A Critical Comparison of ICA Algorithms

Pierre Clapier

MSc (by Research) in Pattern Analysis and Neural Networks

ASTON UNIVERSITY

September 2001

ASTON UNIVERSITY

A Critical Comparison of ICA Algorithms

PIERRE CLAPIER

MSc (by Research) in Pattern Analysis and Neural Networks, 2001

Thesis Summary

Independent component analysis (ICA), is a statistical method for transforming a multi-dimensional random vector into components that are statistically as independent from each other as possible. Recently, in a paper by H. Attias [1], a model called independent factor analysis trained by an Expectation Maximisation (EM) algorithm has been proposed which seems to supersede all earlier work, since it can cope with arbitrary source distributions and non-square mixing matrices. In this thesis we will first explain what ICA is, what are the different ways to solve the ICA problem and present some algorithms with a special highlight on IFA. Then we will propose some methods to reduce the dimensionality and to estimate the noise using PCA and Factor Analysis (FA) tools. Finally we will compare FastICA [13] and IFA, present a method to solve the ICA problem in the case of many sensors and significant noise, then apply this method on a concrete problem: MEG analysis.

Keywords: Independent Component Analysis, blind separation of signals, entropic contrasts, cumulants, mixture of Gaussians, Independent Factor Analysis, automatic choice of dimensionality, cricket songs, MEG, Principal Component Analysis.

Acknowledgements

I specially want to thank my supervisor Ian Nabney for his availability, his help, and useful advice all through the project, Christopher James for his enthusiasm explaining the different applications of ICA to MEG, and the whole NCRG department of Aston University which helped me to spend my MSc in the best conditions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A central problem in neural network research, as well as in statistics, is to find a suitable representation of the data. Let us denote by $\mathbf{x}$ an m-dimensional random variable; the problem is then to find a linear transformation, so that the n-dimensional transform $\mathbf{s} = (s_1, s_2, s_3, ..., s_n)^T$ defined by

$$\mathbf{s} = \mathbf{W}\mathbf{x} \tag{1.1}$$

has some desirable properties.

Several principles and methods have been developed to find a suitable linear transformation. These include principal component analysis (PCA), factor analysis and projection pursuit.

Recently, a particular method for finding a linear transformation, called independent component analysis (ICA), has gained wide-spread attention. As the name implies, the basic goal is to find a transformation in which the components s_i are statistically as independent from each other as possible. ICA can be applied for example to blind source separation (BSS), in which the observed values of $\mathbf{x}$ correspond to a realization of an m-dimensional discrete-time signal $\mathbf{x}(t), t = 1, 2, \ldots, T$. Then the components $s_i(t)$ are called source signals.

Basically, ICA was developed to deal with problems that are closely related to the cocktail-party problem. Imagine that you are in a room where two people are speaking simultaneously. There are two microphones in different locations. They give two recorded time signals $x_1(t)$ and $x_2(t)$, with t the time index. Each of these recorded signals is a weighted sum of the speech signals emitted by the two speakers, $s_1(t)$ and $s_2(t)$. This correspond to the linear system:

$$\begin{aligned} x_1(t) &= a_{11}s_1 + a_{12}s_2 \\ x_2(t) &= a_{21}s_1 + a_{22}s_2, \end{aligned} \tag{1.2}$$

where a_{11}, a_{12}, a_{21} and a_{22} are some parameters that depend on the distances of the microphones from the speaker.

So we chose a problem which correspond to this cocktail-party model and has the characteristics of many ICA problems. We concentrate on the application of ICA to MEG analysis. This is typical to many practical problems in that the data have sensor noise, m is large and we believe that $n < m$.

In order to reduce the dimension of the sensors, past studies have just used the largest few eigenvalues during the whitening stage (see appendix A.2). However, it's often difficult to find a principled estimate of the number of sources only taking into account eigenvalues of the sensor data. Moreover in the MEG problem, many signals we are interested in may have small eigenvalues.

So we need methods to:

- determine the number of underlying source components,
- reduce the dimensionality of the signals without using only largest eigenvalues,
- separate the signals in a reasonable time.

Furthermore, the whole recovery of the sources process, should reduce the noise.

In this paper we first review the theory and methods for ICA (chapter 2), in chapter 3 we describe the model and the learning rules of Independent Factor Analysis [1] and in chapter 4 we discuss some important questions for the application of ICA to MEG analysis: how to estimate the number of sources? is it possible to use IFA when we face large number of sources? which pre-whitening shall we use for FastICA?

Then in chapter 5 we propose a method to perform ICA when n is large and the sensor noise significant, to apply it to MEG analysis in chapter 5.3.

Chapter 2

Independent component analysis

2.1 Definitions of linear independent component analysis

Definition 1 *(Noisy ICA model) ICA of a random vector* $\mathbf{x}$ *consists of estimating the following generative model for the data:*

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \boldsymbol{\eta} \tag{2.1}$$

where the latent variables (components) s_i *in the vector* $\mathbf{s} = (s_1, s_2, s_3, ..., s_n)^T$ *are assumed to be independent. The matrix* $\mathbf{A}$ *is a constant* $m \times n$ *"mixing" matrix, and* $\boldsymbol{\eta}$ *is a* m*-dimensional random noise vector.*

$\mathbf{x}$ is known as the vector of sensors and $\mathbf{s}$ the vector of sources. This model is used in the algorithm of H. Attias ([1] and [2]), but many of the earlier papers consider a noise-free model

$$\mathbf{x} = \mathbf{A}\mathbf{s}. \tag{2.2}$$

Two key issues of ICA are the definition of a measure of independence and the design of algorithms to find the change of basis (or separating matrix) $\mathbf{A}$ optimising this measure.

2.2 Identifiability of the ICA model

The identifiability of the noise-free model as been discussed in [8] and can be assured if:

- all the independent components s_i, with the possible exception of one component, must be non-Gaussian (see appendix B),
- the number of observed linear mixtures m is at least as large as the number of independent components n, i.e., $m \geq n$,
- the matrix $\mathbf{A}$ must be of full column rank.

Usually it's also assumed that $\mathbf{x}$ and $\mathbf{s}$ are centred. Moreover, as we can only determine the columns of $\mathbf{A}$ up to a multiplicative 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).

Furthermore, we should notice that the definitions of ICA given above imply no ordering of the independent components, which is in contrast to PCA.

Finally, some algorithms require square mixing matrices, $n = m$ (it's the case of the FastICA algorithm ([13] and section 2.4.2)). For $n = m$, once we get the mixing matrix $\mathbf{A}$, we can compute its inverse, $\mathbf{W}$ and obtain the independent components:

$$\mathbf{s} = \mathbf{W}\mathbf{x}, \tag{2.3}$$

$\mathbf{W}$ is the unmixing matrix.

Mixing in realistic situations, however, generally includes noise and different numbers of sources and sensors. As the noise level increases, the performance of such a model deteriorates and the separation quality decreases. More importantly, many situations like MEG analysis have a relatively small number of sources but many sen-

sors, the square mixing matrix assumption is not realistic. Hence if we use FastICA or JADE, we will need methods to reduce the dimensionality of the data before separating them with these algorithms.

2.3 Measures of independence

2.3.1 Introduction

Estimation of the independent component analysis model is usually performed by formulating an objective function and then maximising or minimising it. Often such a function is called a contrast function, but some authors reserve this term for a certain class of objective functions [8]; the terms loss function or cost function are also used.

2.3.2 Measures of non-Gaussianity

The central limit theorem states that the distribution of a sum of independent random variables with a finite mean and variance tends toward a Gaussian distribution as the number of variables increases. Thus a sum of two independent random variables usually has a distribution that is closer to Gaussian than either of the two original random variables.

For simplicity let us assume that all the independent components have identical distributions. To estimate one of the independent components, we consider a linear combination of the x_i ; let us denote this by $y = \mathbf{w}^T\mathbf{x}$, where $\mathbf{w}$ is a vector to be determined. If $\mathbf{w}$ was one of the rows of the unmixing matrix $\mathbf{W}$, this linear combination would equal one of the independent components s_i.

Now let $\mathbf{z} = \mathbf{A}^T\mathbf{w}$. Then we have $y = \mathbf{w}^T\mathbf{x} = \mathbf{w}^T\mathbf{A}\mathbf{s} = \mathbf{z}^T\mathbf{s}$. y is thus a linear combination of the s_i. Since a sum of even two independent random variables is more Gaussian than the original variables, $\mathbf{z}^T\mathbf{s}$ is more Gaussian than any of the s_i and becomes least Gaussian when it in fact equals one of the s_i. In this case obviously, only one of the elements z_i of $\mathbf{z}$ is non-zero (note that the s_i here are assumed to have

identical distributions).

Therefore, we could take as $\mathbf{w}$ a vector that maximises the non-Gaussianity of $\mathbf{w}^T\mathbf{x}$. Such a vector would necessary correspond to a $\mathbf{z}$ which has only one non-zero component. This means $\mathbf{w}^T\mathbf{x} = \mathbf{z}^T\mathbf{s}$ equals one of the independent components.

To find several independent components, we need to find all the maxima ($2n$ as the independent components can be estimated only up to a multiplicative constant). This can be performed as the independent components are uncorrelated, we constrain the search to the space that gives estimates uncorrelated with the previous one. This corresponds to orthogonalisation in a suitable space which is done during the whitening stage (cf appendix A.2).

It is interesting to show the link between projection pursuit and this approach to solve the ICA model. Projection pursuit [16, 15, 25, 21] is a technique developed in statistics for finding "interesting" projections of the multidimensional data. In basic (1-D) projection pursuit, we try to find the directions such that the projections of the data in those directions have interesting distributions, i.e. display some structure. It has been argued by Huber [25] and by Jones and Sibson [21] that the Gaussian distribution is the least interesting one and that the most interesting directions are those that show the least Gaussian distribution. This is what we do to estimate the ICA model.

Kurtosis

A classical measure of non-Gaussianity is kurtosis (the fourth order cumulant):

$$\text{kurt}(y) = \text{E}\{y^4\} - 3(\text{E}\{y^2\})^2. \tag{2.4}$$

Actually, when we assume that y is of unit variance, the right-hand side simplifies to $\text{E}\{y^4\} - 3$.

Random variables that have a negative kurtosis are called sub-Gaussian, and those

with positive kurtosis are called super-Gaussian. The Gaussian have zero kurtosis (see figure 2.1).

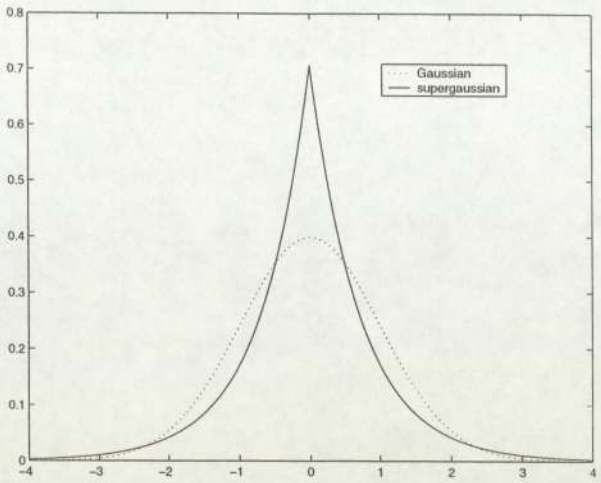

Figure 2.1: The density of the Laplace distribution which is a typical super-Gaussian distribution and the Gaussian distribution in dotted line .

The main problem is that kurtosis is very sensitive to outliers and is therefore not a robust measure of non-Gaussianity.

Entropy and Negentropy

The entropy of a random variable can be interpreted as the degree of information that observation of the variable gives. Entropy H can be generalised for continuous random variables and vectors (differential entropy):

$$\mathrm{H}(\mathbf{y}) = -\int \mathrm{f}(\mathbf{y}) \log \mathrm{f}(\mathbf{y}) dy. \tag{2.5}$$

where $\mathbf{y}$ is a random vector with density $\mathrm{f}(\mathbf{y})$.

A fundamental result of information theory is that a Gaussian variable has the largest entropy among all random variables of equal variance [9]. Thus with entropy it is possible to construct measures of non-Gaussianity; the FastICA (see section 2.4.2) algorithm defines negentropy that is zero for a Gaussian variable and always nonnega-

tive:

$$\mathrm{J}(\mathbf{y}) = \mathrm{H}(\mathbf{y}_{\mathbf{gauss}}) - \mathrm{H}(\mathbf{y}), \tag{2.6}$$

where $\mathbf{y}_{\mathbf{gauss}}$ is a Gaussian random variable with the same covariance matrix as $\mathbf{y}$. This is computationally very difficult to obtain, so in the scientific literature we can see approximations for a random variable, of the form:

$$\mathrm{J}_{\mathrm{G}}(y) = |\,\mathrm{E}_y\{\mathrm{G}(y)\} - \mathrm{E}_\nu\{\mathrm{G}(\nu)\}|^p, \tag{2.7}$$

where G is a sufficiently smooth function, ν a standardised Gaussian random variable, y is assumed to be normalised to unit variance, and the exponent $p = 1$ or 2 typically. The choice of G depends on the statistical properties of the estimator, and the knowledge of the components.

2.3.3 Minimisation of mutual information

Another approach to ICA estimation is minimisation of mutual information. We define the mutual information I between m scalar random variables, $y_i, i = 1, \ldots, m$ as follows:

$$\mathrm{I}(y_1, y_2, \ldots, y_m) = \sum_{i=1}^{m} \mathrm{H}(y_i) - \mathrm{H}(\mathbf{y}). \tag{2.8}$$

An important property of mutual information [9, 26] is that we have for an invertible linear transformation $\mathbf{y} = \mathbf{W}\mathbf{x}$:

$$\mathrm{I}(y_1, y_2, \ldots, y_n) = \sum_{i=1}^{n} \mathrm{H}(y_i) - \mathrm{H}(\mathbf{x}) - \log|\det \mathbf{W}|. \tag{2.9}$$

Then, if the y_i are uncorrelated and of unit variance, this implies that $\mathrm{E}\{\mathbf{y}\mathbf{y}^T\} = \mathbf{I}$ which gives $\det \mathbf{W}$ (when $\mathrm{E}\{\mathbf{y}\mathbf{y}^T\} = \mathbf{I}$) is constant as:

$$\det \mathbf{I} = 1 = (\det \mathbf{W}\, \mathrm{E}\{\mathbf{x}\mathbf{x}^T\}\mathbf{W}^T) = (\det \mathbf{W})(\det(\mathrm{E}\{\mathbf{x}\mathbf{x}^T\}))(\det \mathbf{W}^T).$$

Moreover for y_i of unit variance, entropy and negentropy differ only by a constant and the sign. We obtain:

$$\mathrm{I}(y_1, y_2, \ldots, y_n) = C - \sum_{i=1}^{n} \mathrm{J}(y_i), \tag{2.10}$$

where C is a constant that does not depend on $\mathbf{W}$.

Since mutual information is the natural information-theoretic measure of the independence of random variables, we can use it as the criterion for finding an ICA transform. In this approach we define the ICA of a random vector $\mathbf{x}$ as an invertible transformation, where the unmixing matrix $\mathbf{W}$ ($\mathbf{s} = \mathbf{W}\mathbf{x}$) is determined so that the mutual information of the transformed components s_i is minimised.

It was shown in [32] that ICA estimation by minimisation of mutual information is equivalent to maximising the sum of non-Gaussianities of the estimates, when the estimates are constrained to be uncorrelated. So we can use the simpler form of Eq(2.10) instead of Eq(2.9) when the constraint of uncorrelatedness is satisfied. Thus the formulation of ICA as mutual information optimisation gives an other justification of the idea of finding maximally non-Gaussian directions discussed in section 2.3.2.

2.3.4 Maximum Likelihood Estimation and Infomax Principle

A very popular approach for determining the ICA model is maximum likelihood estimation, which is closely connected to the infomax principle.

It is possible to formulate directly the likelihood in the noise-free ICA model and then estimate the model by a maximum likelihood method. Denoting by $\mathbf{W} = (\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_n})^T$ the unmixing matrix such as $\mathbf{s} = \mathbf{W}\mathbf{x}$, the log-likelihood takes the form (see [10]):

$$\mathrm{L} = \sum_{t=1}^{T} \sum_{i=1}^{n} \log \mathrm{f_i}(\mathbf{w_i}^T \mathbf{x}(t)) + T \log |\det \mathbf{W}| \tag{2.11}$$

where the $\mathrm{f_i}$ are the density functions of the s_i (here assumed to be known), and the

$\mathbf{x}(t)$, $t = 1, \ldots, T$ are the realizations of $\mathbf{x}$. One can maximises the likelihood using gradient descent or Expectation Maximisation (EM) algorithms.

Another related function was derived from a neural network viewpoint. This was based on maximising 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 $g_i(\mathbf{w_i}^T\mathbf{x})$, where g_i are some non-linear scalar functions, and the $\mathbf{w_i}$ are the weight vectors of the neurons. One then wants to maximise the output entropy:

$$L_2 = H(g_1(\mathbf{w_1}^T\mathbf{x}), \ldots, g_n(\mathbf{w_n}^T\mathbf{x})). \tag{2.12}$$

If the g_i are well chosen, this framework also supports estimation of the ICA model. Indeed, it was proved [4][27] that the principle of network entropy maximisation, or "infomax", is equivalent to maximum likelihood estimation. This equivalence requires that the g_i used are chosen as the cumulative distribution functions corresponding to the densities f_i, i.e., $g_i'(.) = f_i(.)$.

The problem with maximum likelihood estimation is that the densities f_i must be estimated correctly. They need not be estimated with any great precision: in fact it is enough to estimate whether they are sub- or super-Gaussian. But if the information on the nature of the independent components is not correct , ML estimation will give completely wrong results.

In contrast, using reasonable measures of non-Gaussianity with algorithms based on Entropy, this problem does not usually arise.

However, people have introduced more flexible density models that support maximum likelihood estimation with little or no prior knowledge of the densities ([1] and [7]), some other models are quite successful.

2.4 Algorithms to find the change of basis

In the scientific literature, it was shown that the basic choice of the ICA method seems to reduce to two questions. First the choice between estimating all the independent components at the same time (multi-unit), and estimating only a subset of them, possibly one-by-one (one-unit). Most ICA research has concentrated on the first option, but in practice, it seems that the second option is very often more interesting, due to computational considerations. Second, one has the choice between on-line algorithms and batch-mode (or block) algorithms. Again, most research has concentrated on the former option, although in many applications, the latter option seems to be preferable, again for computational reasons.

