Estimation (down): (a) castanets signal, (b) viola signal,(c) xylophone signal.

plotted in Fig. 7.41. We note that in this case, for the flute note, we have the time lag $\tau = 2$ and the embedding dimension $m = 9$. Instead for the male voice is $\tau = 4$ and $m = 13$. Applying the phase reconstruct approach on the mixture we obtain $\tau = 2$ and $m = 10$. In Fig. 7.43, we show the separated signals obtained by using NLPCA, the proposed approach, and we compare these results with the approach proposed in [42] in Fig. 7.42. The correlation percentages are 98% for the flute source and 62% for the male voice in the case of Robust NLPCA approach; 94% and 58% in the case of FastICA approach. However, it's possible to note in Fig. 7.45 and in Fig. 7.44 that using the robust PCA NN, we obtain a clearer separation, that can be appreciate also

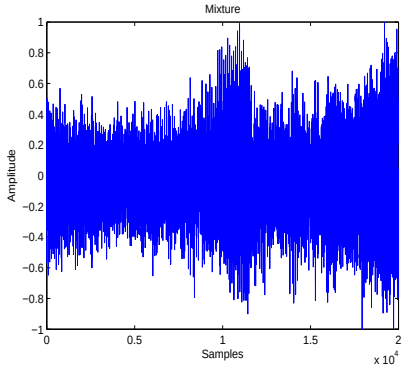


Figure 7.33: The single mixture in experiment 3.

listening the results.

7.6 Conclusions

In this section, we used a methodology to accomplish single channel mixtures BSS. The proposed approach is based on an on-line Robust PCA NN and the embedding dimension and the time lag are used to define the architecture of the NN. We also compared the method with one based on a batch ICA approach. From the experiments that we have made, we found that the robust PCA NN permits to good results compared with those of the other approach. We also can stress that one of the features of the on-line learning is that it permits to define the NN's input dimension that improves the separation of the signals.

