MAXIMUM INDEPENDENT COMPONENT ANALYSIS

by

Ruosi Guo

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Statistics)

at the

UNIVERSITY OF WISCONSIN–MADISON

2018

Date of final oral examination: 10/23/2018

The dissertation is approved by the following members of the Final Oral Committee:

Zhengjun Zhang, Professor, Statistics
Guang-hong Chen, Professor, Medical Physics and Radiology
Miaoyan Wang, Professor, Statistics
Yazhen Wang, Professor, Statistics
Chunming Zhang, Professor, Statistics

To my family

Acknowledgments

I would like to express my sincerest gratitude and appreciation to my advisor, Professor Zhengjun Zhang, for the persistent guidance and encouragement in my Ph.D. life. I have benefited greatly from his invaluable comments and suggestions on my research. Without his professional help, the present thesis would not have been accomplished.

I would like to express my sincere gratitude to my committee members, Professor Guang-hong Chen, Professor Yazhen Wang and Professor Chunming Zhang for their expert suggestions and comments, which have provided me a firm basis for the composing of this thesis. I would like to express my sincere appreciation to Professor Chunming Zhang for the great help and support in my Ph.D. life. I want to thank Professor Chunming Zhang for her contribution to implement the PCA in deciding the dimension of MaxICA, to suggest real data analysis strategy, and to initiate our joint work on Maximum Independent Component Analysis with Augmented Genetic Algorithm and Application to EEG Data. I want to thank Jingcao Wang for his contribution to the simulated annealing experiments.

I also want to thank my research colleagues. The awesome research group brings many ideas of various areas. My thank also go to all other tutors and faculty members.

Last I would like to express my special gratitude to my beloved parents who have supported and caring for me all through these years.

Contents

List of Tables

List of Figures

Abstract

In many different disciplines, finding hidden influential factors behind observed data can be essential and helpful. The majority of existing work in the literature is mainly based on linear transformation assumption, that is the hidden components are linear combinations of original sources. This thesis presents a new framework based on max-linear combinations of original sources. In contrast to independent component analysis (ICA) in linear transformation, we introduce a new type of analysis: maximum independent component analysis (MaxICA). The major tool of solving MaxICA problems is an augmented genetic algorithm with elite weighted sum selection, integrated crossover and dynamic mutation methods. The obtained results indicate that MaxICA can either extract max-linearly combined essential sources in many applications or supply as a better approximation of nonlinearly combined source signals, such as EEG data analysis presented in this thesis.

Chapter 1

Introduction

In many areas, time series data and digital signal processing data are commonly collected and require advanced analysis approaches. With the collected data, exploring hidden sources or signals is essential for analyzing and forecasting, and thus of great interest. A common and useful highly underdertermined problem is known as blind source separation (BSS), which is aimed to separate a set of source signals from a set of mixed signals with very little information about the signals or the mixing process.

The advance of independent component analysis (ICA) has made possible for researchers to discover statistically independent components and their linear representation of non-Gaussian data. The earlier work can be traced back to (Hyvärinen and Oja, 2000; Hyvärinen et al., 2001). Since its introduction, ICA has been successfully applied to blind source separation with complex data sets in many research areas. There exists a very rich literature in the field. We won't give a detailed review in this thesis. Interested readers are referred to the most recent work in analytical chemistry

(Kassouf et al., 2018), in gene expression time series (Nascimento et al., 2017), in brain image analysis (Huang et al., 2018; Artoni et al., 2018), amongst others and the references therein. Besides ICA, there are other methods of blind signal separation, i.e., principal components analysis, singular value decomposition, amongst others.

The basic assumption of ICA is that the original sources are linearly combined. One often told example is the cocktail-party effect. This effect says that if someone goes to a cocktail party and has conversation with other people, his brain will automatically filter out all other noises in the background, so that he can focus on the people he's talking with. Some related studies can be found in (Bronkhorst, 2000).