In the on-line case, most algorithms use stochastic gradient methods. In the case where all the independent components are estimated at the same time, the most popular algorithm is natural gradient ascent or the likelihood, or related contrast functions, like infomax.

In the one-unit case, straightforward stochastic gradient methods give on-line algorithms that maximise negentropy or its approximations.

In the case where the computations are made in batch-mode, much more efficient algorithms are available (FastICA [13], tensor based methods [8] [6]).

One more choice is possible between data-based and statistic-based techniques. In the data-based option, successive linear transformations are applied to the data set until the contrast function is optimised. The alternative is to summarise the data set by a smaller set of "statistics" which are computed once and for all from the data set. The algorithm then estimates a separating matrix as a function of these statistics without accessing the data (this option is the one used in the JADE algorithm).

We now discuss the three most commonly used and provided ICA algorithms: JADE, FastICA and IFA.

2.4.1 JADE algorithm (JF. Cardoso)

JADE [5] (Joint Approximate Diagonalization of Eigen-matrices) is a 4th-order statistic-based algorithm which can be summarised as:

1. *Initialisation.* Estimate a whitening matrix $\hat{\mathbf{W}}$ and set $\mathbf{z} = \hat{\mathbf{W}}\mathbf{x} = \hat{\mathbf{W}}\mathbf{A}\mathbf{s}$.

2. *Form statistics.* Estimate a set $\{\hat{\mathbf{Q}}_\mathbf{i}^\mathbf{z}\}$ of cumulant matrices of $\mathbf{z}$ $(i = 1, \ldots, n)$.

3. *Optimise an orthogonal contrast.* Find the rotation matrix $\hat{\mathbf{V}}$ such that the cumulant matrices are "as diagonal as possible" i.e., solve
   $\hat{\mathbf{V}} = \arg\min \sum_i \text{Off}(\mathbf{V}^T \hat{\mathbf{Q}}_\mathbf{i}^\mathbf{z} \mathbf{V})$.

4. *Separate.* Estimate $\mathbf{A}$ as $\hat{\mathbf{A}} = \hat{\mathbf{V}}\hat{\mathbf{W}}$ and/or estimate the components $\hat{\mathbf{s}} = \hat{\mathbf{A}}^{-1}\mathbf{x} = \hat{\mathbf{V}}^T\mathbf{z}$.

$\text{Off}(\mathbf{F})$ is defined as the sum of the squares of the non-diagonal elements,

$$\text{Off}(\mathbf{F}) \stackrel{def}{=} \sum_{i \neq j} (f_{ij})^2, \tag{2.13}$$

and for any $n \times n$ matrix $\mathbf{M_i}$, the cumulant matrices are defined by:

$$\mathbf{Q}_\mathbf{i}^\mathbf{z} = \mathbf{U}\tilde{\mathbf{\Delta}}\mathbf{U}^T \quad \tilde{\mathbf{\Delta}} = \text{Diag}(\text{kurt}(s_1)\mathbf{u_1}^T\mathbf{M_i}\mathbf{u_1}, \ldots, \text{kurt}(s_n)\mathbf{u_n}^T\mathbf{M_i}\mathbf{u_n}) \tag{2.14}$$

with $\mathbf{U} = \mathbf{WA} = (\mathbf{u_1}, \ldots, \mathbf{u_n})$.

The 4th-order techniques described in [5] are not adaptive to the distribution of the sources (we don't need to know the source distribution) but the main problem is that they have potential sensitivity to outliers (see [5]). Moreover JADE is impractical for a large number of components as the estimation of the cumulant matrices is proportional to $n \times n$.

2.4.2 FastICA algorithm (A. Hyvärinen)

In many practical situations, we don't need on-line algorithms whose convergence is often slow. FastICA is a batch algorithm based on a fixed point iteration [13]. At first the contrast function was kurtosis and later it was generalised for other contrast functions. For the pre-processed data (see appendix A), the one-unit FastICA algorithm has the following form:

$$\mathbf{w}(k) = \mathrm{E}\{\mathbf{x}\, \mathrm{g}(\mathbf{w}(k-1)^T\mathbf{x})\} - \mathrm{E}\{\mathrm{g}'(\mathbf{w}(k-1)^T\mathbf{x})\}\mathbf{w}(k-1), \tag{2.15}$$

where the weight vector $\mathbf{w}$ is also normalised to unit length after every iteration, and the function g is the derivative of the function G used in the general contrast function Eq(2.7).

Instead of using every data point immediately for learning, FastICA uses sample averages computed over larger samples of the data. The convergence speed of the fixed point algorithm is clearly superior to the usual algorithms. Moreover FastICA can be used both to optimise one-unit and multi-unit contrast functions which is rather slow, using the following iterative algorithm:

$$\begin{aligned} &1.\ \text{Let } \mathbf{W} = \mathbf{W}/\sqrt{\| \mathbf{WCW}^T \|} \\ &\text{Repeat 2. until convergence :} \\ &2.\ \text{Let } \mathbf{W} = \frac{3}{2}\mathbf{W} - \frac{1}{2}\mathbf{WCW}^T\mathbf{W} \end{aligned} \tag{2.16}$$

where $\mathbf{C} = \mathrm{E}\{\mathbf{x}\mathbf{x}^T\}$.

The most commonly used version of the FastICA algorithm is the one-unit algorithm as it is much faster than many ICA algorithm. Therefore we will only talk about the one-unit FastICA algorithm in the following sections.

From this point, the reader must be aware that FastICA provides a fast and fairly accurate algorithm to perform ICA, but:

1. the mixing matrix must be square, it implies that we previously reduced the dimensionality of the data,

2. it doesn't estimate the density model which means that to recover the sources we only use the unmixing matrix on the sensors, it won't reduce the noise,

3. the algorithm is supervised as we have to choose the contrast function, some functions fit to super-Gaussian source distribution, other to sub-Gaussian distribution. However, there is a general purpose contrast function when we don't know the source densities.

2.4.3 Independent Factor Analysis method (H. Attias)

Independent factor analysis (IFA) [2] is a two-step procedure where each source is described by a mixture of Gaussians. In the first step, the source densities, mixing matrix and noise covariance are estimated from the observed data by maximum likelihood. For this purpose we use an expectation-maximisation (EM) algorithm, which performs unsupervised learning of an associated probabilistic model of the mixture. In the second step, the sources are reconstructed from the observed data by an optimal non-linear estimator.

Though, if we model each source with a mixture of n_i Gaussians, for each step of the EM algorithm, we have to compute $\prod_{i=1}^{n} n_i$ components, therefore when the number of sensors increases the algorithm becomes intractable. A variational approximation is derived for this case.

It seems that this algorithm is superior to ICA because:

- it models noisy data and non square mixing matrix,

- it gives a framework for PCA, factor analysis (FA) and ICA since it reduces to FA when the sources become Gaussian, and to an EM algorithm for PCA in the

zero-noise limit (see section 4.1.1),

- it can learn arbitrary source densities from the data.

That's why we decided to compare this method with a more "classical" algorithm: FastICA, in order to check whether this algorithm is as appealing as it seems to be.

The next chapter describes the algorithm in more detail.

Chapter 3

Independent Factor Analysis

3.1 The Generative model

In the noisy case, we've seen that the generative model is:

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \boldsymbol{\eta} \tag{3.1}$$

We model the sources s_i as m independent random variables with arbitrary distribution $p(s_i|\theta_i)$, where the individual i-th source density is parameterised by the parameter θ_i. The noise $\boldsymbol{\eta}$ is assumed to be Gaussian with zero mean and full covariance matrix $\boldsymbol{\Lambda}$, allowing correlations between sensors. Hence

$$\mathrm{p}(\boldsymbol{\eta}) = \mathcal{G}(\boldsymbol{\eta}, \boldsymbol{\Lambda}) \tag{3.2}$$

with $\mathcal{G}(\mathbf{x} - \mu, \Sigma) = |\det(2\pi\Sigma)|^{-1/2} \exp[-(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)/2]$. We denote the independent factor (IF) parameters collectively by

$$\Omega = (\mathbf{A}, \boldsymbol{\Lambda}, \theta). \tag{3.3}$$

The resulting model sensor density is

$$\mathrm{p}(\mathbf{x}, \Omega) = \int dx\, \mathrm{p}(\mathbf{x}|\mathbf{s})\, \mathrm{p}(\mathbf{s}) = \int dx\, \mathcal{G}(\mathbf{x} - \mathbf{A}\mathbf{s}, \Lambda) \prod_{i=1}^{n} \mathrm{p}(s_i, \theta_i), \tag{3.4}$$

3.1.1 Source Model

We need to choose a parametric form for $\mathrm{p}(s_i)$, which

- is sufficiently general to model arbitrary source densities,
- allows the integral in Eq(3.4) to be performed analytically.

These conditions can be satisfied by using a mixture of Gaussian (MOG) model, so

$$\mathrm{p}(s_i|\theta_i) = \sum_{q_i=1}^{n_i} w_{i,q_i} \mathcal{G}(s_i - \mu_{i,q_i}, \nu_{i,q_i}), \;\; \theta_i = \{w_{i,q_i}, \mu_{i,q_i}, \nu_{i,q_i}\}, \tag{3.5}$$

where q_i runs over the n_i Gaussians of source i. For this mixture to be normalised, the mixing proportions for each source should be positive and sum up to unity: $\sum_{q_i} w_{i,q_i} = 1$.

Viewed in m-dimensional space, the joint source density $\mathrm{p}(\mathbf{s})$ formed by the product of the one-dimensional MOG's (3.5) is itself a MOG. Its collective hidden states

$$\mathbf{q} = (q_1, \ldots, q_m) \tag{3.6}$$

consist of all possible combinations of the individual source states q_i. Each state $\mathbf{q}$ corresponds to an m-dimensional Gaussian density whose mixing proportions $\mathbf{w_q}$, mean $\boldsymbol{\mu_q}$ and diagonal covariance matrix $\mathbf{V_q}$ are determined by those of the constituent source states,

$$\mathbf{w_q} = \prod_{i=1}^{m} w_{i,q_i} = w_{1,q_1} \ldots w_{m,q_m}, \;\; \boldsymbol{\mu_q} = (\mu_{1,q_1}, \ldots, \mu_{m,q_m}), \;\; \mathbf{V_q} = \mathrm{diag}(\nu_{1,q_1}, \ldots, \nu_{m,q_m}). \tag{3.7}$$

Hence we have

$$\mathrm{p}(\mathbf{s}|\theta) = \prod_{i=1}^{m} \mathrm{p}(\mathrm{s}_i|\theta_i) = \sum_{\mathbf{q}} \mathrm{w}_{\mathbf{q}} \mathcal{G}(\mathbf{s} - \boldsymbol{\mu}_q, \mathbf{V}_{\mathbf{q}}), \tag{3.8}$$

where the Gaussians factorize, $\mathcal{G}(\mathbf{s} - \boldsymbol{\mu}_q, \mathbf{V}_{\mathbf{q}}) = \prod_i \mathcal{G}(s_i - \mu_{i,q_i}, \nu_{i,q_i})$, and the sum over collective states $\mathbf{q}$ represents summing over all the individual source states, $\sum_{\mathbf{q}} = \sum_{q_1} \cdots \sum_{q_m}$.

3.1.2 Sensor Model

With the generative model in Eq(3.1) combined with the source model Eq(3.8) and the noise model Eq(3.2), we deduce that

$$\mathrm{p}(\mathbf{x}|\mathbf{s}) = \mathcal{G}(\mathbf{x} - \mathbf{A}\mathbf{s}, \boldsymbol{\Lambda}). \tag{3.9}$$

It is important to emphasise that the IF generative model is probabilistic, it describes the distribution of the unobserved sources and observed sensor signals rather than the actual signals $\mathbf{s}$ and $\mathbf{x}$. This model is fully described by the joint density of the state $\mathbf{q}$, the sources $\mathbf{s}$ and the observed data $\mathbf{x}$,

$$\mathrm{p}(\mathbf{q}, \mathbf{s}, \mathbf{x}|\Omega) = \mathrm{p}(\mathbf{q})\, \mathrm{p}(\mathbf{s}|\mathbf{q})\, \mathrm{p}(\mathbf{x}|\mathbf{s}), \tag{3.10}$$

it follows that

$$\mathrm{p}(\mathbf{x}|\Omega) = \sum_{\mathbf{q}} \int d\mathbf{s}\, \mathrm{p}(\mathbf{q})\, \mathrm{p}(\mathbf{s}|\mathbf{q})\, \mathrm{p}(\mathbf{x}|\mathbf{s}) = \sum_{\mathbf{q}} \mathrm{p}(\mathbf{q})\, \mathrm{p}(\mathbf{x}|\mathbf{q}), \tag{3.11}$$

where, thanks to the Gaussian forms, the integral over the sources can be performed analytically to yield

$$\mathrm{p}(\mathbf{x}|\mathbf{q}) = \mathcal{G}(\mathbf{x} - \mathbf{A}\boldsymbol{\mu}_q, \mathbf{A}\mathbf{V}_{\mathbf{q}}\mathbf{A}^T + \boldsymbol{\Lambda}). \tag{3.12}$$

3.2 Learning the IF model

We choose the Kullback-Leibler (KL) distance as an error function to measure the difference between our model sensor density $\mathrm{p}(\mathbf{x}|\Omega)$ Eq(3.11) and the observed density $\mathrm{p}^o(\mathbf{x})$

$$\mathcal{E}(\Omega) = \int d\mathbf{x}\, \mathrm{p}^o(\mathbf{x}) \log \frac{\mathrm{p}^o(\mathbf{x})}{\mathrm{p}(\mathbf{x}|\Omega)} = -\,\mathrm{E}[\log \mathrm{p}(\mathbf{x}|\Omega)] - H_{\mathrm{p}^o} \tag{3.13}$$

where the operator E performs averaging over the observed $\mathbf{x}$.

The error Eq(3.13) consists of two terms: the first is the negative log-likelihood of the sensors given the model parameters Ω; the second H_{p^o}, the sensor entropy is independent of Ω. Thus minimising $\mathcal{E}$ is equivalent to maximising the likelihood of the data with respect to the model.

A straightforward way to minimise the error Eq(3.13) would be to use the gradient-descent method where, starting from random values, the parameters are incremented at each iteration by a small step in the direction of the gradient $\partial\mathcal{E}/\partial\Omega$. However, this results in rather slow learning. Instead we shall employ the expectation-maximisation approach to develop an efficient algorithm for learning the IF model.

We implement the EM algorithm by noting that, in addition to the likelihood of the observed sensor data Eq(3.13), one may consider the likelihood of the complete data, composed of both the observed data and the missing data, i.e., the unobserved source signals and states. Each iteration then consists of two steps:

(E) Calculate the expected value of the complete-data likelihood, given the observed data and the current model: $\mathcal{F}(\Omega', \Omega)$

(M) Minimise $\mathcal{F}(\Omega', \Omega)$ with respect to Ω to obtain the new parameters.

It was proved in [2] that the new parameters obtained from the M-step satisfy

$$\mathcal{E}(\Omega) \leq \mathcal{F}(\Omega', \Omega) \leq \mathcal{F}(\Omega', \Omega') = \mathcal{E}(\Omega'), \tag{3.14}$$

showing that the current EM step does not increase the error.

We obtain the learning rules in terms of the old parameters Ω' for the mixing matrix and noise covariance:

$$\begin{aligned} \mathbf{A} &= \mathrm{E}\,\mathbf{x}\langle \mathbf{s}^T|\mathbf{x}\rangle(\mathrm{E}\langle \mathbf{s}\mathbf{s}^T|\mathbf{x}\rangle)^{-1}, \\ \mathbf{\Lambda} &= \mathrm{E}\,\mathbf{x}\mathbf{x}^T - \mathrm{E}\,\mathbf{x}\langle \mathbf{s}^T|\mathbf{x}\rangle\mathbf{A}^T, \end{aligned} \tag{3.15}$$

whereas the rules for the source MOG parameters are

$$\begin{aligned} \mu_{i,q_i} &= \frac{\mathrm{E}\,\mathrm{p}(q_i|\mathbf{x})\langle s_i|q_i,\mathbf{x}\rangle}{\mathrm{E}\,\mathrm{p}(q_i|\mathbf{x})}, \\ \nu_{i,q_i} &= \frac{\mathrm{E}\,\mathrm{p}(q_i|\mathbf{x})\langle s_i^2|q_i,\mathbf{x}\rangle}{\mathrm{E}\,\mathrm{p}(q_i|\mathbf{x})} - \mu_{i,q_i}^2, \\ w_{i,q_i} &= \mathrm{E}\,\mathrm{p}(q_i|\mathbf{x}). \end{aligned} \tag{3.16}$$

Where $\langle \mathbf{s}|\mathbf{x}\rangle$ is a $n \times 1$ vector denoting the conditional mean of the sources given the sensors; the $n \times n$ matrix $\langle \mathbf{s}\mathbf{s}^T|\mathbf{x}\rangle$ is the source covariance conditioned on the sensors. E performs averaging over the observed $\mathbf{x}$.

In addition, we maintain the variance of each source at unity by performing the following scaling transformation at each iteration.

$$\begin{aligned} &\sigma_j^2 = \sum_{q_j}^{n_j} w_{j,q_j}(\nu_{j,q_j} + \mu_{j,q_j}) - (\sum_{q_j}^{n_j} w_{j,q_j}\mu_{j,q_j})^2, \\ &\mu_{j,q_j} \to \frac{\mu_{j,q_j}}{\sigma_j}, \quad \nu_{j,q_j} \to \frac{\nu_{j,q_j}}{\sigma_j}, \quad \mathbf{A}_{i,j} \to \mathbf{A}_{i,j}\sigma_j. \end{aligned} \tag{3.17}$$

3.3 Recovering the sources

We now have all the IF parameters, we can reconstruct the sources, but a perfect reconstruction is only possible when there is no noise ($\mathbf{\Lambda} = 0$), and the mixing matrix is invertible. Then, the estimated sources are $\hat{\mathbf{s}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{x}$.