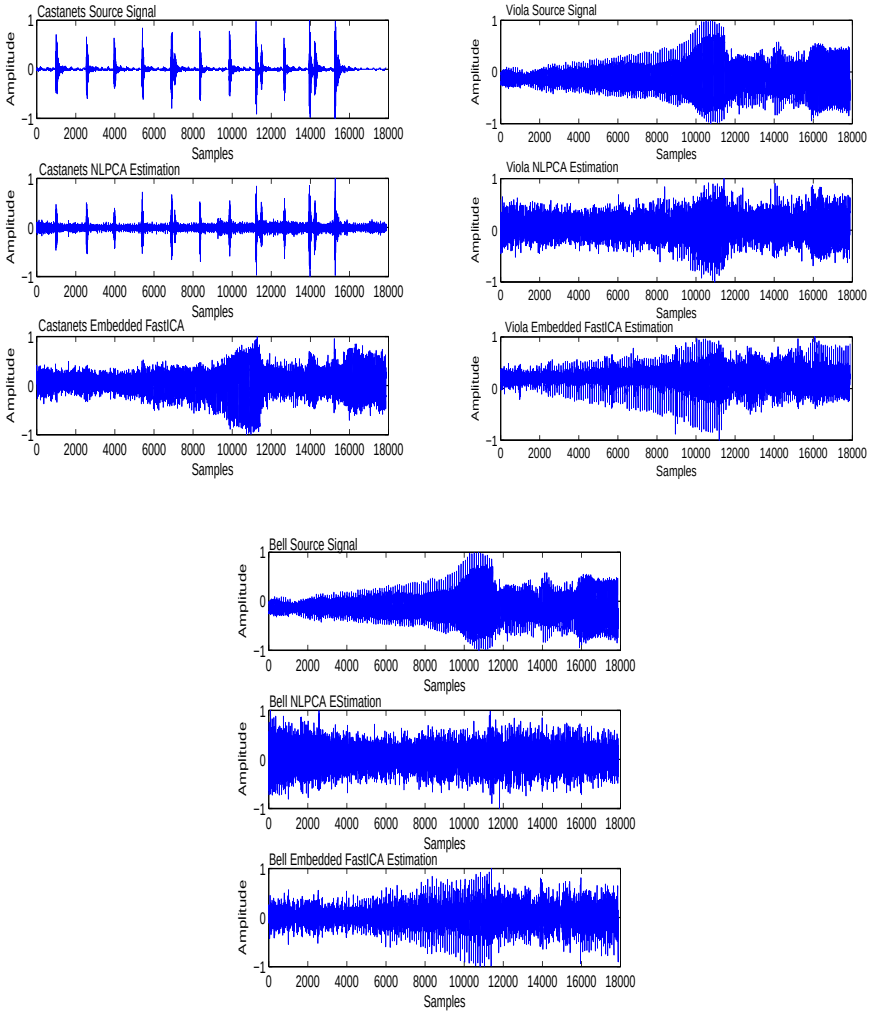


Figure 7.34: The comparison among original source signal (top), NLPCA Estimation (middle), Embedded FastICA Estimation (down): (a) castanets signal, (b) viola signal,(c) bells signal.

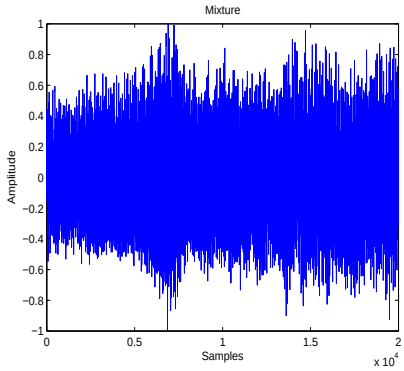


Figure 7.35: The single mixture in experiment 4.

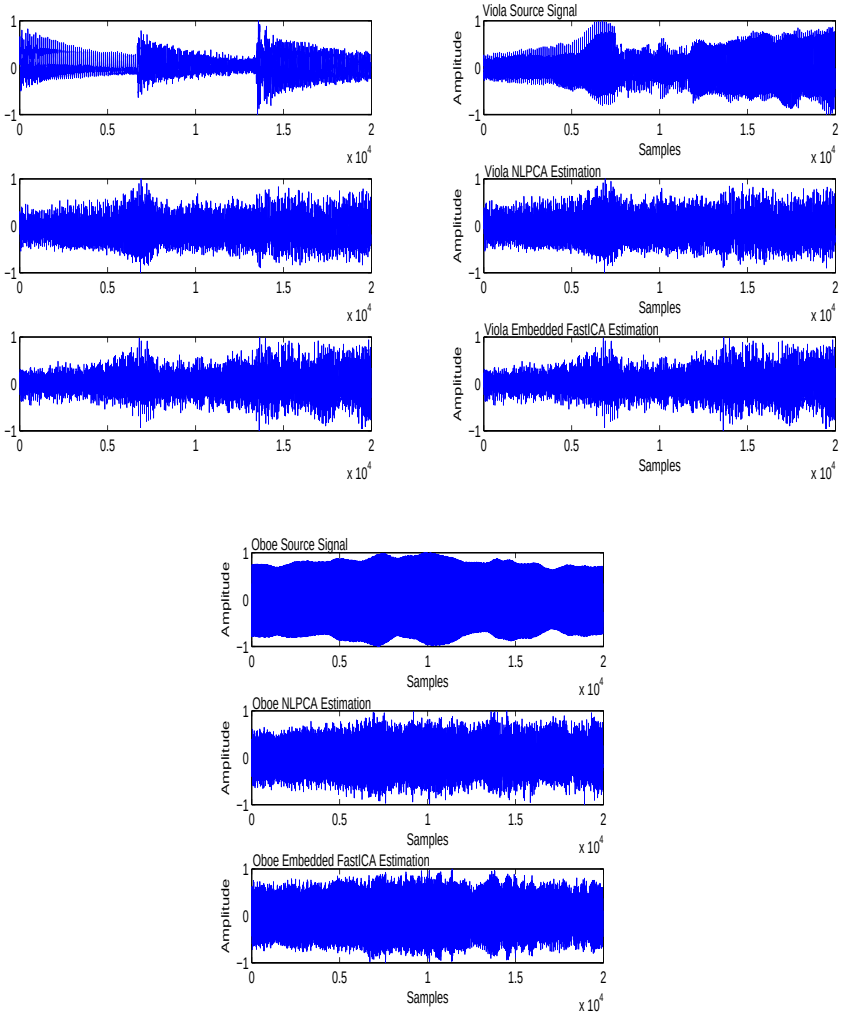


Figure 7.36: The comparison among original source signal (top), NLPCA Estimation (middle), Embedded FastICA Estimation (down): (a) guitar signal, (b) viola signal,(c) oboe signal.