If we record two speakers' speeches at the same time using two recording devices located in a distance, and each recorded signal is a weighted sum of the speech signals emitted by the two speakers, then using ICA we can recover the two original speech signals. Such simple linear combination of the original signals is a good approach and can capture the essential structure of the data in many applications, but it is not sufficient for fully solving the problem described next. For most of the situations in practice, using linear combination assumption and recording linearly combined signals contains too much information, even though it is perfectly detailed. The estimated source signals separated from the recorded signals by using present methods like ICA contains all information, as the method was designed to separate data as precise as possible, as explained in (Hyvärinen and Oja, 2000). In some situations it is good and precise and fast as well, but for some other situations we might not need that much information or that much information is not accessible. Let's continue with the cocktail party example under a different scenario: there were two groups of people

talking to each other within each group. Suppose two partygoers Mary and John were in the party, but not in the conversations with any of these two groups. They also did not pay attention to those two group conversations. When conversations in these two groups sometimes mentioned their names or something they were interested in, then they would be attracted to the group's conversation initiatively for a short time period. As a result, by the end they might have only memorized some major information, while some recording devices could have all voice signals recorded. Let's further consider an experiment: Suppose the brain activities of both Mary and John can be recorded. Certainly, the output signals of the brain activities of both Mary and John won't be a linear combinations of the original signals, i.e., very different from the output signals from the recording devices. The output signals from the brain activities of both Mary and John are selectively combined signals. This thesis intends to develop a new model to combine the signals which is taking maximum of the source signals. This kinds of output signals are more like many practice situations, assuming the signal we recorded is the maximum of sources, and minor information can be filtered out. In nature, this phenomenon is more close to the way human brain works.

For illustration purpose, let's consider the following simple example to demonstrate the idea mentioned above. Suppose two original signals s_1 and s_2 are defined as:

$$s_1(x) = \sin(0.5x + 2),\ \ s_2(x) = 0.5\sin(1.3x + 2.5). \tag{1.1}$$

The two original signals are shown in the top panel of Figure 1.1. The middle panel of Figure 1.1 is a linear combination of the two signals: i.e., $s_1(x) + s_2(x)$. The bottom

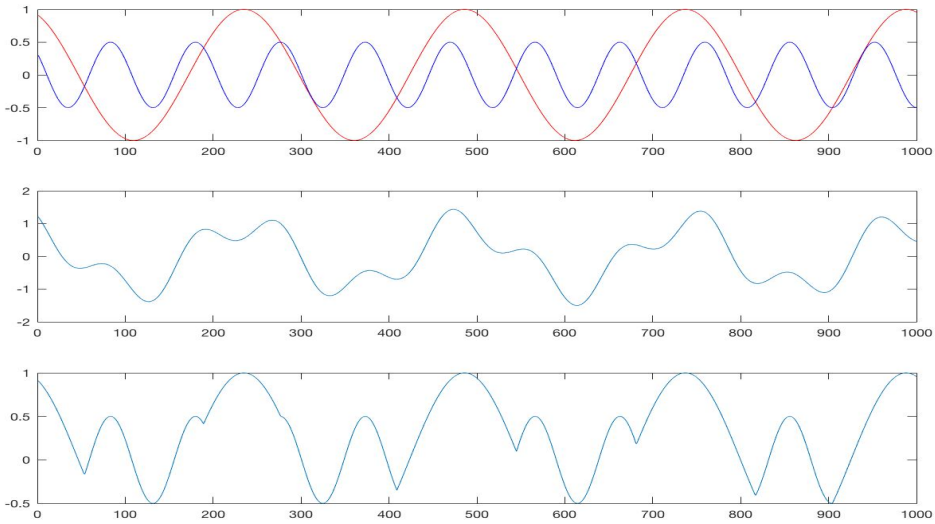


Figure 1.1: *An example shows differences between linear combination and maximum combination. The top panel is for two original signals. The middle panel is a linear combination of original signals. The bottom panel is the maxima of original signals.*

panel of Figure 1.1 is the maxima of the two signals: i.e., $\max(s_1(x), s_2(x))$. Linear combination contains all information from the two original signals. All details will be reflected in the combined signals. Even trivial turbulence will have effects on a combined signal. Under the maximum combination, we can see that in the bottom plot of Figure 1.1, the curve consists of fragments exactly came from one of the two original signals. Some small waves will be covered by larger waves. As a result, using maximum combination assumption can reduce the effects generated from small turbulence, while it mainly focuses on major information.