However, generally, an estimate of the sources must be made. In the following, we discuss two of them: least mean squares (LMS) and maximum a-posteriori probability

(MAP). Both are non-linear functions of the data, but each satisfies a different optimality criterion. For Gaussian sources, they both reduce to the same estimator of ordinary Factor Analysis (FA), but for non-Gaussian sources, LMS and MAP estimators differ and neither has an *a priori* advantage over the other.

3.3.1 LMS Estimator

The least mean square estimator minimises $\mathrm{E}(\hat{\mathbf{s}}-\mathbf{s})^2$ and is equal to the conditional mean of the sources given the observed sensors,

$$\hat{\mathbf{s}}^{LMS}(\mathbf{x}) = \langle \mathbf{s}|\mathbf{x} \rangle = \int d\mathbf{s}\, \mathbf{s}\, \mathrm{p}(\mathbf{s}|\mathbf{x}, \Omega). \tag{3.18}$$

This is equal to

$$\hat{\mathbf{s}}^{LMS}(\mathbf{x}) = \sum_{\mathbf{q}} \mathrm{p}(\mathbf{q}|\mathbf{x})(\mathbf{A_q}\mathbf{x} + b_{\mathbf{q}}), \tag{3.19}$$

where $\mathbf{A_q} = \Sigma_{\mathbf{q}}\mathbf{A}^T\mathbf{\Lambda}^{-1}$, $b_{\mathbf{q}} = \Sigma_{\mathbf{q}}{\mathbf{V_q}}^{-1}\mu_{\mathbf{q}}$, and $\sigma_{\mathbf{q}} = (\mathbf{A}^T\mathbf{\Lambda}^{-1}\mathbf{A} + {\mathbf{V_q}}^{-1})^{-1}$.

3.3.2 MAP Estimator

The maximum a-posteriori (MAP) estimator finds the sources values that maximise the source posterior density $\mathrm{p}(\mathbf{s}|\mathbf{x})$. For a given observation $\mathbf{x}$, maximising the posterior is equivalent to maximising the joint density $\mathrm{p}(\mathbf{s}, \mathbf{x})$ or its logarithm

$$\hat{\mathbf{s}}^{MAP}(\mathbf{x}) = \arg\max_{\mathbf{s}}[\log \mathrm{p}(\mathbf{x}|\mathbf{s}) + \sum_{i=1}^{m} \log(s_i)]. \tag{3.20}$$

A simple way to compute this estimator is iterate the method of gradient ascent, for each data vector $\mathbf{x}$, with

$$\delta\hat{\mathbf{s}} = \eta\mathbf{A}^T\mathbf{\Lambda}^{-1}(\mathbf{x} - \mathbf{A}\hat{\mathbf{s}}) - \eta\,\phi(\hat{\mathbf{s}}), \tag{3.21}$$

where η is the learning rate and

$$\phi(s_i) = -\frac{\partial \log \mathrm{p}(s_i)}{\partial s_i} = -\sum_{q_i=1}^{n_i} \mathrm{p}(q_i|s_i) \frac{s_i - \mu_{i,q_i}}{\nu_{i,q_i}}. \tag{3.22}$$

A good initialisation is given by the pseudo inverse relation $\hat{\mathbf{s}}(\mathbf{x}) = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{x}$.

Chapter 4

Towards a Practical ICA Approach

To perform ICA on MEG data (which are the real world data we use in section 5.3), we must first estimate the number of sources. Then we can use IFA or FastICA to recover the sources. The issues we shall address in this chapter are:

1. Determine the number of sources of the MEG data. This is important first step, if n is overestimated, we may have source signals that will over fit the real sources, the sources may be badly recovered and we will have some additive signals, mixture of the original sources. If n is underestimated some source signals won't obviously be recovered.

2. The choice of the ICA algorithm when we face many sources: FastICA or IFA?

3. The pre-processing of the data that we shall use when we face sensor noise.

4.1 Choosing the number of sources

4.1.1 Introduction

A simple way to encode input patterns is to suppose that each input can be well-approximated by a linear combination of component vectors, where the amplitudes of the vector are modulated to match the input. For a given training set, the most appropriate set of component vectors will depend on how we expect the modulation levels

to behave and how we measure the distance between the input and its approximation. These effects can be captured by a generative model that specifies a distribution $p(\mathbf{s})$ over modulation levels $\mathbf{s} = (s_1, \ldots, s_n)^T$ and a distribution $p(\mathbf{x}|\mathbf{s})$ over sensors $\mathbf{x} = (x_1, \ldots, x_m)^T$ given the modulation levels. The linear combination is given by

$$E[\mathbf{x}|\mathbf{s}] = \mathbf{A}\mathbf{s} \tag{4.1}$$

where each column of $\mathbf{A}$ is a component vector. $\mathbf{A}$ is a $n \times m$ matrix.

In a maximum likelihood approach, PCA, FA and ICA can be viewed as maximum likelihood estimate of such a model, where we assume that the appropriate modulation levels (the sources in the ICA case) are independent and the overall sensor noise is given by the sum of the individual sensor noises.

If we choose

$$p(s_i) = \mathcal{G}(s_i, 1), \;\; p(x_j|\mathbf{s}) = \mathcal{G}(x_j - \mathbf{A}_j\mathbf{s}, \Lambda_j), \tag{4.2}$$

where $\mathbf{A}_j$ is the j^{th} row of $\mathbf{A}$ and Λ_j is the j^{th} element of the diagonal noise covariance matrix $\mathbf{\Lambda}$, then, if $n = m$ and $\mathbf{\Lambda} = 0$, maximum likelihood estimate of $\mathbf{A}$ gives PCA, if $n \leq m$ and all the Λ_j are the same then it performs probabilistic PCA (PPCA) and if $n \leq m$ and the Λ_j can have different values, it performs FA.

In the case of ICA, the sources can have any distribution, usually there is no noise and A is square ($n = m$). In the literature we find the term of probabilistic ICA (PICA) in [31] when the Gaussian noise covariance matrix is isotropic and $n \leq m$ or IFA [1] when this matrix is a full one and $n \leq m$.

We need a fast and simple technique for deciding the number of sources. As the PCA and FA models are very close to the ICA model (though the sources are assumed Gaussians in PCA and FA and they must be non-Gaussian in ICA), we may use PCA and FA tools to approximate the noise and the number of sources provided that the assumption about the source densities doesn't weaken the estimates too much.

4.1.2 Recovering the noise with FA

Why are we interested in recovering the noise level?

We will see that this noise can be interpreted as a reconstruction error which is a function of n, so it can help us to choose the number of sources of the data (see section 4.1.3) . Moreover if we have an estimate of n, we can build some synthetic data with the same noise to see how confident we can be with this estimate. Finally it will tell us whether it's necessary to use ICA tools which take into account the noise, since they are often slow and may not converge when the noise level is small (for example IFA and most of the algorithms using an Expectation Maximisation algorithm to optimise the model).

In order to benchmark the estimation of the noise with FA, we will use 4 sources (figures 4.1 and 4.2) with variance equal to 1, mixed into 8 sensors (with variance set to 1) and with different levels of noise. We fixed $T = 1600$ because most of our MEG dataset uses this number of observations. Moreover FA needs sufficiently many examples for the noise model to converge (usually more than 500 is sufficient if the noise is not too large). Finally, we know the following limitation for FA [33]:

$$n \leq \frac{1}{2}(2m + 1 - \sqrt{8m + 1}) \tag{4.3}$$

To perform the factor analysis, we use the Expectation Maximisation algorithm for FA described in [11]. The recovered noise covariance matrix is diagonal so we want to see how well uncorrelated Gaussian noise with zero mean is estimated.

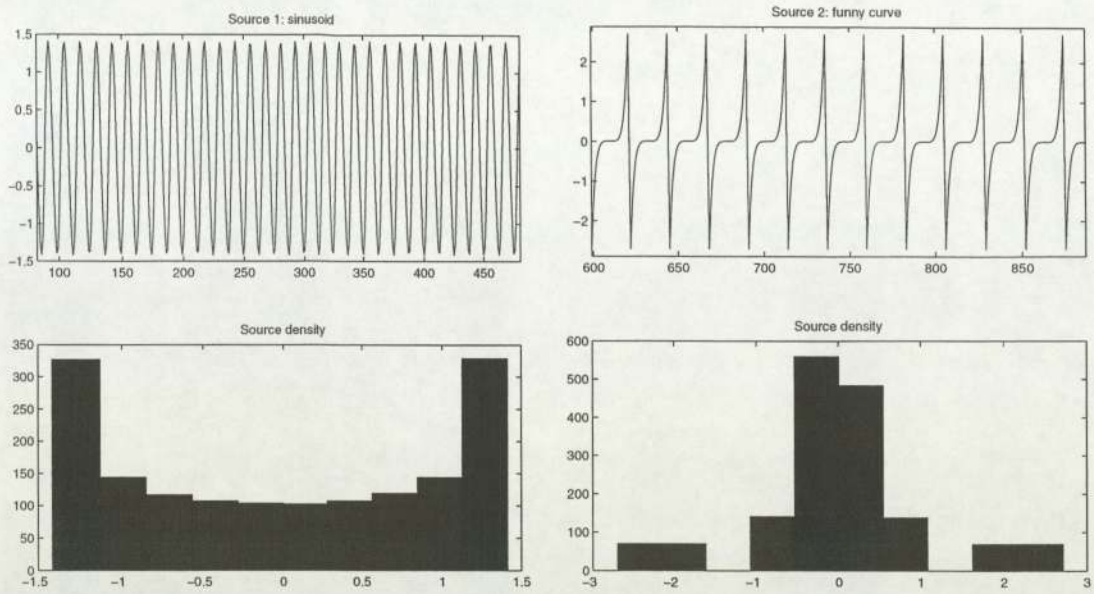

Figure 4.1: The sinusoid and funny curve sources

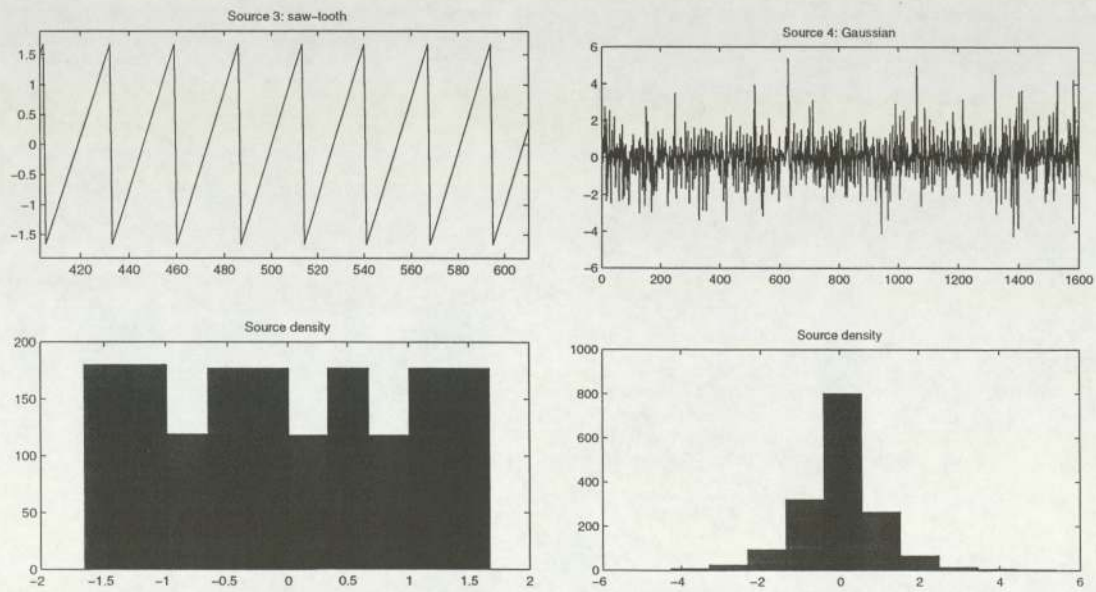

Figure 4.2: The saw-tooth and the Gaussian sources

For a really low noise level

$$\mathbf{\Lambda_{diag}} = [0.001\ 0.002\ 0.003\ 0.004\ 0.005\ 0.001\ 0.002\ 0.003]^T$$
$$\rightarrow \hat{\mathbf{\Lambda}}_{\mathbf{diag}} = [0.0010\ 0.0011\ 0.0036\ 0.0036\ 0.0051\ 0.0012\ 0.0016\ 0.0033]^T,$$

where $\mathbf{\Lambda_{diag}}$ is the diagonal of $\mathbf{\Lambda}$ and $\hat{\mathbf{\Lambda}}_{\mathbf{diag}}$ is the estimated $\mathbf{\Lambda_{diag}}$.

In the same way, for larger noise levels

$$\mathbf{\Lambda_{diag}} = [0.01\ 0.02\ 0.03\ 0.04\ 0.05\ 0.01\ 0.02\ 0.03]^T$$
$$\rightarrow \hat{\mathbf{\Lambda}}_{\mathbf{diag}} = [0.0018\ 0.0206\ 0.0252\ 0.0380\ 0.0480\ 0.0166\ 0.0192\ 0.0338]^T,$$

$$\mathbf{\Lambda_{diag}} = [0.1\ 0.2\ 0.3\ 0.4\ 0.5\ 0.1\ 0.2\ 0.3]^T$$
$$\rightarrow \hat{\mathbf{\Lambda}}_{\mathbf{diag}} = [0.0989\ 0.2031\ 0.2135\ 0.3732\ 0.5033\ 0.0886\ 0.2005\ 0.3634]^T.$$

And even for a noise with variance close to the signal variance, we get:

$$\mathbf{\Lambda_{diag}} = [1\ 0.9\ 1\ 0.9\ 0.8\ 1\ 0.9\ 0.8]^T$$
$$\rightarrow \hat{\mathbf{\Lambda}}_{\mathbf{diag}} = [1.1431\ 0.8195\ 0.4555\ 1.0445\ 0.8314\ 0.9885\ 0.7203\ 0.8274]^T.$$

We can conclude that using FA on mixtures of independent sources can give a good approximation of uncorrelated Gaussian sensor noise. However as n, m and T can change its accuracy (if n is close to m or if T is not large enough), one can suggest when we face a real ICA problem to check this estimator on synthetic data with the same n, m and T as in the problem.

4.1.3 Using the recovered noise in order to reduce the dimensionality

What happens if the mixing matrix is non square i.e. $n \neq m$ and we don't know n? Usually people use the eigenvalues of the covariance matrix and cut the dimensionality where the eigenvalues become too small (meanwhile they try to keep the maximum of

information about the data, they reduce the dimension of the data). However when the data are noisy, this method can give completely a wrong estimate of n.

On the other hand with FA we obtain an estimate of the noise covariance matrix. Then if we sum the diagonal terms of the noise covariance matrix recovered by FA, for each dimension we have the overall sensor noise which can be interpreted as a reconstruction error. When the number of sources is underestimated, the recovered overall sensor noise is larger than the real one, as it should explain the large reconstruction error (we can't recover the sensors as well as with the right n since the sources are independent), this overall sensor noise will decrease until it equals to n and then it should become stationary as the reconstruction shouldn't be better with more sources. Therefore, we can choose the dimension n to be the point at which this reconstruction error becomes stationary, we will modestly call this method the Clapier criterion.

We illustrate this approach using 15 sources created by 15 mixtures of 3 Gaussians (each one with random centres, variances) mixed into 25 sensors. Sensor noise is added and we will compare this method to eigenvalue analysis (see figures 4.3, 4.4 and 4.5). We emphasize that the statistics of the sources are not Gaussian.

One of the problem with using eigenvalues is that the corner of the curve on realistic data often occurs for too small values of n, even if the underlying number of sources is larger. That's why, one can prefer to use the Clapier criterion to estimate n, as its corner is closer to n. However, both methods are quite subjective and require visual inspection of the graphs. We would rather use an estimator which optimise a certain cost function, in order to make the decision of the number of sources automatic.

4.1.4 A PCA tool to estimate n

Following this idea of taking PCA or FA tools in ICA, one may want to use one tool from PCA which has been proven computationally efficient and accurate to estimate

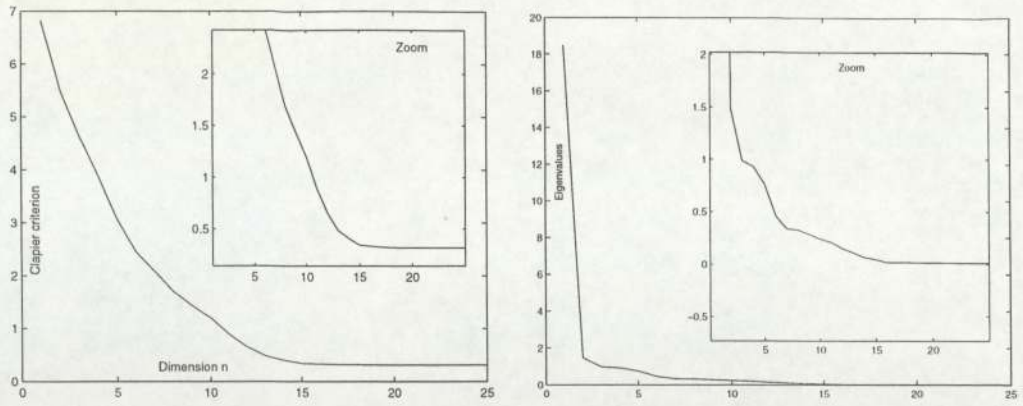

Figure 4.3: choice of dimensionality: 15 sources, 25 sensors, an isotropic sensor noise with variance = 0.1; on the left, the Clapier criterion, on the right, eigenvalues of the noise covariance matrix of the sources.

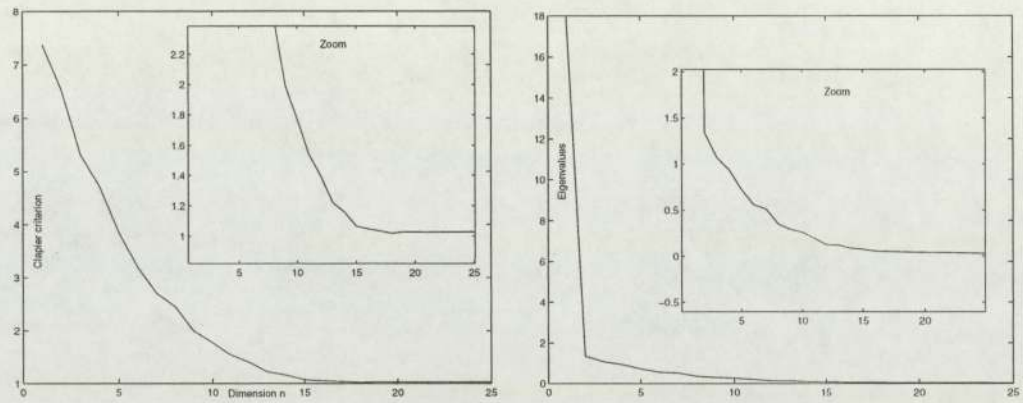

Figure 4.4: choice of dimensionality: 15 sources, 25 sensors, an isotropic sensor noise with variance = 0.3; on the left, the Clapier criterion, on the right, eigenvalues the noise covariance matrix of the sources.