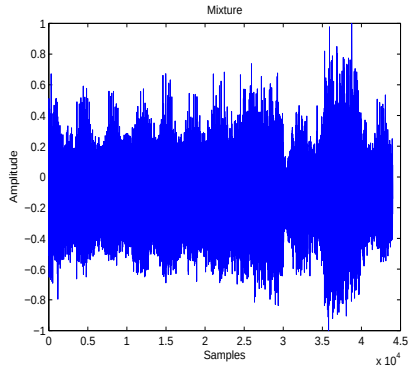


Figure 7.37: The single mixture in experiment 5.

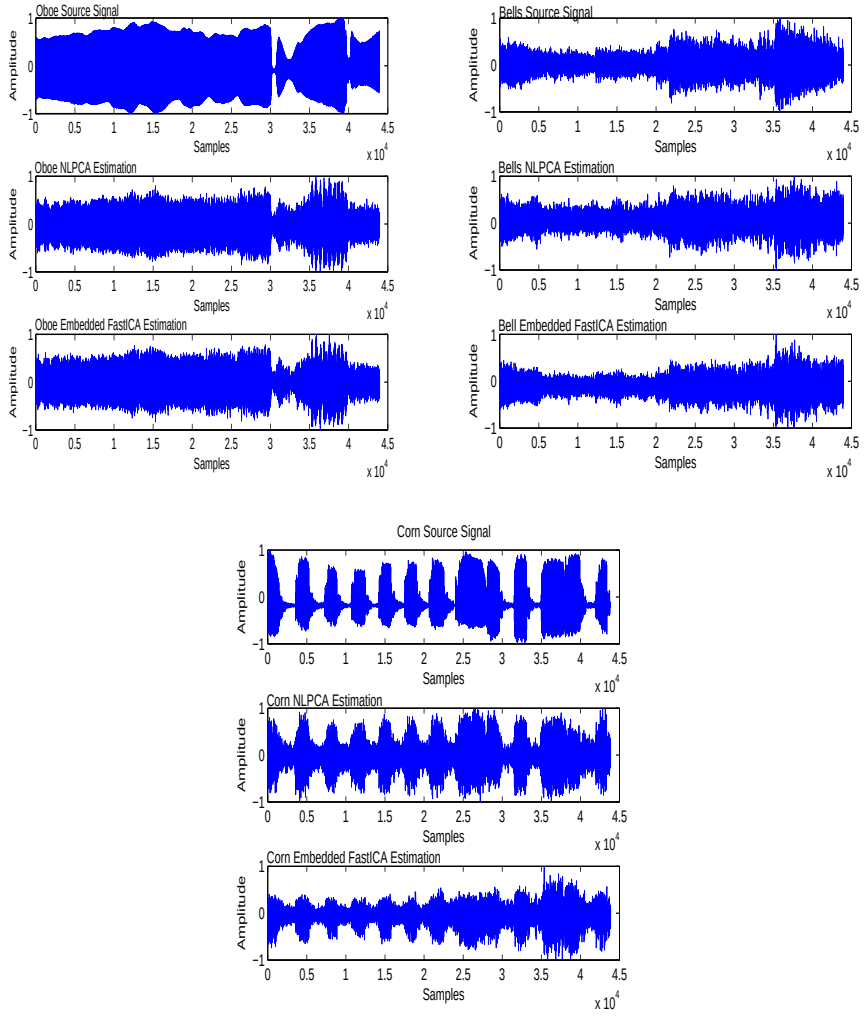
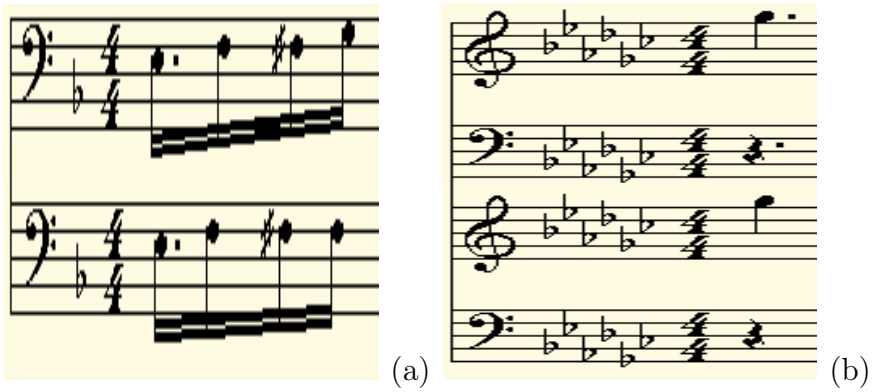


Figure 7.38: The comparison among original source signal (top), NLPCA Estimation (middle), Embedded FastICA Estimation (down): (a) oboe signal, (b) bells signal,(c) corn signal.