As an analog of ICA, one wishes to recover the original signals from the observed signals demonstrated in the bottom panel of Figure 1.1. By its nature of taking maxima of the original signals, it is natural to call the problem as maximum indepen-

dent component analysis (MaxICA) problem. Let's further consider another simple case. Figures 1.2 – 1.5 are respectively original signals, maximum combined signals, maximum combined signals with added random errors, and the estimated signals from our estimation approach developed in this study. The original signals are like those in Figure 1.1, i.e., there are two different waves. Figure 1.3 shows the maximum of the original signals from Figure 1.2 by using two different combination algorithms. After adding some random errors, waves are like those in Figure 1.4. A natural/logic issue is to recover the data in Figure 1.2 based on the data in Figure 1.4. With the MaxICA method, the estimation results are given in Figure 1.5. One can see waves are similar to those in Figure 1.2.

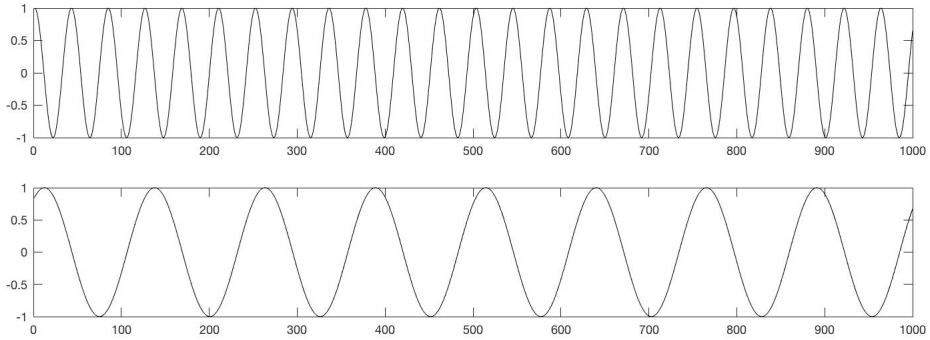


Figure 1.2: *The original signals*

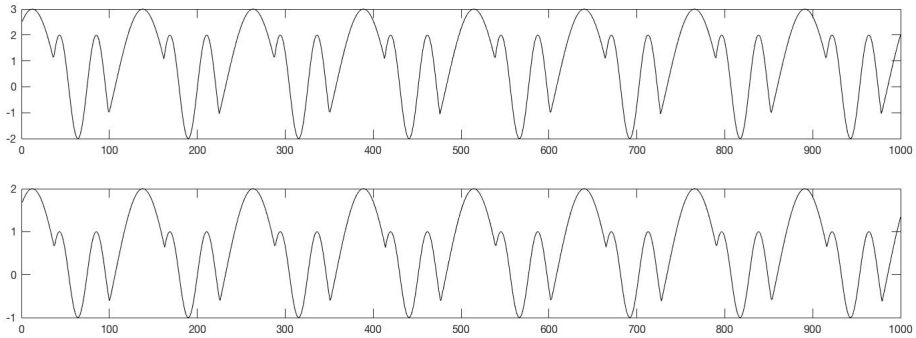


Figure 1.3: *The maximum of the original signals*

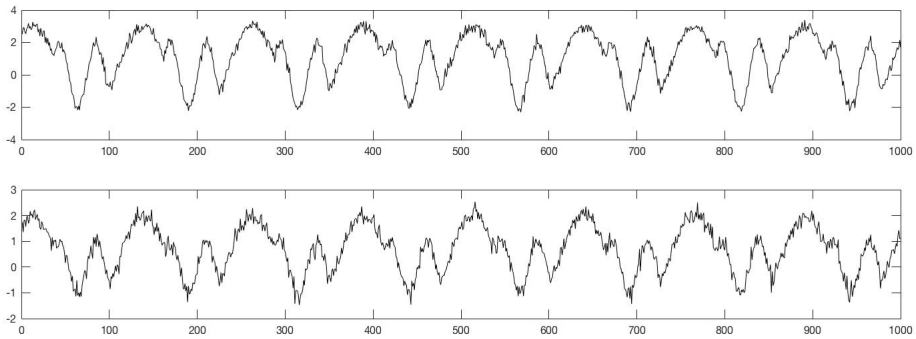


Figure 1.4: *The mixtures with errors*

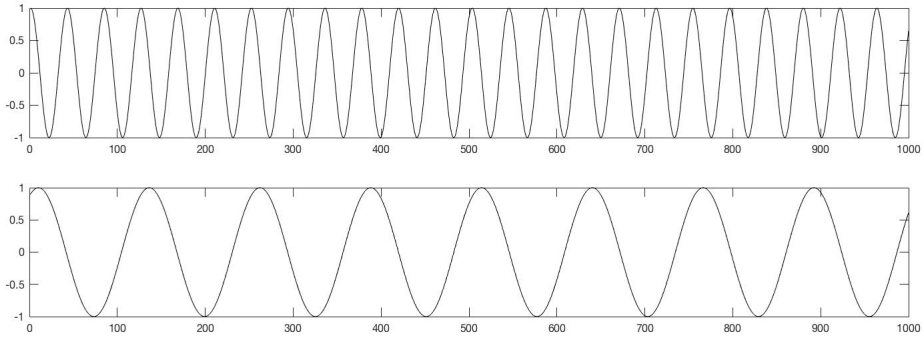


Figure 1.5: *Two signals estimated by MaxICA*

One can immediately notice that we are dealing with multivariate nonlinear time series inference problems. In time series literature, there have been quite many developments in modeling nonlinear time series. For example, the autoregressive conditional heteroskedasticity (ARCH) model and the generalized autoregressive conditional heteroskedasticity (GARCH) model are widely applied in financial time series, earthquake time series, and even in functional MRI data, e.g., (Engle, 1982; Bollerslev, 1986; Zhang et al., 2015) among others. On studying neuronal functional connectivity, readers are referred to (Zhang et al., 2016) and references therein. On the other hand, parametric models which are analog to the classical ARMA models have been developed in the literature. These models are constructed based on max-linear structures. In particular, (Zhang and Smith, 2004) demonstrated theoretical properties for a wide-class of moving maxima models. (Heffernan et al., 2007) provided a general frame work of modeling asymptotically (in)dependent time series. (Zhang, 2008) proposed a model for sea wave modeling. The paper by (Naveau et al., 2011) deals with weakly maxima of river flows. In high-frequency financial time series data modeling, (Zhang and Zhu, 2016; Idowu and Zhang, 2017) can be widely applicable. In a competing risk modeling, (Cui and Zhang, 2018) offers a new framework for max-linear regression. All of these parametric models can be candidate models for fMRI data and EEG data in neuronal image study.