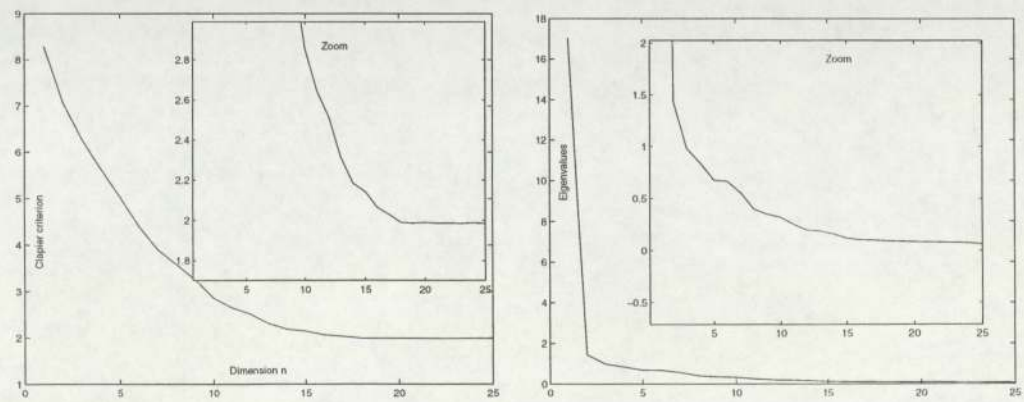

Figure 4.5: choice of dimensionality: 15 sources, 25 sensors, an isotropic sensor noise with variance = 0.6; on the left, the Clapier criterion, on the right, eigenvalues of the noise covariance matrix of the sources.

n in ICA.

In [23], P. Minka introduced an estimator which is a Laplace approximation of the probability of the sensors given n in a Bayesian probabilistic PCA model. In PCA, it provides a simple and fast criterion for choosing the dimensionality:

$$\mathrm{p}(\mathbf{x}|n) \approx \mathrm{p}(\mathbf{U}) \left(\prod_{j=1}^{n} l_j \right)^{-T/2} \hat{v}^{-T(m-n)/2} (2\pi)^{(r+n)/2} |\mathbf{H}_Z|^{-1/2} T^{-n/2} \tag{4.4}$$

where l_j are the eigenvalues of the sensors in a descending order, $r = mn - n(n+1)/2$ and

$$\mathrm{p}(\mathbf{U}) = 2^{-n} \prod_{j=1}^{n} \Gamma((m-j+1)/2) \pi^{-(m-j+1)/2}, \hat{v} = \frac{\sum_{j=n+1}^{m} l_j}{m-n}, \tag{4.5}$$

$$|\mathbf{H}_Z| = \prod_{i=1}^{n} \prod_{j=i+1}^{m} (\hat{l}_j^{-1} - \hat{l}_i^{-1})(l_j^{-1} - l_i^{-1})^T, \hat{\mathbf{l}} = [l_1, \ldots, l_{m-n+1}, \hat{v}, \ldots, \hat{v}] \tag{4.6}$$

It has been also discussed in [31] that it is a fast and accurate criterion in comparison with other PCA and ICA criterion applied to ICA (see figure 4.6). However this criterion doesn't seem to be accurate when m is large (see table 4.2).

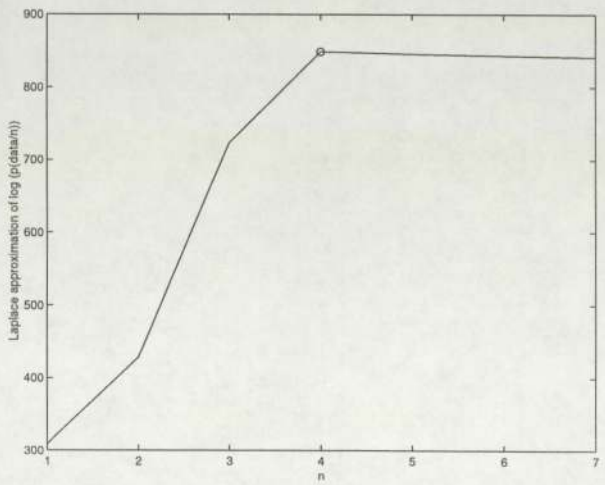

Figure 4.6: Choice of dimensionality using the Laplace approximation introduced in [31]: the graph correspond to the values of the Laplace approximation applied on 8 mixtures of the 4 signals described in figures 4.1 and 4.2 for different number of sources n. The maximum correspond to the right number of underlying dimensions, $n = 4$.

The problem comes from the sensor signals with small eigenvalues (smaller than

the noise level), they take too much importance. One way to solve this problem is to include only those eigenvalues greater than the variance of the largest sensor noise, as it seems difficult to recover signals with a smaller variance than the noise. In this section we will show that the following method improves this criterion when we suppose that n satisfies Eq(4.3).

1. centre the sensors and set their variance to 1,
2. compute the eigenvalues of the sensors and let $\mathbf{l}$ be the vector of the eigenvalues in a decreasing order,
3. do FA on the sensors with n the number of sources satisfying Eq(4.3) and let a be the largest value of the noise covariance matrix,
4. set the elements of $\mathbf{l}$ that are less than a to 0, let d be the number of elements of $\mathbf{l}$ that are greater than 0, and apply Eq(4.4) for $n = \{1, \ldots, d\}$ using the new $\mathbf{l}$,
5. choose n that maximises Eq(4.4).

To compare the two algorithms, we take sources created by mixture of 3 Gaussians, we mix them, set their variance to 1 and add different level of noise (see tables 4.1 and 4.2).

The old method gives completely inaccurate results, but our proposed modification seems to work quite well and to be robust to noise.

4.2 IFA versus FastICA

If FastICA does well when the noise is small, we expect from IFA that it will recover a better signal when the noise increases, thanks to the non-linear estimators (see section 3.3). Moreover, IFA doesn't use the highest eigenvalues of the sensors to reduce the dimensionality, which is one of our criteria.

Mixing matrix	σ^2			
	0.001	0.01	0.1	0.3
5×25	$5.4_{0.52}$	$5.6_{0.96}$	$5.6_{0.7}$	$5.2_{0.42}$
12×25	$12.1_{0.32}$	12_0	$11.9_{0.32}$	$11.8_{0.42}$
18×25	18_0	18_0	$17.5_{0.71}$	$15.6_{1.42}$
10×90	$11.1_{0.73}$	$10.7_{0.82}$	$11_{0.82}$	$11.3_{0.95}$
50×90	50_0	50_0	50_0	50_0
75×90	75_0	75_0	$73.4_{1.08}$	$68.2_{2.15}$
10×150	$12.1_{0.88}$	$12.5_{1.84}$	$12.5_{1.27}$	$11.8_{0.92}$
20×150	$20.4_{0.52}$	$21.2_{0.92}$	$20.6_{0.7}$	$20.6_{0.7}$
80×150	80_0	80_0	80_0	80_0
130×150	130_0	130_0	$126.9_{1.29}$	$116.7_{2.99}$

Table 4.1: Estimation of n with the Laplace approximation introduced by P. Minka [23], using only the eigenvalues greater than the variance of the greatest sensor noise estimated by FA. Each experiment has been run 10 times, we give the mean and the standard deviation of the criterion over these experiments. We notice that when n is close to m, the estimator is not robust to noise.

Mixing matrix	σ^2			
	0.001	0.01	0.1	0.3
5×25	20	12	22	21
12×25	14	21	23	19
18×25	20	19	21	19
10×90	50	55	38	34
50×90	54	54	50	55
75×90	75	75	75	76
20×150	106	130	27	33
80×150	141	80	86	80
130×150	130	132	130	130

Table 4.2: Estimation of n with the Laplace approximation introduced by P. Minka [23]. We ran the experiment only once since the results were very inaccurate. We just want to show how the modification of the algorithm improve the criterion (see table 4.1).

4.2.1 Toy experiments

In this experiment we used a 8×4 mixing matrix, the sources are as shown in figures 4.1 and 4.2, $T = 5000$ the sources had variance set to 1, and we added an isotropic sensor noise of variance 0.1. We recovered the sources in figure 4.7 after 5000 iterations of the EM algorithm (each source was modelled by a mixture of 2 Gaussians), using the LMS estimator (MAP was equivalent):

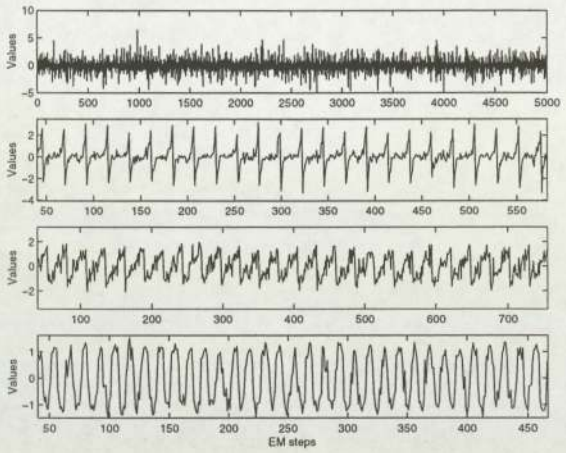

Figure 4.7: The sources of figures 4.1 and 4.2 with $T = 5000$, have been mixed with a 8×4 mixing matrix and the sensor noise has a variance of 0.1. We recovered the sources with IFA.

The recovered noise covariance matrix is:

$$\hat{\Lambda} = \begin{bmatrix} \mathbf{0.0982} & 0.0003 & -0.0022 & 0.0008 & -0.0005 & 0.0012 & 0.0002 & 0.0033 \\ 0.0003 & \mathbf{0.0974} & -0.00019 & -0.0004 & 0.0009 & -0.0011 & -0.0025 & 0.0032 \\ -0.0022 & -0.0019 & \mathbf{0.1016} & -0.0020 & 0.0017 & 0.0004 & -0.0017 & -0.0003 \\ 0.0008 & -0.0004 & -0.0020 & \mathbf{0.1002} & 0.0001 & -0.0001 & -0.0010 & 0.0014 \\ -0.0005 & 0.0009 & 0.0017 & 0.0001 & \mathbf{0.1008} & -0.0006 & -0.0001 & -0.0022 \\ 0.0012 & -0.0011 & 0.0004 & -0.0001 & -0.0006 & \mathbf{-0.1010} & 0.0007 & -0.0006 \\ 0.0002 & -0.0025 & -0.0017 & -0.0010 & -0.0001 & 0.0007 & \mathbf{0.0998} & -0.0010 \\ 0.0033 & 0.0032 & -0.0003 & 0.0014 & -0.0022 & -0.0006 & -0.0010 & \mathbf{0.1026} \end{bmatrix}$$

In order to measure the mixing matrix convergence, we use

$$\mathbf{J} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{A}^o = \mathbf{A}^\dagger\mathbf{A}^o, \tag{4.7}$$

where $\mathbf{A}^o$ is the original mixing matrix and $\mathbf{A}^\dagger$ the pseudo-inverse of $\mathbf{A}$. We note that for the correct estimate $\mathbf{A} = \mathbf{A}^o$, $\mathbf{J}$ becomes the unit matrix $\mathbf{I}$. Thus we plot the element of $\mathbf{J}$ and we should observe the diagonal elements converging towards 1 and the others converging towards 0 since the sources had unit variance.

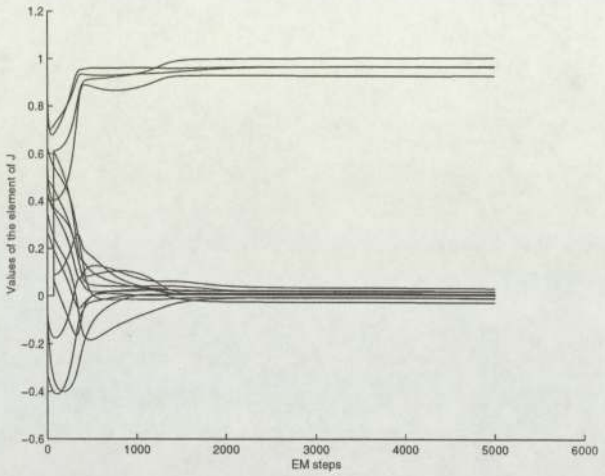

Figure 4.8: We recovered the signals with IFA and we observe the mixing matrix convergence toward the original. We observe that the diagonal elements are converging towards 1 and the others towards 0.

To observe the convergence of the estimated noise covariance matrix Λ towards the true one Λ^o, we use the Kullback-Leibler (KL) distance (Cover and Thomas 1991) between the corresponding noise densities (figure 4.9)

$$K_n = \int du \mathcal{G}(u, \Lambda^o) \log \frac{\mathcal{G}(u, \Lambda^o)}{\mathcal{G}(u, \Lambda)} = \frac{1}{2}\mathrm{Tr}\Lambda^{-1}\Lambda^o - \frac{n}{2} - \frac{1}{2}\log|\det \Lambda^{-1}\Lambda^o|. \tag{4.8}$$

And then to observe the convergence of the error Eq(3.13), at each iteration we compute the part of the error which depends on the IF parameters: $-E[\log(\mathrm{p}(\mathbf{x}|\Omega)]$ (figure 4.10).

We see that all the IF parameters are converging, but we have to compare this

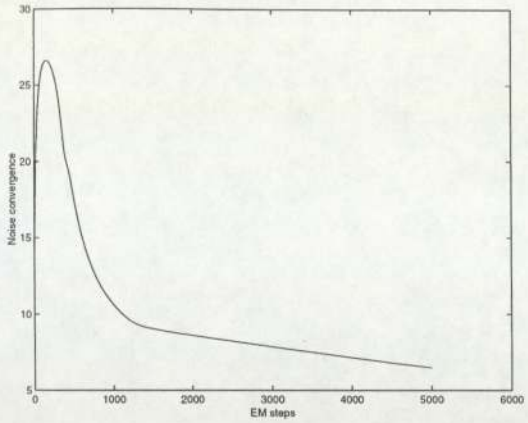

Figure 4.9: Noise covariance matrix convergence in IFA

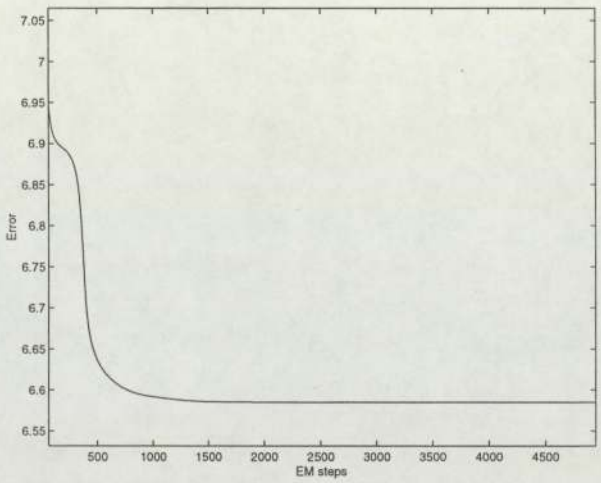

Figure 4.10: Evolution of the error in IFA computed from the part of Eq(3.13) which depend on the IF parameters

result with the sources recovered by FastICA (figure 4.11). We note that the sine wave recovered by IFA is a little bit more regular than with FastICA, but the difference between the two recovered sources is not very large.

Moreover, we have to be aware that this experiment took two days of computation with IFA and less than 2 minutes with FastICA[1]. We can guess that for 10 sources modelled by mixture of 2 Gaussians, it will last 128 days ($2 \times 2^{10}/2^4$) for the same number of EM steps. The computational cost doesn't seem worth the small difference in the results.

[1] On a UltraSPARC 5: 400 MHz UltraSPARC IIi processor

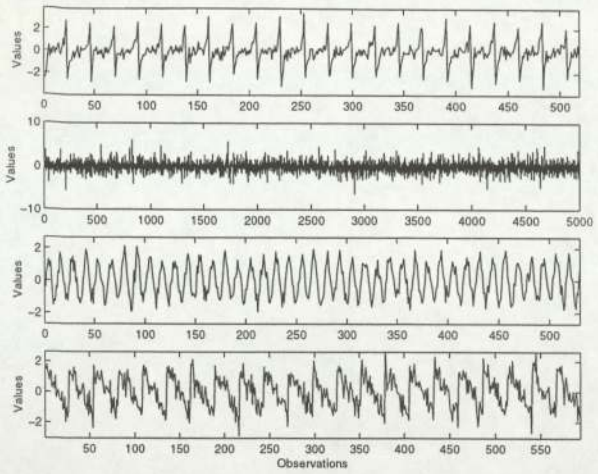

Figure 4.11: Recovered sources with FastICA: compare with those recovered by IFA figure 4.7. We reduced the dimension pre-whitening the data with PCA and keeping the 4 first signals with the largest eigenvalues.

4.2.2 Remarks concerning IFA

In further experiments we noticed that using IFA gives good results when the noise level is sufficient (the number of EM steps before convergence of the parameters increases when the sensor noise decreases, see figure 4.12) but:

- the noise covariance matrix convergence needs a large data set (greater than 2000 examples),
- and the most important point is that it becomes intractable when the number of sources is large. In the process of the EM algorithm or any gradient descent algorithms, it is necessary to calculate the conditional distribution over all the configurations of the source states. If we represent each source by n_i Gaussians, we have $\prod_{i=1}^{n} n_i$ posterior computed other all the combination of source state at each EM step.

Improved initialisation of the parameters that decreases the number of EM steps, and thus the computational time, doesn't solve the problem which is still intractable for large number of sources.