Experiment	Music Instruments	NLPCA Approach	FastICA Approach
Experiment 1	...	...	...
...	Oboe	94%	75%
...	Cello	85%	70%
...	Gong	50%	45%
Experiment 2	...	...	...
...	Castanets	50%	20%
...	Xylophone	50%	15%
...	Viola	83%	51%
Experiment 3	...	...	...
...	Castanets	50%	5%
...	Bell	55%	18%
...	Viola	83%	62%
Experiment 4	...	...	...
...	Guitar	50%	32%
...	Oboe	85%	75%
...	Viola	80%	66%
Experiment 5	...	...	...
...	Corn	50%	40%
...	Oboe	90%	62%
...	Bell	55%	30%



(a) (b)

Figure 7.39: Musical transcription: (a) the cello scores extracted from the source signal (up) and from the separated signal (down); (b) the oboe scores extracted from the source signal (up) and from the separated signal (down);.

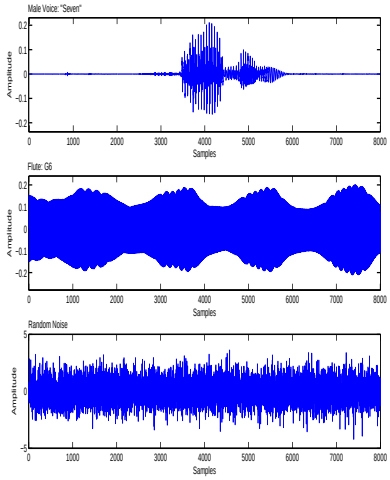


Figure 7.40: Seven - Flute note Experiment source signals: male voice (up); flute note (middle); gaussian Noise (down).

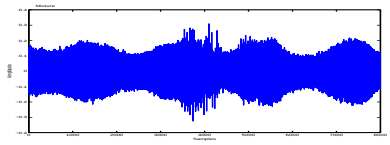


Figure 7.41: Seven - Flute note Experiment mixture.

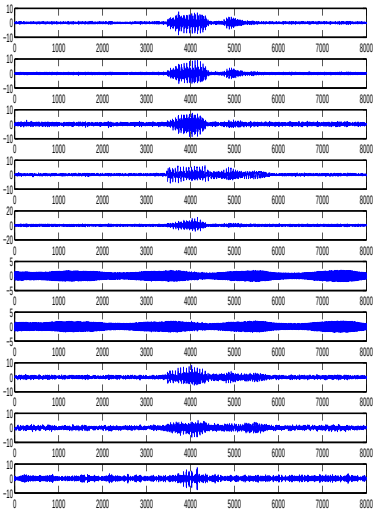


Figure 7.42: Seven - Flute note Experiment: Embedded Fastica Results

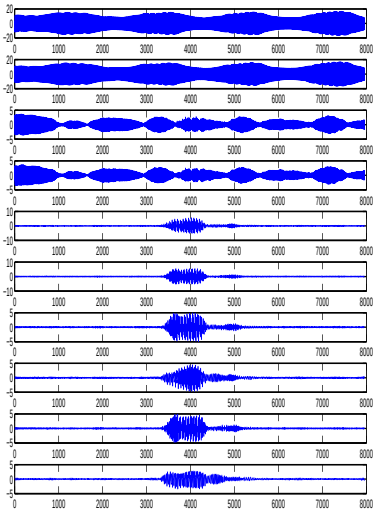


Figure 7.43: Seven - Flute note Experiment: Non Linear PCA Results

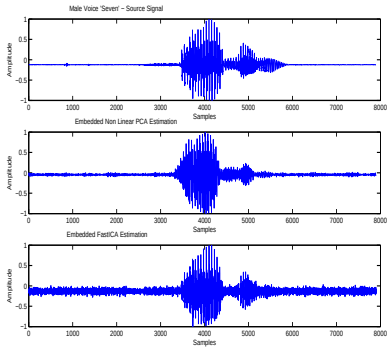


Figure 7.44: Comparison of the original sources (top) with the Embedded Non Linear PCA approach (middle) and FastICA approach (down): male voice.