Different from the existing earlier work, we introduce a new type of analysis: maximum independent component analysis (MaxICA). Our new framework can be thought analog to the spectral analysis in classical time series. In the spectral analysis, observed signals are decomposed into linear combinations of sine waves/functions. In

our new framework, observed signals are decomposed into max-linear combinations of sine waves/functions. The major tool of solving MaxICA is an augmented genetic algorithm with elite weighted sum selection, integrated crossover and dynamic mutation methods. The obtained results indicate that MaxICA can extract max-linear combined essential sources in many applications, such as EEG data analysis presented in this thesis.

The remaining chapters of the thesis are structured as follows. In Chapter 2, we introduce the general MaxICA model. In Chapter 3, we develop a new genetic algorithm which augments a popular genetic algorithm to meet our need to solve MaxICA models. We present simulations examples in Chapter 4 to show the performance of our augmented genetic algorithm. With a satisfied performance in Chapter 4, we analyze two sets of EEG data, i.e., visual processing Data and epilepsy data. Chapter 6 concludes.

Chapter 2

The model for maximum independent component analysis

In this chapter, we first briefly review two widely applied time series models in the literature. Then we introduce our proposed model for maximum independent component analysis.

2.1 The spectral analysis

The spectral analysis is widely used for data analysis in geophysics, oceanography, atmospheric science, astronomy, engineering etc., it's used with time series data. Usually it is the case that several cyclical patterns are simultaneously presented in a time series. This is different with analyzing the correlation properties of time series in the time domain, spectral analysis is analyzing frequency properties of time series, and said to be working in the frequency domain. Referring to (Cryer and Chan, 2008),

we consider a time series represented as

$$Y_t = \sum_{j=1}^{m} [A_j \cos(2\pi f_j t) + B_j \sin(2\pi f_j t)] \tag{2.1}$$

where frequencies $0 < f_1 < f_2 < ... < f_m < \frac{1}{2}$ are fixed and A_j and B_j are independent normal random variables with zero means and $Var(A_j) = Var(B_j) = \sigma_j^2$. Then Y_t is stationary with mean zero and variance

$$\gamma_k = \sum_{j=1}^{m} \sigma_j^2 \cos(2\pi k f_j).$$

If for $0 < f < \frac{1}{2}$ we define two random step functions by

$$a(f) = \sum_{\{j|f_j \leq f\}} A_j, \quad b(f) = \sum_{\{j|f_j \leq f\}} B_j$$

then we can write Equation (2.1) as

$$Y_t = \int_0^{1/2} \cos(2\pi f t) da(f) + \int_0^{1/2} \sin(2\pi f t) db(f). \tag{2.2}$$

In real data inference, (2.1) has been widely applied, especially for detecting hidden periodicities. Any zero-mean stationary process may be represented as in Equation (2.2), see (Cramer and Leadbetter, 1967). It shows how stationary processes may be represented as linear combinations of infinitely many cosine-sine pairs over a continuous frequency band.