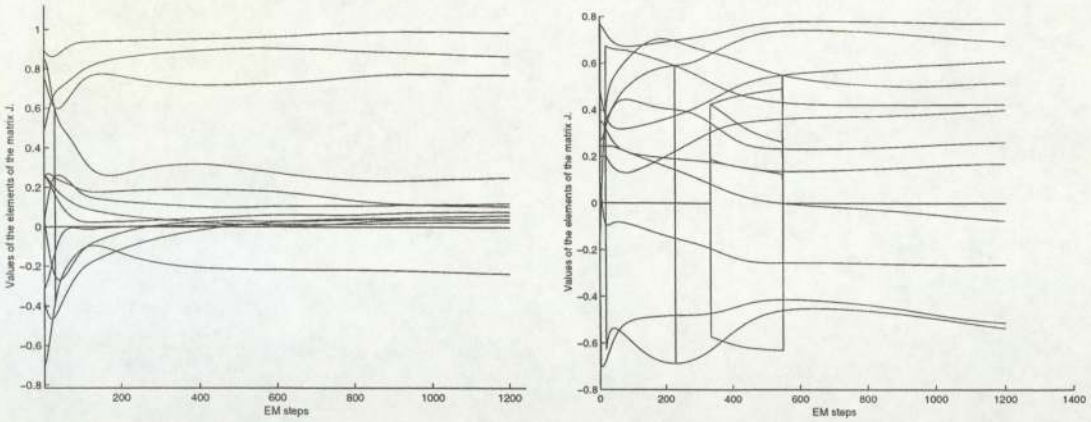

Figure 4.12: Convergence of the mixing matrix with an isotropic sensor noise of variance 0.1 on the left and 0.01 on the right: the sensors are generated by the sources of figures 4.1 and 4.2, and mixed by a 8×4 mixing matrix, finally sensor noise is added. We show the convergence of the mixing matrix using the elements of matrix $\mathbf{J}$ (see Eq(4.7)). We notice on the first 1200 steps of IFA the data with high sensor noise converge better than the one with low sensor noise. A noise free IFA algorithm was derived in [2] for the case with low sensor noise.

One solution is proposed in [2] with a variational approximation of IFA and a data-independent approximation of the variational approximation (see appendix C).

The data-independent approximation gave results worse than FastICA on the experiment of section 4.2.1 (some of the recovered sources did not converge toward the original sources) and was still computationally demanding in comparison with FastICA.

The variational approximation doesn't seem to be worth to use in our case as it is very intensive, at each EM step, we have to solve a linear system with $\sum_{i=1}^{n} n_i$ unknown until convergence of the variational parameters, for each data point (see appendix C).

ICA methods based on a Bayesian framework have been recently introduced [3], [20], [28], [24]. In [3] and [28] a prior has been added over the variances in the mixing matrix which aims to determine the number of underlying components n, which is known as Auto Relevance Determination (ARD) [22]. These models are theoretically interesting, (in [28] they take the flexible source model of IFA [2]), however they remain either computationally intensive or at least they don't help us to choose the

dimensionality, since the ARD is really sensitive to sensor noise.

For all these reasons, we decided to use FastICA as the ICA algorithm to separate our data.

4.3 Choice of whitening method

4.3.1 Introduction

In FastICA and many other ICA algorithms [5], [8], we often first pre-process the data to make it uncorrelated (see appendix A.2). This is often done by PCA, however in many real world problems, we have to face sensor noise; thus it's natural to think about using FA instead, and this approach has been discussed in [14]. Moreover, pre-whitening the data with an EM algorithm for FA [11] will allow us to reduce the dimensionality without using only those sensors with the largest eigenvalues.

After centring the sensors, we estimate the following FA model:

$$\mathbf{x} = \mathbf{A_{FA}}\mathbf{f} + \epsilon \tag{4.9}$$

where $\mathbf{A_{FA}}$ is a $m \times n$ matrix, $\mathbf{f}$ is normally distributed $\mathbf{f} \sim N(0, \mathbf{I_n})$, ϵ is normally distributed $\epsilon \sim N(0, \mathbf{\Sigma})$, $\mathbf{\Sigma}$ is diagonal and $\mathbf{f}$ and ϵ are mutually independent.

Let $\mathbf{W_{FA}}$ be the pseudo inverse of $\mathbf{A_{FA}}$, then we now want to separate $\mathbf{z}$ with FastICA:

$$\mathbf{z} = \mathbf{W_{FA}}\mathbf{x}, \tag{4.10}$$

and $\mathbf{z}$ is sphered data. The pseudo-inverse is not unique, so we choose the one which minimise the expected norm $\mathrm{E}[(\mathbf{x} - \mathbf{A_{FA}}\mathbf{z})^T\mathbf{\Sigma}^{-1}(\mathbf{x} - \mathbf{A_{FA}}\mathbf{z})]$, which is the difference between $\mathbf{x}$ and the reconstructed observation $\mathbf{A_{FA}}\mathbf{z}$ measured with $\mathbf{\Sigma}^{-1}$. It was shown

in [30] that it corresponds to:

$$\mathbf{W_{FA}} = (\mathbf{A_{FA}}^T \mathbf{\Sigma}^{-1} \mathbf{A_{FA}})^{-1} \mathbf{A_{FA}}^T \mathbf{\Sigma}^{-1}. \tag{4.11}$$

This also helps to reduce the sensor noise when we reconstruct the data from independent components (see [14]).

4.3.2 Toy experiments

In this toy experiment we mixed a low frequency sine wave, high frequency sine wave and a Gaussian (figure 4.13) with a 10×3 mixing matrix and we added an isotropic sensor noise of variance 0.1.

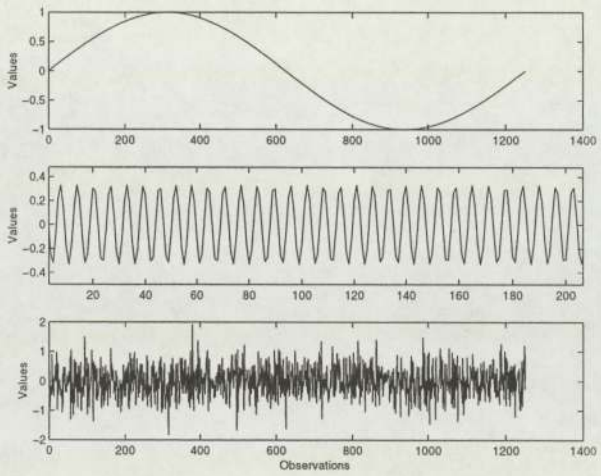

Figure 4.13: The sources: a low frequency sine wave, a high frequency sine wave and a Gaussian

The recovered sources with FastICA pre-processed by FA and PCA are shown in figure 4.14, the high frequency sine wave is more regular and the low one is less noisy using FA.

However, often we need a large noise to observe a difference between FA and PCA as a preprocessing of ICA, and FA is constrained by Eq(4.3). Most of the time, when the noise is not that large, we can use PCA to pre-whiten the data.

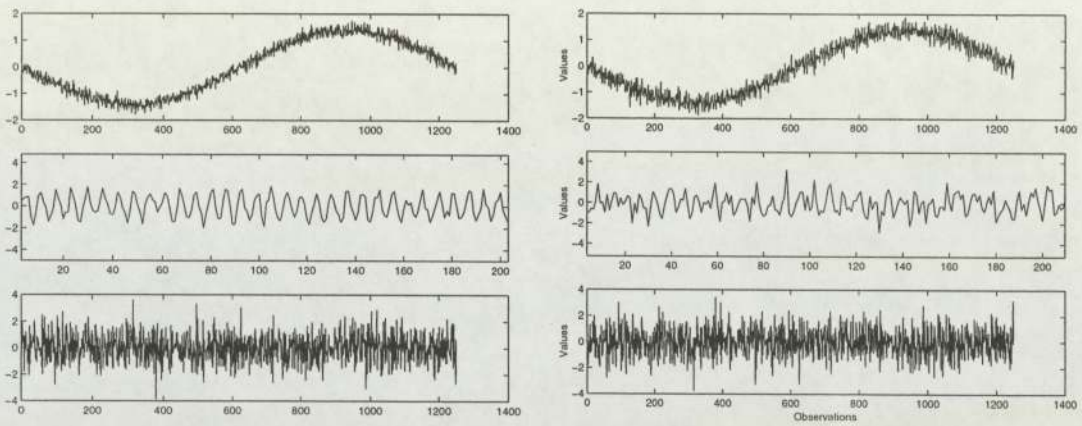

Figure 4.14: FastICA on 10 mixtures of the sources (figure 4.13), on the left pre-whiten by FA, on the right PCA.

Chapter 5

A solution to solve the ICA problem

5.1 The proposed method

On the basis of the analysis of chapter 4 we propose the following method to apply ICA in the noisy case when n is supposed to satisfy Eq(4.3).

1. Centre $\mathbf{x}$.
2. Determine the number of sources using a PCA method: the corrected Laplace approximation from P. Minka introduced in section 4.1.4 or if the noise is too large and this estimator is biased, use the Clapier criterion (see 4.1.3).
3. Whiten the data using an EM algorithm for Factor Analysis: it will whiten the sensors and give an estimate of the noise.
4. Separate the signals with FastICA.

5.2 Application to cricket songs

One major issue in bioacoustic recognition of animal sounds is to perform species counting. For example, cricket songs differ greatly between species, so we hope to be

able to count the number of species from a single recording. It is known that the distribution of the cricket song is super-Gaussian. We emphasize that the song of each specie of cricket is independent from any other specie of cricket, that's why we expect to recover the song of each specie and not of each cricket. In this section we propose a modus operandi to perform the separation process.

1. Record the songs of the crickets using directional microphones, each one oriented in a specific direction (see figure 5.1).

2. Perform the method described in section 5.1 on the recorded signals to obtain the number of sources (number of cricket species), and the sources (the original song of each species).

Each coefficient in the mixing matrix will depend on how far the crickets from the same species are from a given microphone, how loud they are singing, and how many there are.

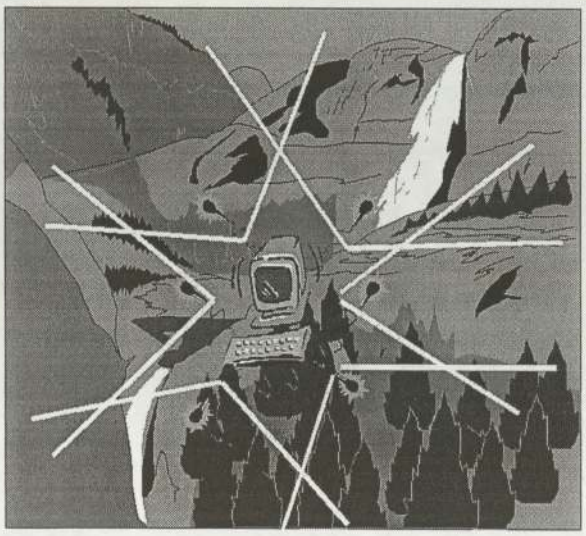

Figure 5.1: Modus operandi: in order to record the mixture of the crickets songs, we take directional microphones which are all situated at the same point but oriented in different directions. We must take enough microphones (m) in order that the number of species n satisfies Eq(4.3).

To demonstrate the feasibility of this approach, we modelled the recorded signals by mixing songs from 4 different species of crickets with a 10×4 mixing matrix and

added noise with variance 0.1.

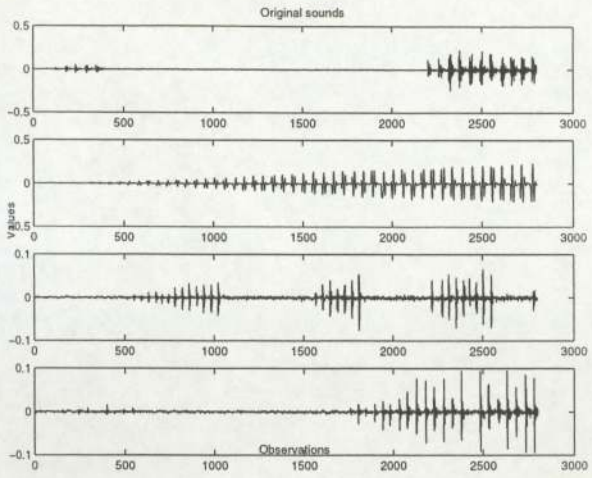

Figure 5.2: Original cricket songs: the sources

Applying the method of section 5.1, we obtained $n = 4$ and the following recovered sources:

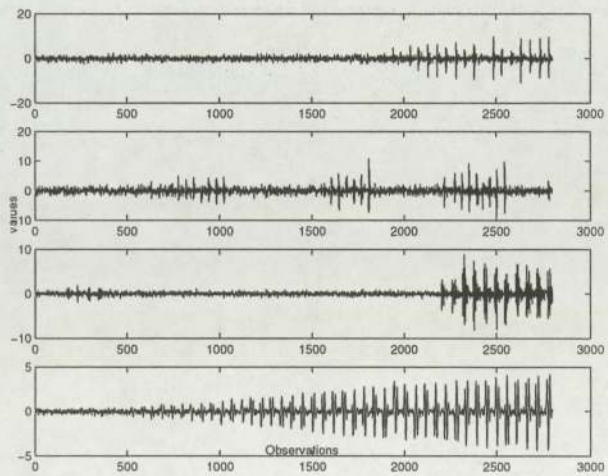

Figure 5.3: The sources in figure 5.2 have been mixed with a 10×4 mixing matrix, we added a sensor noise with variance 0.1 and tried to recover the sources estimating first n with the method proposed in section 4.1.4, then pre-whitening the sensors with FA and then separating the signals with FastICA.

This results are encouraging to apply the method of section 5.1 to real world data, for example, MEG.

5.3 Application to single- and multi-channel MEG

In this section, we performed ICA using FastICA 2.1, the non-linear function was "pow3": $g(u) = u^3$ (as it gave good results on toy experiments) and the approach "deflation" (which correspond to the one-unit version of FastICA).

5.3.1 Multi channel MEG

Introduction

When using a magnetoencephalographic (MEG) recording, as a research or clinical tool, the investigator faces the problem of extracting the essential features of the neuromagnetic signals in the presence of artifacts. Actually, the signals contains a lot of environmental noise. We can categorize the noise into two major categories, one is called the artifacts and the other is the sensory noise. The artifacts include all the source signals we are not interested in: the noise from electric power supply, the earth magnetism, breathing and the brain activity. The amplitude of the sensor noise may be higher than that of the brain signals, and the artifacts may look like pathological signals. Since the sources of the brain activity are assumed to be localised on different points and that the signal spread linearly in the brain, recently researchers have tried to apply ICA to MEG signals ([14, 17]) to solve these problems.

In this section we will use test data which consists of multi-channel MEG data recorded at the Wellcome Trust Laboratory for MEG studies at Aston University, on the 151 channel Omega MEG system (CTF Systems Inc.). The data is down-sampled to a suitable rate of 200 samples/sec and mean corrected in the rows and columns. The example used here consists of over 6 seconds of 150 channel MEG data collected from the cortex of a young girl with a known tumour in the right temporal region of the brain. In the presence of the tumour it is expected that slow-wave theta activity ($\sim$5Hz) should be measured over the right temporal region of the scalp.

Results

The first task is to estimate the number of sources n. The criterion given in section 4.1.4 gives $n = 42$. To check the robustness of this criterion we added some sensor noise and tried again to find the number of sources (table 5.3.1). When we add a small level of noise, the estimate remains identical and for larger noise it gives a lower estimate which depends on the level of noise. This encourages us to think that 42 is a good estimate of the number of sources which appear to be possible on the graph of the eigenvalues and on the Clapier criterion (figure 5.4) as it belongs to the range of values where both curves are making a corner .

Experiment #	σ^2			
	0.001	0.01	0.1	0.3
#1	42	43	27	23
#2	43	42	28	22
#3	39	39	29	23
#4	42	41	28	22

Table 5.1: Estimation of the number of underlying components of the 150 channel MEG: we add different level of sensor noise to the MEG data and then try to recover n. With low level of noise, the criterion still gives $n \sim 42$, and when the noise increases, the estimated n decreases. Therefore, 42 seems to be a good approximation of the number of sources since the criterion remains unchanged for a low level of noise.

Given the number of sources we now want to recover the supposedly independent underlying sources. We can compare the results of FastICA pre-whitened by FA and by PCA; they are not that different (see appendix D.1). The results are a little bit better (heartbeat smoother, 50 Hz more regular) with PCA pre-processing; this is due to the low level of sensor noise. The noise matrix from FA had mean 0.0035 and variance $7.6e-5$ on the centred and sphered MEG data taking 133 sources in order to satisfy Eq(4.3). The EM algorithm doesn't converge very well for such a low level of noise.

Moreover, since we know that the EM algorithm often underestimates the level of noise, the criterion may be affected and give a too large estimate of the number of

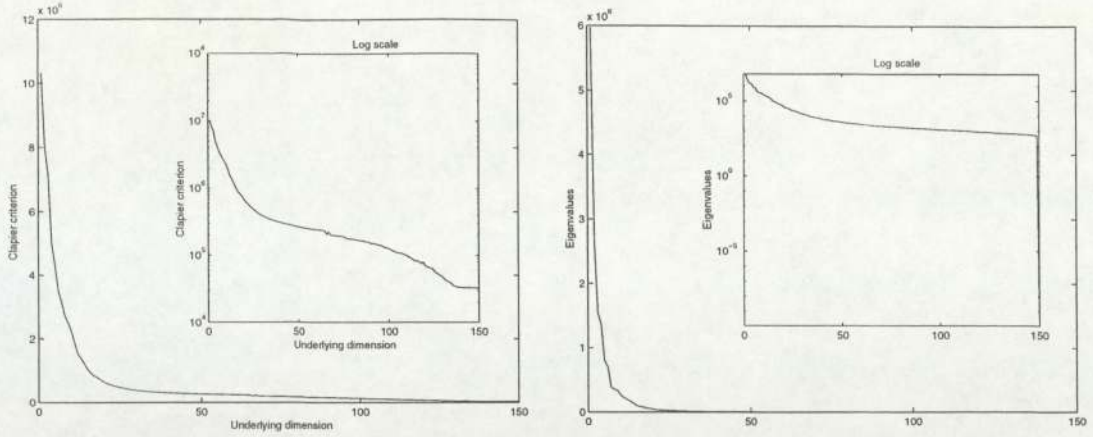

Figure 5.4: On the left, the Clapier criterion (see section 4.1.3) on the MEG, on the right the eigenvalues. We can notice that the number of sources ($n = 42$) recovered by the corrected Laplace approximation of section 4.1.4 belongs to the range of possible n.

sources as it could take into account too many irrelevant eigenvalues (those with a variance smaller than the variance of the noise).