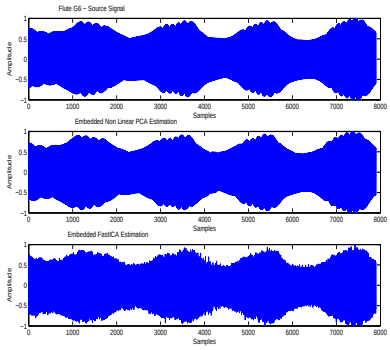


Figure 7.45: Comparison of the original sources (top) with the Embedded Non Linear PCA approach (middle) and FastICA approach (down): flute note.

Chapter 8

Conclusions

In this thesis, we introduce a methodology to accomplish single channel mixtures BSS. The proposed approach is based on the combination of an on-line Robust PCA NN and the chaotic system theory. In particolar, we use the embedding dimension and the time lag to define directly the architecture of the NN. This is a very interesting and innovative approach, in fact we can say that it solves the problem of independent components separation in a good way. We can also say that at the moment the methods described in literature that can accomplish separation of independent components from single mixtures all use *a priori* knowledge about the original sources that form the mixture or have some knowledge about the mixing process. Our method insteand is completely blind, the only thing that we need to know for applying it is the data of the mixture.

We also compared the method with that based on a batch ICA approach. From our experiments, we found that, as in this example, in many cases the robust PCA NN permits to obtain better results than in the other approach. We also can stress that one of the features of the on-line learning is that permits to define the NN's input dimension that improves the separation of the signals.

In the next future the authors will focus their attention on the application of the method to separate signals coming from real environments: astrophysics, geophysics and music, and to find the correlation between the embedding dimension of the signals and the separation ability of our model.

Bibliography

[1] H. Abarbanel. *Analysis of Observed Chaotic Data*. Springer, 1996.

[2] F. Acernese and et al. Virgo status. *Classical and Quantum Gravity*, 21:S385–S394, 2004.

[3] S. Amari, S. Douglas, A. Cichoki, and H. Yang. Multichannel blind deconvolution and equalization using the natural gradient. In *Proc. IEEE Workshop on Signal Processing Advances in Wireless Communications*, pages 101–104, 1997. Paris, France.

[4] S. I. Amari, A. Cichocki, and Y. H.H. *A New Learning Algorithm for Blind Source Separation Advances in Neural Information Processing Systems 8*. MIT Press, Cambridge, MA, 1996.

[5] B. Ans, J. Hérault, and C. Jutten. Adaptive neural architectures: Defection of primitives. In *Proceedings of COGNITIVA '85*, pages 593–597, 1985. Paris, France.

[6] H. Attias. Independent factor analysis. *Neural Computation*, 11(4):803–851, 1999.

[7] F. R. Bach and M. J. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

[8] F. R. Bach and M. J. Jordan. Beyond independent components: Trees and clusters. *Journal of Machine Learning Research*, 4:1205–1233, 2003.

[9] A. J. Bell and T. J. Sejnowski. An information - maximization approach to blind source separation and blind deconvolution. *Neural Computing*, 7:1129–1159, 1995.

[10] A. J. Bell and T. J. Sejnowski. *A Non Linear Information Maximization Algorithm that Performs Blind Separation. Advances in Neural Information Processing Systems 7*. MIT Press, Cambridge, MA, 1995.

[11] C. Bishop. *Neural networks for pattern recognition*. Clarendon Press, 1996. Oxford.

[12] A. S. Bregman. *Auditory Scene Analysis*. MIT Press, 1994.

[13] L. Cao. Practical method for determining the minimum embedding dimension of a scalar time series. *Physica D*, 110:43–50, 1997.

[14] J.-F. Cardoso. Eigen-structure of the fourth-order cumulant tensor with application to the blind source separation problem. In *Proc. ICASSP'90*, pages 2655–2658, 1990. Albuquerque, NM, USA.

[15] J.-F. Cardoso. Iterative techniques for blind source separation using only fourth order cumulants. In *Proc. EUSIPCO*, pages 739–742, 1992. Brussels, Belgium.