2.2 The model for independent component analysis

Independent component analysis (ICA) is a standard statistical and computational technique for finding hidden factors that underlie sets of random variables, measurements or signals. ICA defines a model for the observed multivariate data which are assumed to be linear mixtures of some unkown variables, the mixing system is also unknown. The hidden components are assumed nongaussian and mutually independent.

Consider $\mathbf{X}$ as a matrix whose rows are observed mixtures $\mathbf{x}_1, ..., \mathbf{x}_n$, let $\mathbf{S}$ be the matrix whose rows are vectors $\mathbf{s}_1, ..., \mathbf{s}_n$. Let $\mathbf{A}$ be a matrix with elements a_{ij}, and columns of matrix A are $\mathbf{a}_j$. So the mixing model is

$$\mathbf{X} = \mathbf{AS}. \tag{2.3}$$

It can also be written as

$$\mathbf{X} = \sum_{i=1}^{n} \mathbf{a}_i \mathbf{s}_i.$$

The model in Equation (2.3) is called ICA model. $\mathbf{A}$ and $\mathbf{S}$ are unknown. Let $\mathbf{W}$ be the inverse of the estimated $\mathbf{A}$, then independent components can be obtained by

$$\mathbf{S} = \mathbf{WX}.$$

Algorithms for estimating the mixing matrix **A** and recovering the signals **s** were first given by (Hyvärinen and Oja, 2000). For more recent developments in ICA and applications, readers are referred to the references listed in Introduction.

In the next section, we present a new framework which shows some of the modeling ideas in sections 2.1-2.2.

2.3 The model for maximum independent component analysis

Consider a set of harmonic frequencies $\omega_j = \frac{j}{n}$ for $j = 1, 2, ..., n/2$, assume n to be even. Then a time series may be represented as:

$$\mathbf{s}(t) = \sum_{j=1}^{n/2} [\beta_1(\frac{j}{n}) \cos(2\pi\omega_j t) + \beta_2(\frac{j}{n}) \sin(2\pi\omega_j t)] \tag{2.4}$$

where $\beta_1(j/n)$ and $\beta_2(j/n)$ are considered as regression coefficients. Based on the transformation $\cos(2\pi\omega_j t) = \sin(2\pi\omega_j t + \frac{\pi}{2})$, one gets:

$$\mathbf{s}(t) = \sum_{j=1}^{n/2} [\beta_1(\frac{j}{n}) \sin(2\pi\omega_j t + \frac{\pi}{2}) + \beta_2(\frac{j}{n}) \sin(2\pi\omega_j t)]. \tag{2.5}$$

Suppose there are N original signals, by (2.5):

$$\mathbf{s}_i(t) = \sum_{j=1}^{n_i} [b_{ij} \sin(\alpha_{ij} t + \beta_{ij})], \quad i = 1, 2, ..., N \tag{2.6}$$

where $\mathbf{s}_i(t)$ is the ith source, n_i is the number of sine waves for the ith source, b_{ij}, α_{ij} and β_{ij} are unkown parameters. We now define our model for the maximum independent component analysis (MaxICA) as:

$$\begin{cases} \mathbf{x}_1(t) = \max\{a_{11}\mathbf{s}_1(t), a_{12}\mathbf{s}_2(t), \cdots, a_{1N}\mathbf{s}_N(t)\} + \epsilon_1(t), \\ \mathbf{x}_2(t) = \max\{a_{21}\mathbf{s}_1(t), a_{22}\mathbf{s}_2(t), \cdots, a_{2N}\mathbf{s}_N(t)\} + \epsilon_2(t), \\ \qquad\qquad\qquad \vdots \\ \mathbf{x}_p(t) = \max\{a_{p1}\mathbf{s}_1(t), a_{p2}\mathbf{s}_2(t), \cdots, a_{pN}\mathbf{s}_N(t)\} + \epsilon_p(t), \end{cases} \tag{2.7}$$

where $\mathbf{s}_i(t)$ is defined in (2.6), $\mathbf{x}_1(t), \mathbf{x}_2(t), ..., \mathbf{x}_p(t)$ are p observed signals, $\epsilon_1, \epsilon_2, ..., \epsilon_p$ are error terms.