In fact, if we look at figure D.5 in appendix D.1 we notice that for $n = 24$ the heartbeat and the 50 Hz signals are still well recovered, that may let us think that the right number of sources is less than 42. When we go further, we can notice that the heartbeat is no longer well recovered for smaller number of sources ($n \leq 20$), which tell us that n must be greater than 20.

Performing ICA on the sensors doesn't recover the 5Hz signal expected from the tumour, we must look at other techniques to recover the sources. Another way to perform ICA on MEG data is to apply it on single-channel MEG using an embedding matrix, which has been shown to be able to isolate components from single-channel data in [17].

5.3.2 Single channel MEG

Introduction

From a single channel MEG signal ($x(t)$) one can construct a series of delay vectors, where the state of the system at time t, $\mathbf{X}(t)$, is given by:

$$\mathbf{X}(t) = [x(t-\tau), x(t-2\tau), \ldots, x(t-(m-1)\tau)] \tag{5.1}$$

where τ is the lag and m is the number of lags or the embedding dimension. If we denote by T the number of delay vectors, once we have define appropriate values for m and τ (cf [17], $m = 90$ and $\tau = 1$ in this section), we can represent the embedding matrix

$$\mathbf{X} = \begin{bmatrix} x_t & x_{t+\tau} & \cdots & x_{t+N\tau} \\ x_{t+\tau} & x_{t+2\tau} & \cdots & x_{t+(N+1)\tau} \\ \vdots & \vdots & \ddots & \vdots \\ x_{t+(m-1)\tau} & x_{t+m\tau} & \cdots & x_{t+(m+N-1)\tau} \end{bmatrix} \tag{5.2}$$

in a convenient spanning basis provided by ICA. We assume that the m row vectors of $\mathbf{X}$ are linearly generated by n independent vectors. However in [17], the choice of the n independent components of interest was made by hand from m independent components recovered by FastICA. In this section we propose to use the criterion proposed in section 4.1.4 to make a first analysis of the m components in order to reduce them to the estimated n.

Once the independent components have been obtained, we project them back to the measurement space of those components such that:

$$\mathbf{Y}^i = \mathbf{a}_i \mathbf{s}_i^T, \tag{5.3}$$

where $\mathbf{s}_i$ is the i^{th} independent component ($i = 1, \ldots, n$), $\mathbf{a}_i$ the corresponding column

of the mixing matrix $\mathbf{A}$ and $\mathbf{Y}^i$ the resulting matrix representing the source $\mathbf{s}_i$ in $\mathbf{X}$. From $\mathbf{Y}^i$ it's now possible to extract the time series $y_i(t)$ by averaging the rows of $\mathbf{Y}^i$, in order to unembed the time series:

$$y_i(t) = \frac{1}{m} \sum_{k=1}^{m} \mathbf{Y}^i_{k,(t+k-1)}, \tag{5.4}$$

for $t = 1, \ldots, T$ where $\mathbf{Y}^i_{k,(t+k-1)}$ is the element of $\mathbf{Y}^i$ indexed by row k and column $t + k - 1$.

Results

Using the previous method to find the components of a single channel, we must be aware that we will recover many similar components displaced one from the other, this is due to the way we construct the embedding matrix. But at the end they will correspond to the same signal in the measurement space. That's why we have more sources than the original number of sources using this method. This will be illustrated in the next section.

Toy experiment:

In this experiment we mixed the 3 signals in figure 4.14 (a low frequency sine wave, a high frequncy sine wave and a Gaussian) in order to obtain the single signal in figure 5.5. The Gaussian represents the added sensor noise and will has a small variance of 0.01.

Then we tried to recover the components of the embedding matrix constructed with the same parameters as in section 5.3.2 with $n = 2$ (see figure 5.6) which is the right number of sources and $n = 3$ (see figure 5.7) which gives better reconstruction of the sources when we put the embedding components in the measurement space: figure 5.8.

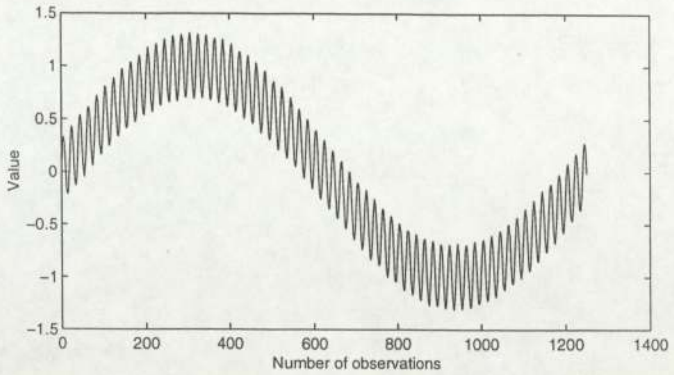

Figure 5.5: The single signal generated by a mixture of 3 sources of figure 4.14, the Gaussian represents the sensor noise and has a small variance.

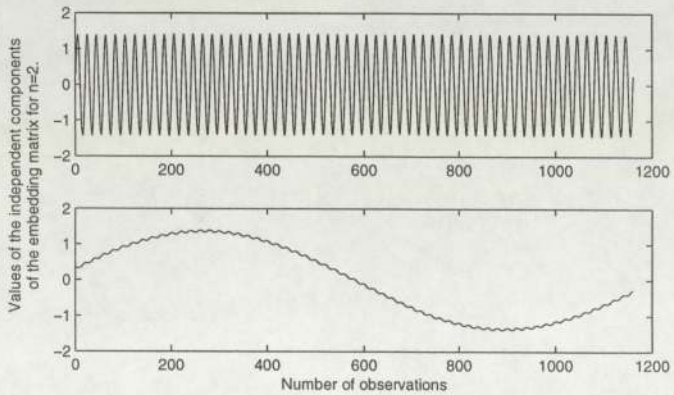

Figure 5.6: The components of the embedding matrix of the single signal with $n = 2$. We notice that the low frequency sine wave is noisy, that's why we prefer the components of figure 5.7 where this component is smoother.

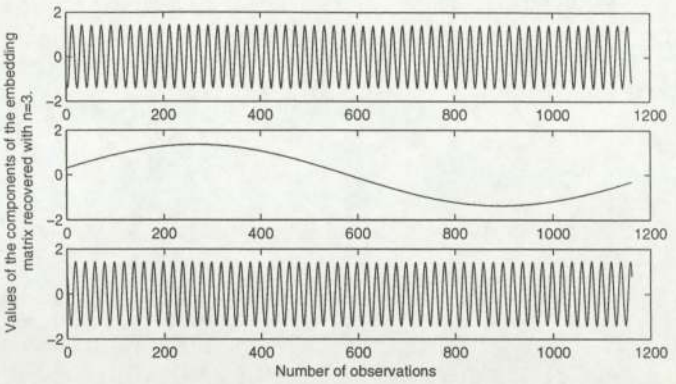

Figure 5.7: The components of the embedding matrix of the single signal with $n = 3$. We notice that there are two components representing the high frequency sine wave. They are not identical as there is a little lag (a quarter of the period of the sine wave: the two components are independent, which won't be the case if the lag was an half of the period of this signal since they would be the same signal up to the sign) between the two signals, but in the measurement space, they reduce to the same signal (see figure 5.8).

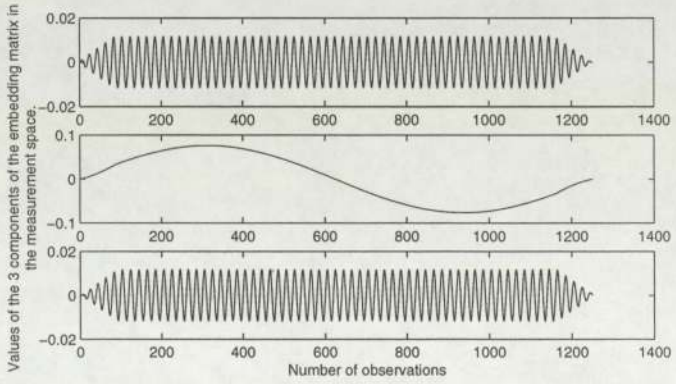

Figure 5.8: The three components of the embedding matrix in the measurement space. We've taken the three components in figure 5.7 to construct three matrices $\mathbf{Y_i}$ and average each one as in equation Eq(5.4) to get the three sources of the embedding matrix in the measurement space. We had to use three sources for the embedding matrix in order to recover the two sources of the single signal.

The conclusion of this toy experiment is that we may need more components to explain the embedding matrix of the single channel MEG than the real number of underlying components.

Single channel-MEG:

We are interested in a single channel MEG recorded from over the right temporal lobe of a child with a known tumour in the right temporal lobe (figure 5.9).

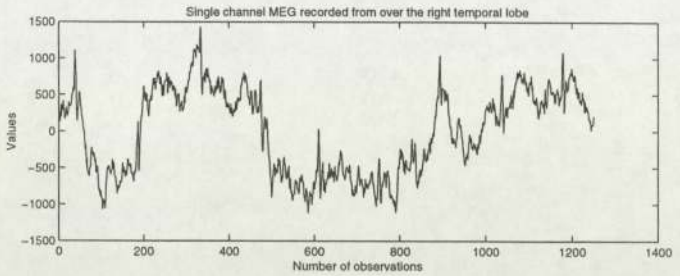

Figure 5.9: Single-channel MEG recorded from over the right temporal lobe of a child with a known tumour in the right temporal lobe.

We used the criterion of section 4.1.4 and estimated the number of underlying components of the embedding matrix of the single channel to be 31. Then we obtained 31 components (using FastICA pre-whitened by PCA as the noise level is really low) of the embedding matrix that we projected back to the measurement space. Many of them correspond to the same signal (see appendix D.2), the MCG activity, the theta band activity or the alpha band activity. We also recovered the base-line shift (see figure 5.10).

Therefore, reducing the dimension of the embedding matrix with the corrected Laplace approximation, we still manage to recover all the interesting components given in [17], this must be a good estimate of the number of sources (of the embedding matrix) since when we reduce the dimension to 25, the MCG activity disappear.

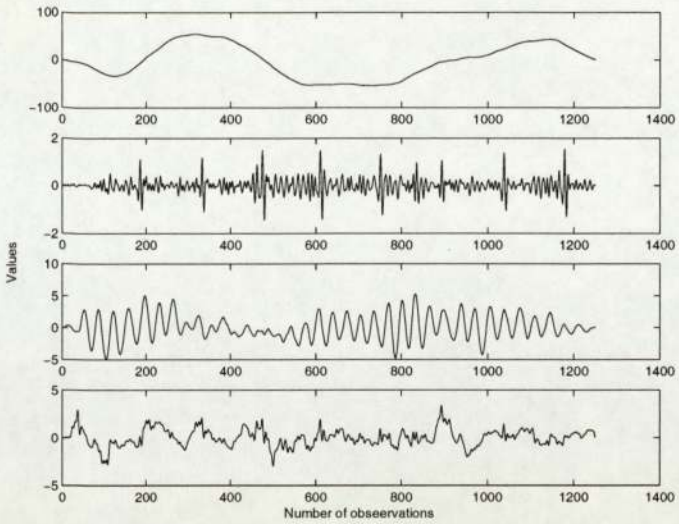

Figure 5.10: The four components of interest of the embedding matrix in the measurement space: from up to down, the global shape of the single-channel MEG, the MCG activity (heartbeat), the alpha band activity and the theta band activity (see figure 5.11) , which we assume generated by the tumour.

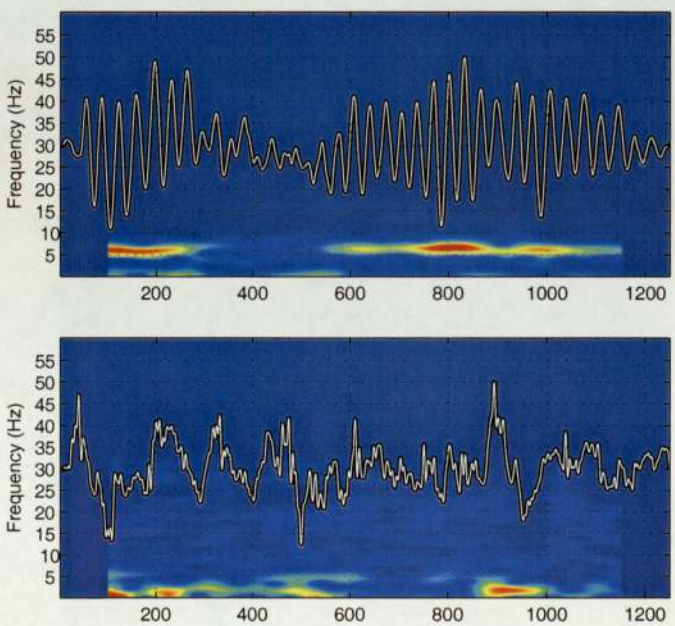

Figure 5.11: The frequencies of the recovered alpha band and theta band activity correspond respectively to 7-8 Hz and 3Hz.

Chapter 6

Conclusion

The starting point of this thesis was the article from H. Attias [2] introducing a new method to perform Independent Component Analysis: Independent Factor Analysis. This algorithm can cope with arbitrary source densities, noise and non-square mixing matrix. Then we wanted to compare this algorithm on a concrete problem, with a more "conventional" algorithm, FastICA [13] which has been proven fast and accurate, but which doesn't take into account the noise and for which the mixing matrix must be square.

The concrete problem, chosen to evaluate algorithms for ICA is the extraction of sources from MEG data, since it has been a really trendy topic in the biosignal analysis community in the past few years ([18], [19], [29] and [12]). In this problem, we face a large number of sensors, the underlying number of components is suppose to be much less than the number of sensors and there is sensor noise. After showing that this problem can't be solved in a reasonable time by IFA, we looked for methods which estimate the number of sources, so that we can reduce the dimension of the sensors keeping the maximum of information before separating the signals with FastICA.

Since ICA and PCA models are similar, we tried to use methods which have been proven efficient for the PCA problem on ICA. On synthetic data, we showed that the method introduced by P. Minka in [23] to estimate the number of sources in PCA, if we choose the eigenvalues carefully, gives an accurate and relatively fast criterion in

the ICA case. Moreover we introduced an algorithm interpreting the recovered sensor noise for a given number of sources as a reconstruction error, which helps in practice in cutting down the dimensionality.

In order to reduce the dimensionality once we have the estimated number of sources, we proposed to use Factor Analysis which gives a better whitening of the data when we face non-isotropic noise or non-Gaussian sources. Finally we applied these tools to a MEG dataset to check the validity of this approach on a concrete problem.

As the MEG data had very few sensor noise, the pre-whitening by FA instead of PCA wasn't necessary. However the corrected Laplace approximation (section 4.1.4) gave a good estimate of the number of sources, a bit overestimated in the multi-channel case. This could be due to the underestimation of the sensor noise by the EM algorithm for FA on our data.

We said that it was impossible to use IFA when we face many sources for computational reasons. However it may be interesting in order to reduce the computational time at each EM step to freeze some parameters (the source densities or the noise model) or to derive an IFA algorithm where the noise covariance matrix will be a diagonal matrix or an isotropic matrix and not a full one as we don't really need such a complicated noise model. We suggest to initialize the parameters using FastICA for the mixing matrix, FA for the noise covariance matrix and finally the k-means algorithm and/or an EM algorithm to estimate the parameters of the sources (the parameters of each mixture of Gaussian) from the sources recovered by FastICA.

A key issue of Independent Component Analysis is the estimation of the number of sources, future work may be oriented toward solutions robust to noise and to large number of sensors. It may be interesting to derive the Laplace approximation of P. Minka for the ICA case, where the sources are not Gaussian.

Bibliography

[1] H. Attias. EM algorithms for independent component analysis. *Neural Network for Signal Processing*, VIII:132–141, 1998.

[2] H. Attias. Independent Factor Analysis. *Neural Computation*, 11(4):803–851, 1999.

[3] C. M. Bishop and N. D. Lawrence. Variational Bayesian Independent Component Analysis. Technical report, Computer Laboratory, University of Cambridge, 2000.

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

[5] J.-F. Cardoso. High-order contrasts for independent component analysis. *Neural Computation*, 11:157–192, 1999.

[6] J.-F. Cardoso and A. Souloumiac. Blind beamforming for non gaussian signals. *IEE Procedings-F*, 140(6):362–370, 1993.

[7] S. Choi and O. Lee. Flexible independent component analysis. *Neural Network for Signal Processing*, VIII:83–92, 1998.

[8] P. Comon. Independent Component Analysis - a new concept? *Signal Processing*, 36:287–314, 1994.

[9] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, 1991.

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

[11] B. J. Frey. Factor analysis using batch and online em. *Internal UW/CS, Adaptive Computation TR-99-2*, 1999.

[12] G. Wübbeler, A. Ziehe, B.-M. Mackert, K.-R Müller, L. Trahms and G. Curio. Independent component analysis of noninvasively recorded cortical magnetic dc-fields in humans. *IEEE Trans. Biomed. Eng.*, 47(5):594–599, 2000.

[13] A. Hyvãrinen. Fast and robust fixed-point algorithms for independent component analysis. *Neural Computing Surveys*, 2:94–128, 1999.

[14] S. Ikeda and K. Toyama. Independent component analysis for noisy data–MEG data analysis. *Neural Networks*, 13:1063–1074, December 2000.

[15] J. H. Friedman. Exploratory projection pursuit. *J. of the American Statistical Association*, 82(397):249–266, 1987.

[16] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. of Computers*, c-23(9):881–890, 1974.

[17] C. J. James and D. Lowe. Extracting information from single channel electromagnetic brain signal. 4^{th} *International Conference "Neural Networks and Expert Systems in Medicine and Healthcare"*, 2001.

[18] K. Kobayashi, C. J. James, T. Nakahori, T. Akiyama and J. Gotman. Isolation of epileptic discharges from unaveraged eeg by independent component analysis. *Electroenceph. clin. Neurophysiol.*, 110:1755–1763, 1999.