[16] J.-F. Cardoso. Infomax and maximum likelihood for source separation. *IEEE Letters on Signal Processing*, 4:112–114, 1997.

[17] J.-F. Cardoso. Entropic contrasts for source separation. In S. Haykin, editor, *Adaptive Unsupervised Learning*. 1999.

[18] J.-F. Cardoso and P. Comon. Independent component analysis, a survey of some algebraic methods. In *Proc. ICASSP'96*, volume 2, pages 93–96, 1996.

[19] J. F. Cardoso and A. Souloumaic. Equivariant adaptive source separation. In *IEE Proceedings - F*, volume 140, pages 362–370, 1993.

[20] S. Choi, H. M. Park, and S. Lee. Blind source separation and independent component analysis: A review. *Neural Information Processing - Letters and Reviews*, 6(1), 2005.

[21] A. Cichocki and L. Moszczynski. A new learning algorithm for blind source separation of sources. *Electronics Letters*, 28(21):1986–1987, 1992.

[22] A. Cichocki and R. Unbehauen. *Neural networks for optimization and signal processing*. John Wiley, 1993. New York.

[23] A. Cichocki and R. Unbehauen. Robust estimation of principal components by using neural network learning algorithms. *Electronic Letters*, 29:1869–1870, 1993.

[24] A. Cichocki and R. Unbehauen. Robust neural networks with on-line learning for blind identification and blind separation of sources. *IEEE Transaction on Circuits and Systems*, 43(11):894–906, 1996.

[25] P. Comon. Separation of stochastic processes. In *Proc. Workshop on Higher-Order Spectral Analysis*, pages 174–179, 1989. Vail, Colorado.

[26] P. Comon. Independent component analysis - a new concept? *Signal Processing*, 36:287–311, 1994.

[27] A. M. Fraser and H. L. Swinney. Independent coordinates for strange attractors from mutual information. *Physical Review A*, 33:1134–1140, 1986.

[28] J. Hérault and B. Ans. Circuits neuronaux è synapses modifiables: Décodage de messages composities par apprenitssage non supervisé. *C.-R. de l'Accadémie des Sciences*, 299(III-3):525–528, 1984.

[29] H. H. Harman. *Modern Factor Analysis*. University of Chicago Press, 2nd Edition, 1967.

[30] S. Haykin. *Adaptive Filter Theory*. Prentice Hall, 1995.

[31] C. W. Helstrom. *Statistical Theory of Signal Detection*. Pergamon Press, 1968. London, England.

[32] P. J. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–475, 1985.

[33] A. Hyvärinen. One-unit contrast functions for independent component analysis: a statistical analysis. In *Proc. IEEE Workshop on Neural Networks for Signal Processing*, pages 388–397. Neural Networks for Signal Processing VII, 1997. Amelia Island, Florida.

[34] A. Hyvärinen. Independent component analysis in the presence of gaussian noise by maximizing joint likelihood. *Neurocomputing*, 22:49–67, 1998.

[35] A. Hyvärinen. New approximations of differential entropy for independent component analysis and projection pursuit. *Advances in Neural Information Processing System*, (10):273–279, 1998. MIT Press.

[36] A. Hyvärinen, P. Hoyer, and M. Inki. Topographic independent component analysis. *Neural Computation*, 13(7):1527–1558, 2001.

[37] A. Hyvärinen and M. Inki. Estimating overcomplete independent component bases for image windows. *Journal of Mathematical Imaging and Vision*, 17:139–152, 2002.

[38] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.

[39] A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.

[40] A. Hyvärinen and E. Oja. Independent component analysis by general non-linear hebbianlike learning rules. *Signal Processing*, 64(3):301–313, 1998.

[41] A. Hyvärinen and E. Oja. Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128, 1999.

[42] C. James and D. Lowe. Extracting multisource brain activity from a single electromagnetic channel. *Artificial Intelligence in Medicine*, 28:89–104, 2003.

[43] G.-J. Jang and T. W. Lee. The statistical structures of male and female speech signals. In *Proc. ICASSP 2001*, 2001. Salt Lake City, Utah.

[44] G.-J. Jang and T. W. Lee. A maximum likelihood approach to single channel source separation. *Journal of Machine Learning*, 4:1365–1392, 2003.