[19] L. De Lathauwer, B. De Moor, J. Vandewalle. Fetal electrocardiogram extraction by blind source subspace separation. *IEEE Trans. Biomed. Eng.*, 47(5):567–572, 2000.

[20] H. Lappalainen. Ensemble learning for independent component analysis. *Proceedings of the First International Workshop on Independent Component Analysis*, pages 7–12, 1999.

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

[22] D. J. C. MacKay. Probable Networks and Plausible Predictions - A review of Practical Bayesian Methods for Supervised Neural Networks. *Network: Computation in Neural Systems*, 6:469–505, 1995.

[23] T. P. Minka. Automatic choice of dimensionality for PCA. *Neural Information Processing Systems 13*, 2000.

[24] J. W. Miskin and D. J. C. MacKay. *ICA: Principles and Practice*, chapter Ensemble learning for blind source separation. Cambridge University Press, 2000.

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

[26] A. Papoulis. Probability, random variables and stochastics processes. *Mc Graw-Hill*, 3rd edition, 1991.

[27] B. A. Pearlmutter and L. C. Parra. Maximum likelihood blind source separation: A context-sensitive generalization of ICA. *Advances in Neural Information Processing Systems*, 9:613–619, 1997.

[28] R. Choudrey, W. D. Penny and S. J. Roberts. An Ensemble Learning approach to Independent Component Analysis. *IEEE International Workshop on Neural Networks for Signal Processing, Sydney, Australia*, 2000.

[29] R. Vigário, J. Särelä, V. Jousmäki, M. Hämäläinen and E. Oja. Independent component approach to the analysis of eeg and meg recordings. *IEEE Trans. Biomed. Eng.*, 47(5):589–593, 2000.

[30] C. R. Rao. Linear statistical inference and its applications second edition. *Wiley Series in Probability and Mathematical Statistics*, 1973.

[31] W. D. Penny, S. J. Roberts and R. M. Everson. *ICA: Principles and Practice*, chapter ICA: Model order selection and dynamic source models, pages 299–314. Cambridge University Press, 2000.

[32] 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, 2001.

[33] W. Ledermmann. On the rank of the reducedcorrelational matrix in multiple factor analysis. *Psychometrika*, 2:85–93, 1937.

Appendix A

Preprocessing for ICA

A.1 Centring

The first preprocessing that ICA algorithms use is centring $\mathbf{x}$, i.e. subtracting its mean vector $\mathbf{m} = \mathrm{E}\{\mathbf{x}\}$. This preprocessing is made solely to simplify the ICA algorithms, it does not mean that the mean could not be estimated. After estimating the mixing matrix $\mathbf{A}$ with centred data, we can complete the estimate by adding the mean vector of $\mathbf{s}$ back to the centred estimates of $\mathbf{s}$. The mean vector of $\mathbf{s}$ is given by $\mathrm{A}^{-1}\mathbf{m}$.

A.2 Whitening

Another useful preprocessing strategy in ICA is to first whiten the observed variables. This means that before the application of the ICA algorithm (after centring), we transform the observed vector $\mathbf{x}$ linearly so that we obtain a new vector $\tilde{\mathbf{x}}$ which is white (its components are uncorrelated and their variances equal unity):

$$\mathrm{E}\{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\} = I \tag{A.1}$$

The whitening transformation is always possible. The utility of whitening resides in the fact that the new mixing matrix ($\tilde{\mathbf{x}} = \tilde{\mathbf{A}}\mathbf{s}$) is orthogonal, so instead of having to estimate n^2 parameters that are elements of the original matrix $\mathbf{A}$, we only need to estimate the new orthogonal mixing matrix $\tilde{\mathbf{A}}$ with $n(n-1)/2$ degrees of freedom.

It may also be useful to reduce the dimension of the data at the same time as we do the whitening. Usually we look at the eigenvalues d_j of $\mathrm{E}\{\mathbf{x}\mathbf{x}^T\}$ and discard those that are too small, as is often done in the statistical technique of PCA. This has often the effect of reducing noise. Moreover, dimension reduction prevents over-learning.

Appendix B

Why the sources must be non-Gaussian?

The main restriction to ICA is that the independent components must be non-Gaussian, this the difference with PCA or FA where the sources are supposed to be Gaussian. One visual way to understand that is to assume that the mixing matrix is orthogonal and the s_i are Gaussian. Therefore if we take x_1 and x_2 ($\mathbf{x} = (x_1, \ldots, x_m)^T$), they are Gaussian, uncorrelated and of unit variance. Their joint density is given by:

$$\mathrm{p}(x_1, x_2) = \frac{1}{2\pi} \exp(-\frac{x_1^2 + x_2^2}{2}). \tag{B.1}$$

The distribution of this function (see figure B.1) shows that the density is completely symmetric. Then, it doesn't give any information on the directions of the columns of the mixing matrix **A**, **A** can't be estimated.

More rigorously, one can show that the distribution of any orthogonal transformation of the gaussian (x_1, x_2) has exactly the same distribution as (x_1, x_2) and x_1 and x_2 are independent. Thus, in the case of Gaussian sources, we can only estimate the ICA model up to an orthogonal transformation, **A** is not identifiable. Actually if just one of the sources is Gaussian, the ICA model can be estimated.

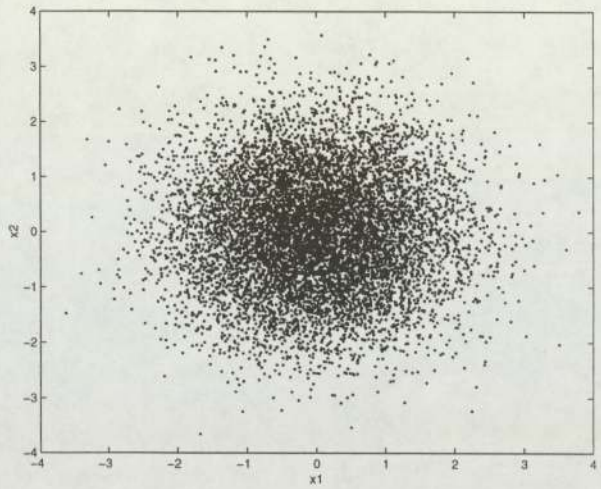

Figure B.1: The multivariate distribution of two independent Gaussian variables.

Appendix C

The Factorized Variational Approximation

In the factorized variational approximation, we assume that even when conditioned on a data vector, the sources are independent. The approximate source density is defined as follows. Given a data vector $\mathbf{x}$, the source x_i at state q_i is described by a Gaussian with a $\mathbf{x}$-independent mean ψ_{i,q_i} and variance ξ_{i,q_i}, weighted by a mixing proportion κ_{i,q_i}. The posterior is defined by the product:

$$\mathrm{p}'(\mathbf{q},\mathbf{s}|\mathbf{x},\tau) = \prod_{i=1}^{n} \kappa_{i,q_i}(\mathbf{x})\mathcal{G}[s_i - \psi_{i,q_i}(\mathbf{x}), \xi_{i,q_i}], \;\; \tau_i = \{\kappa_{i,q_i}, \psi_{i,q_i}, \xi_{i,q_i}\}. \tag{C.1}$$

The variances ξ_{i,q_i} will turn out to be $\mathbf{x}$-independent. Eq(C.1) implies a MOG form for the posterior of s_i:

$$\mathrm{p}'(s_i|\mathbf{x},\tau_i) = \sum_{q_i=1}^{n_i} \kappa_{i,q_i}(\mathbf{x})\mathcal{G}(s_i - \psi_{i,q_i}(\mathbf{x}), \xi_{i,q_i}). \tag{C.2}$$

which is in complete analogy with its prior in Eq(3.5).

The factorized posterior Eq(C.1) is advantageous since it facilities performing in the E-step calculations in polynomial time. Once the variational parameters $\boldsymbol{\tau} = \{\tau_i\}$ have been determined, the data-conditioned mean of the sources, required for the EM

learning rules Eq(3.15) are

$$\begin{aligned}
\langle s_i|\mathbf{x}\rangle &= \sum_{q_i=1}^{n_i} \kappa_{i,q_i}\psi_{i,q_i}, \\
\langle s_i^2|\mathbf{x}\rangle &= \sum_{q_i=1}^{n_i} \kappa_{i,q_i}(\psi_{i,q_i}^2 + \xi_{i,q_i}), \\
\langle s_i s_{j\neq i}|\mathbf{x}\rangle &= \sum_{q_i q_j} \kappa_{i,q_i}\kappa_{j,q_j}\psi_{i,q_i}\psi_{j,q_j},
\end{aligned} \tag{C.3}$$

where those required for the rules Eq(3.16) are given by

$$\mathrm{p}(q_i|\mathbf{x}) = \kappa_{i,q_i}, \quad \langle s_i|q_i,\mathbf{x}\rangle = \psi_{i,q_i}, \quad \langle s_i^2|q_i,\mathbf{x}\rangle = \psi_{i,q_i}^2 + \xi_{i,q_i}. \tag{C.4}$$

We find the estimate of $\boldsymbol{\tau}$ by iteration. First we define the $n \times n$ matrix $\overline{\mathbf{A}}$ by

$$\overline{\mathbf{A}} = \mathbf{A}^T\boldsymbol{\Lambda}^{-1}\mathbf{A}. \tag{C.5}$$

From this matrix we update the variances ξ_{i,q_i}:

$$\xi_{i,q_i} = (\overline{A}_{ii} + \frac{1}{\nu_{i,q_i}})^{-1}. \tag{C.6}$$

The means ψ_{i,q_i} and mixing proportions κ_{i,q_i} are obtained by iterating the following mean-field equations for each data vector $\mathbf{x}$:

$$\sum_{j\neq i}\sum_{q_j=1}^{n_j} \overline{A}_{ij}\,\kappa_{j,q_j}\psi_{j,q_j} + \frac{1}{\xi_{i,q_i}}\psi_{i,q_i} = (\mathbf{A}^T\boldsymbol{\Lambda}^{-1}\mathbf{x})_i + \frac{\mu_{i,q_i}}{\nu_{i,q_i}}, \tag{C.7}$$

$$\log\kappa_{i,q_i} = \log w_{i,q_i} + \frac{1}{2}\left(\log\xi_{i,q_i}^2 + \frac{\mu_{i,q_i}^2}{\xi_{i,q_i}}\right) - \frac{1}{2}\left(\log\nu_{i,q_i}^2 + \frac{\mu_{i,q_i}^2}{\nu_{i,q_i}}\right) + z_i, \tag{C.8}$$

where the z_i are the Lagrange multipliers that enforce the normalisation conditions $\sum_{q_i}\kappa_{i,q_i} = 1$. To solve these equations we first initialise $\kappa_{i,q_i} = w_{i,q_i}$. Eq(C.7) is a linear $(\sum_i n_i) \times (\sum_i n_i)$ system and can be solved for ψ_{i,q_i} using standard methods.

The new κ_{i,q_i} are then obtained form Eq(C.8) via

$$\kappa_{i,q_i} = \frac{e^{\alpha_{i,q_i}}}{\sum_{q_i'} e^{\alpha_{i,q_i'}}}. \tag{C.9}$$

These values are substituted back into Eq(C.7) and the procedure is repeated until convergence.

Data-independent approximation. A simpler approximation results from setting $\kappa_{i,q_i}(\mathbf{x}) = w_{i,q_i}$ for all data vectors $\mathbf{x}$. The means ψ_{i,q_i} can then be obtained from Eq(C.7) in a single iteration for all data vectors at once, since this equation becomes linear in $\mathbf{x}$.

Appendix D

Recovering the underlying components of MEG

D.1 Multi-channel MEG

On the 150 channel MEG described in section 5.3.1, we pre-processed the data with FA taking 42 sources (estimation given by the corrected Laplace approximation described in section 4.1.4) and then applied FastICA to separate the signals (see figures D.1 and D.2).

Next we pre-processed the data with PCA taking the first 42 signals corresponding to the highest eigenvalues and applied Fastica (see figures D.3 and D.4).

D.2 Single-channel MEG

See figure D.6.

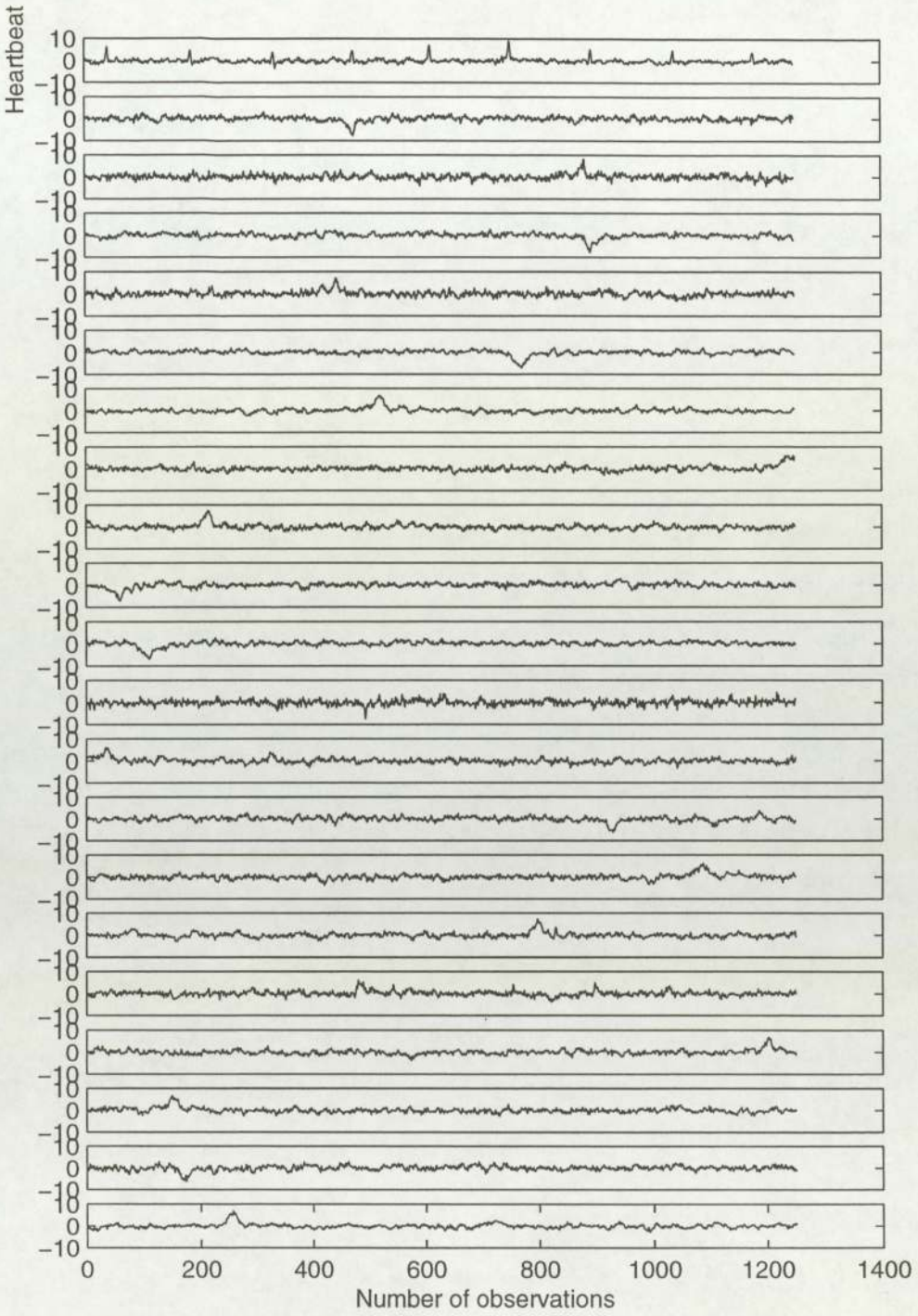

Figure D.1: FastICA pre-processed by FA on the 150 channel MEG data described in 5.3.1 with 42 sources: first 21 components.

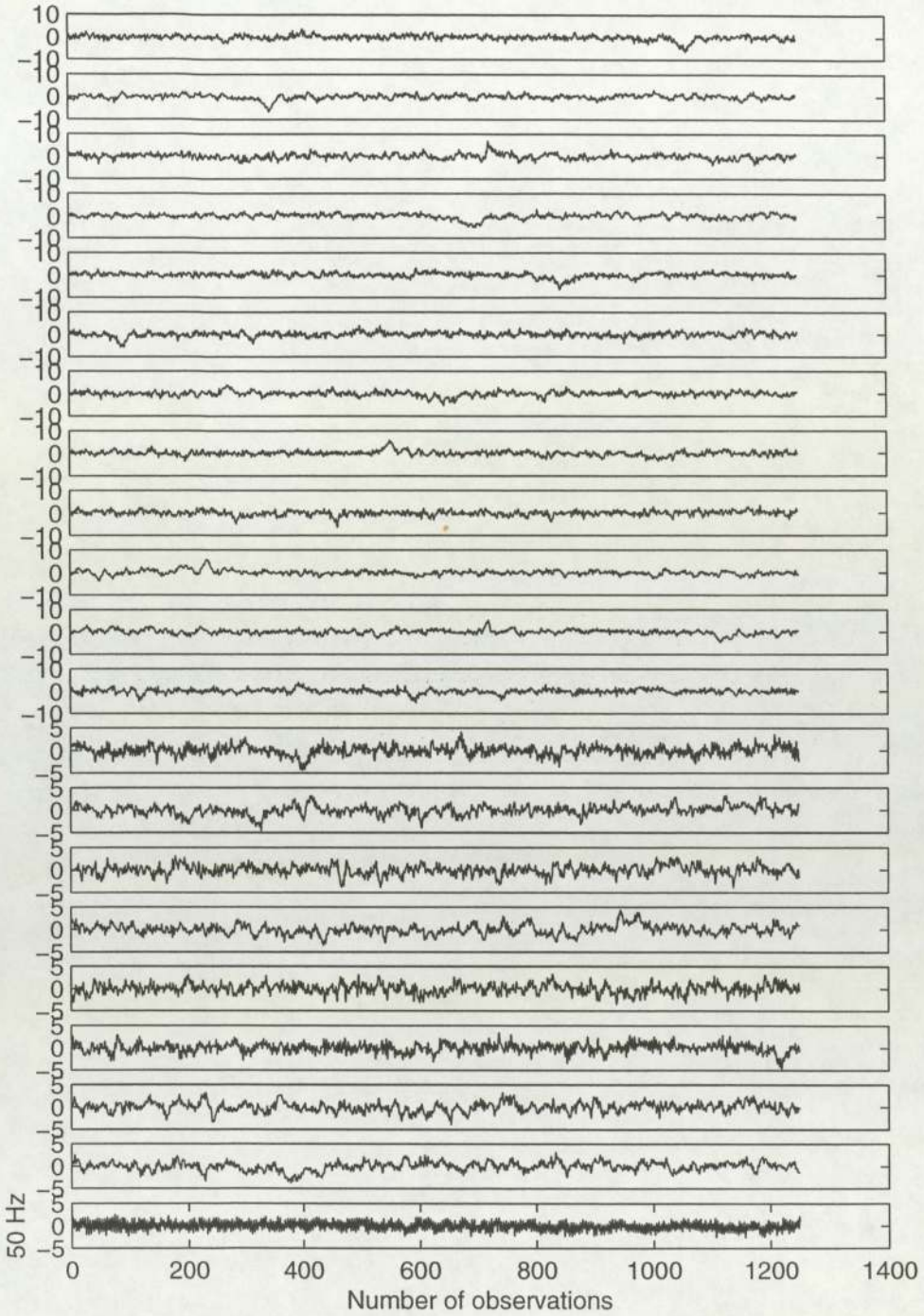

Figure D.2: FastICA pre-processed by FA on the 150 channel MEG data described in 5.3.1 with 42 sources: last 21 components.

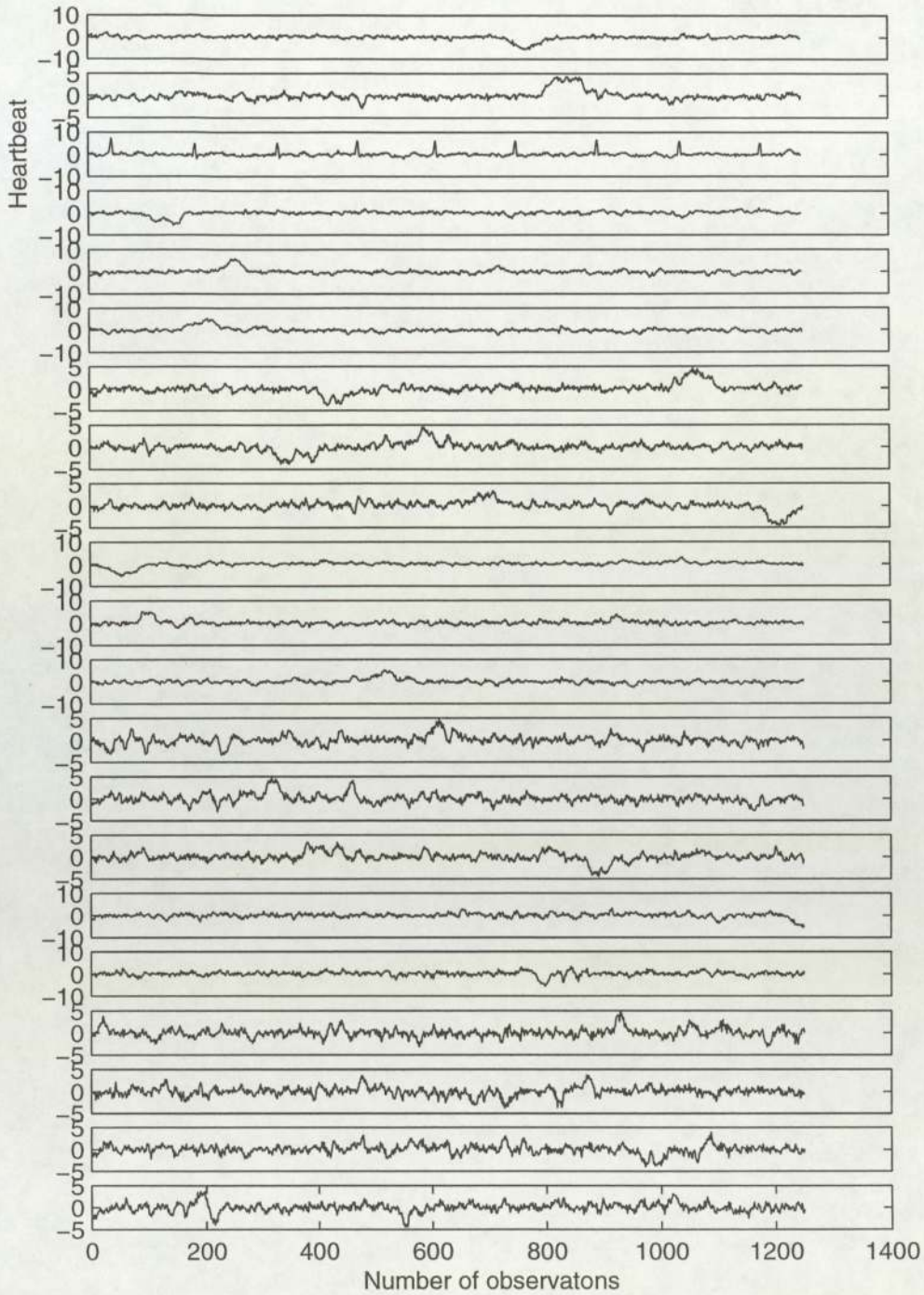

Figure D.3: FastICA pre-processed by PCA on the 150 channel MEG data described in 5.3.1 with 42 sources: first 21 components.

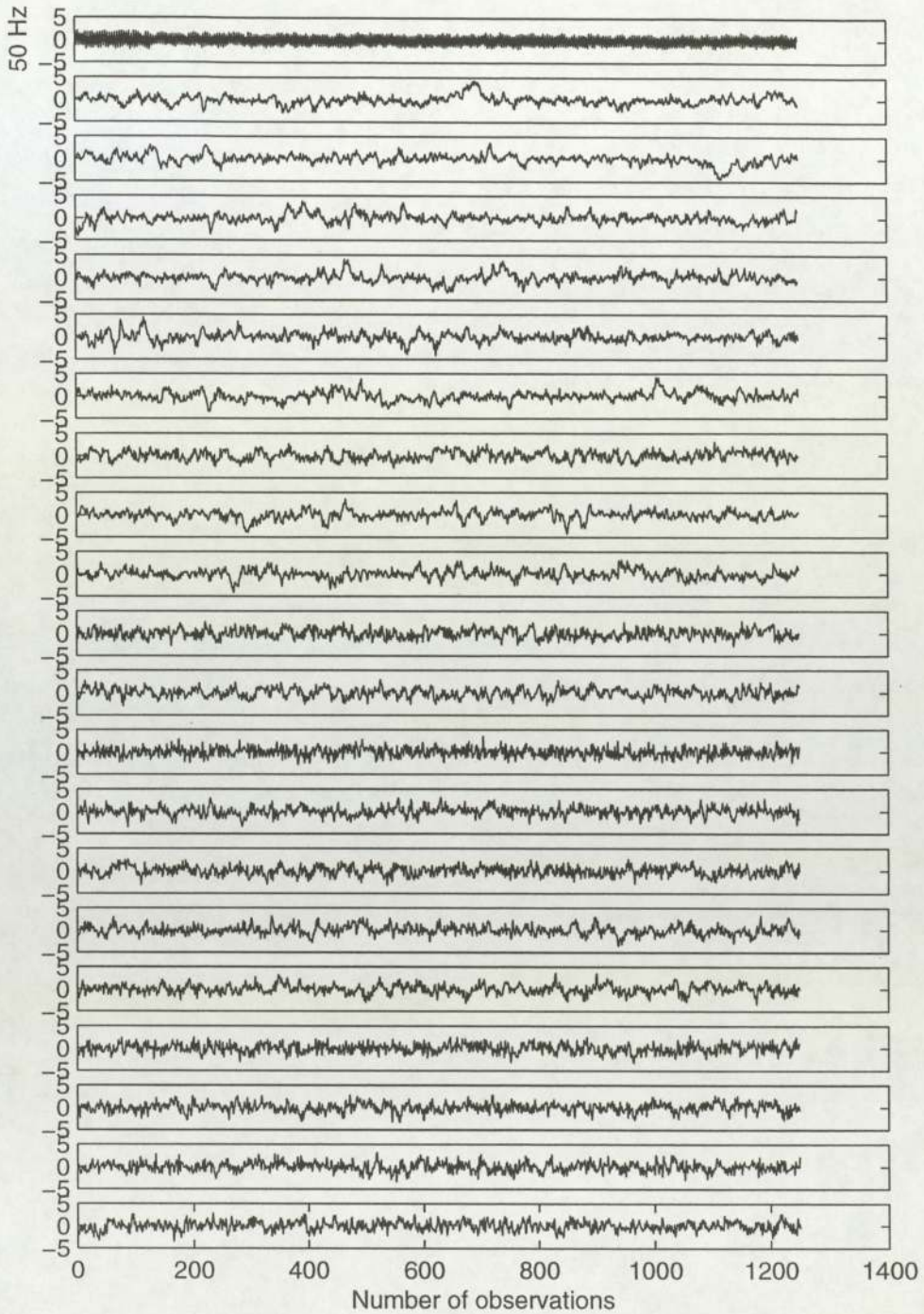

Figure D.4: FastICA pre-processed by PCA on the 150 channel MEG data described in 5.3.1 with 42 sources: last 21 components.

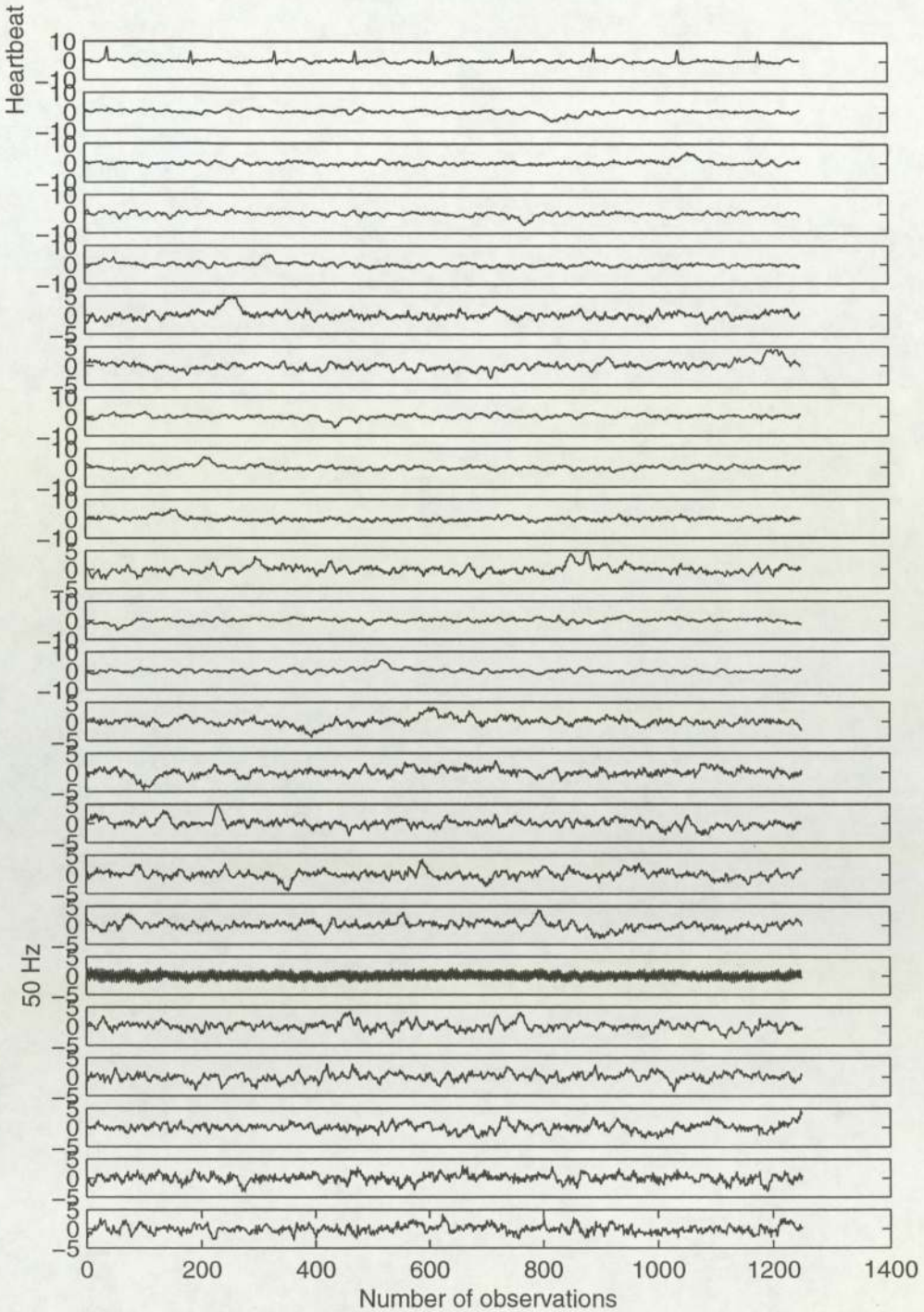

Figure D.5: FastICA pre-processed by PCA on the 150 channel MEG data described in 5.3.1 with 24 source.

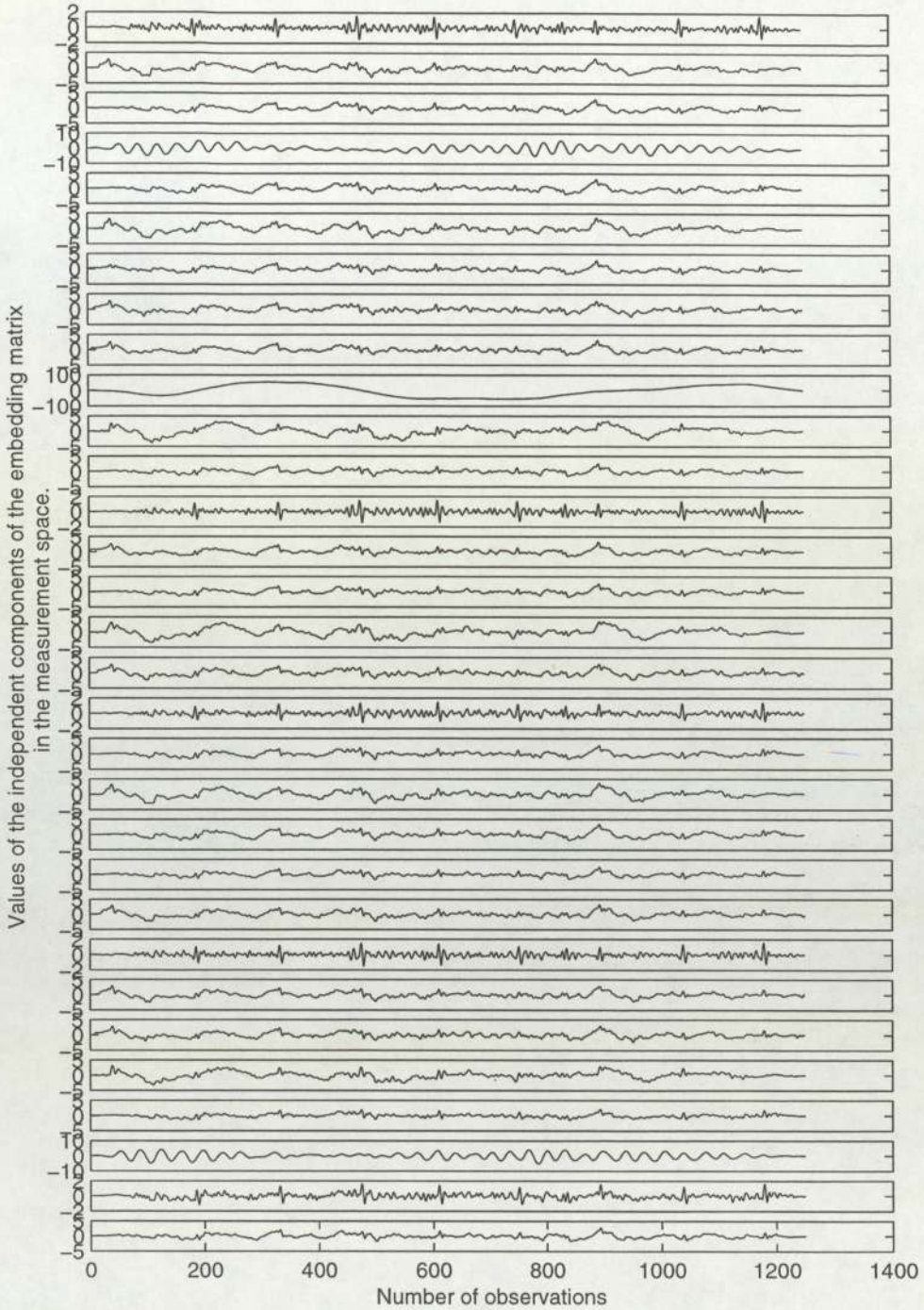

Figure D.6: The independent components of the embedding matrix of the single-channel MEG in the measurement space. We can notice that many signals corresponds to the ECG activity, this is due to the way we construct the embedding matrix, its components may be identical up to a displacement of the signal, but they reduce to the same signal in the measurement space.

Appendix E

Notation

PCA: Principal Component Analysis

ICA: Independent Component Analysis

FA: Factor Analysis

IFA: Independent Factor Analysis

Variables and constants:

i: General-purpose index, also: imaginary unit

m: Dimension of the observed data (sensors)

n: Dimension of the transformed component vector (sources)

T: Number of observations of the sources

All the vectors are printed in boldface lowercase letters,

x: Observed data, sensors, an m-dimensional random vector

s: n-dimensional random vector of transformed components s_i

η: m-dimensional random noise vector

w: m-dimensional constant vector

y: m-dimensional general-purpose random vector

All the matrices are printed in boldface uppercase letters,

A: The constant $m \times n$ mixing matrix in the ICA model

W: The constant $n \times m$ unmixing matrix in the ICA model

Functions:

E{.}: Mathematical expectation

f(.): A probability density function

f_i(.): Marginal probability density function

g(.): A scalar non-linear function

H(.): Differential entropy

I(.): Mutual information

J(.): Negentropy

J_G(.): Generalized contrast function

kurt(.): Kurtosis, or fourth-order cumulant