[45] G.-J. Jang and T. W. Lee. A probabilistic approach to single channel source separation. In M. Press, editor, *Advances in Neural Information Processing Systems*, volume 15, 2003. Cambridge.

[46] I. T. Jolliffe. *Principal Component Analysis*. SpringerVerlag, 1986.

[47] M. C. Jones and R. Sibson. What is projection pursuit? *J. of the Royal Statistical Society*, ser. A,(150):1–36, 1987.

[48] C. Jutten and J. Hérault. Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10, 1991.

[49] J. Karhunen and J. Joutsensalo. Sinusoidal frequency estimation by signal subspace approximation. *IEEE Transaction on Signal Processing*, 40(12):2961–2972, 1992.

[50] J. Karhunen and J. Joutsensalo. Representation and separation of signals using non-linear pca type learning. *Neural Networks*, 7(1):113–127, 1994.

[51] J. Karhunen and J. Joutsensalo. Generalizations of principal component analysis, optimization problems and neural networks. *Neural Networks*, 8(4):549–562, 1995.

[52] J. Karhunen, E. Oja, L. Wang, R. Vigario, and J. Joutsensalo. A class of neural networks for independent component analysis. *IEEE Trans. on Neural Networks*, 8(3):486–504, 1997.

[53] M. B. Kennel, R. Brown, and H. D. Abarbanel. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical Review A*, 45(6):3403–3411, 1992.

[54] D. D. Lee and H. S. Seung. Learning of the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[55] M. Lewicki and T. Sejnowski. Learning nonlinear overcomplete representations for efficient coding. *Advances in Neural Information Processing Systems*, 10:815–821, 1998.

[56] K. Matsuoka, M. Ohya, and M. Kawamoto. A neural net for blind separation of nonstationary signals. *Neural Networks*, 8(3):411–419, 1995.

[57] E. Oja. Principal components, minor components and linear neural networks. *Neural Networks*, 5:927–935, 1992.

[58] E. Oja, J. Karhunen, L. Wang, and R. Vigario. Principal and independent components in neural networks - recent developments. In *Proc. of VII Italian Workshop on Neural Networks*, 1995. Vietri sul mare, Salerno, Italy.

[59] E. Oja, H. Ogawa, and J. Wangviwattana. Learning in non-linear constrained hebbian networks. In *Proc. Int. Conf. on Artificial Neural Networks (ICANN'91)*, pages 385–390, 1991. Espoo, Finland.

[60] A. Papoulis. *Signal Analysis*. Mc Graw Hill, 1984. Singapore.

[61] A. Papoulis. *Probability, Random Variables and Stochastic Process*. McGraw Hill, 1991.

[62] B. Pearlmutter and L. Parra. A context-sensitive generalization of ica. In *Proc. ICONIP'96*, pages 151–157, 1996. Hong Kong.

[63] D. T. Pham, P. Garrat, and C. Jutten. Separation of a mixture of independent sources through a maximum likelihood approach. In *Proc. EUSIPCO*, pages 771–774, 1992. Brussels, Belgium.

[64] S. Roweis. One microphone source separation. In *Neural Information Processing Systems*, volume 13, pages 793–799, 2000.

[65] T. Sanger. Optimal unsupervised learning in a single-layer linear feedforward network. *Neural Networks*, 2:459–473, 1989.

[66] T. Sanger. An optimality principle for unsupervised learning. *Advances in Neural Information Processing Systems*, 1:11–19, 1989.

[67] P. R. Saulson. *Fundamentals of Interferometric Gravitational Wave Detectors*. World Scientific Pub., 1994.

[68] D. Sigg and et al. Ligo status. *Classical and Quantum Gravity*, 21:S409–S415, 2004.

[69] Takahashi and et al. Tama status. *Classical and Quantum Gravity*, 21:S403–S408, 2004.

[70] F. Takens. Detecting strange attractors in turbulence. In D. Rand and L. Y. Springer-Verlag, editors, *In Dynamical Systems and Turbulence Lecture Notes in Mathematics*, volume 898, pages 366–381, 1995. Berlin.

[71] B. Willke and et al. Geo status. *Classical and Quantum Gravity*, 21:S417–S423, 2004.

[72] L. Zhang, A. Cichocki, and S. Amari. Multichannel blind deconvolution of nonminimum-phase systems using filter decomposition. *IEEE Transactions on Signal Processing*, 52(5):1430–1442, 2004.