Finding the exact values of b_{ij} and a_{ki} in (2.6) and (2.7) is not necessary, as we can have a_{ki} absorbed by b_{ij}. We can define $b_{ijk} = a_{ki}b_{ij}, \quad i = 1, 2, ..., N, \quad j = 1, 2, ..., n_i, \quad k = 1, 2, ..., p$. Here b_{ijk} is the jth coefficient of the ith source signal from the kth observation sequence. We can redefine the source signals:

$$\mathbf{s}_{ki}(t) = \sum_{j=1}^{n_i} [b_{ijk} \sin(\alpha_{ij}t + \beta_{ij})], \quad i = 1, 2, ..., N, k = 1, 2, ..., p \tag{2.8}$$

and the model for MaxICA is:

$$\begin{cases} \mathbf{x}_1(t) = \max\{\mathbf{s}_{11}(t), \mathbf{s}_{12}(t), \cdots, \mathbf{s}_{1N}(t)\} + \epsilon_1(t), \\ \mathbf{x}_2(t) = \max\{\mathbf{s}_{21}(t), \mathbf{s}_{22}(t), \cdots, \mathbf{s}_{2N}(t)\} + \epsilon_2(t), \\ \qquad\qquad\qquad \vdots \\ \mathbf{x}_p(t) = \max\{\mathbf{s}_{p1}(t), \mathbf{s}_{p2}(t), \cdots, \mathbf{s}_{pN}(t)\} + \epsilon_p(t), \end{cases} \tag{2.9}$$

where $\mathbf{s}_{ki}(t)$ is defined in (2.8), $\epsilon_1, \epsilon_2, ..., \epsilon_p$ are error terms. Now the $\mathbf{s}_{ki}(t)$ is an original signal with specific coefficients. $b_{ijk}, \alpha_{ij}, \beta_{ij}$ are parameters to be estimated. One can see model (2.9) has a total $(p+2)(n_1 + n_2 + ... + n_N)$ unknown parameters to be estimated.

For convenience in practice, if we assume b_{ijk} are identical for all j in each original signal, in another words, the coefficients of each sine wave in a source signal are all same, then the source model can be further simplified:

$$\mathbf{s}_{ki}(t) = \sum_{j=1}^{n_i} [b_{ik} \sin(\alpha_{ij} t + \beta_{ij})], \quad i = 1, 2, ..., N, k = 1, 2, ..., p. \tag{2.10}$$

The number of unknown parameters becomes $Np + 2(n_1 + n_2 + ... + n_N)$. The number has decreased dramatically comparing with model (2.8) - (2.9).

Similar to the spectral analysis, alternative model for s_{ki} can be defined as:

$$\mathbf{s}_{ki}(t) = \sum_{j=1}^{n_i} [B_{ijk} \sin(\alpha_{ij} t + \beta_{ij})], \quad i = 1, 2, ..., N, k = 1, 2, ..., p \tag{2.11}$$

where B_{ijk} are selected independently from normal distribution with means of zero. Then the alternative model is MaxICA with (2.11). We will perform a simulation example to demonstrate the original signals in the data simulated from this model can also be recovered from our MaxICA with augmented genetic algorithm.

For example, let $\mathbf{s}_1$ and $\mathbf{s}_2$ be the two original signals:

$$\begin{aligned} \mathbf{s}_1(t) = {} & 0.17724 \sin(1.2t + 2) + 0.4609 \sin(0.5t + 1) + 0.80601 \sin(0.34t + 23) \\ & -2.4463 \sin(1.3t + 2) - 0.34539 \sin(0.6t - 3), \end{aligned}$$

$$\mathbf{s}_2(t) = -0.57756\sin(0.5t + 1.7) - 1.1414\sin(2t + 2.1) - 0.044345\sin(1.4t + 2)$$
$$+0.16376\sin(0.25t + 1) + 0.23883\sin(1.67t + 2).$$

$b_{11} = 0.21265, b_{12} = -1.2195, b_{21} = 0.69503, b_{22} = -0.55459$ generated from standard normal distribution.

$$\mathbf{x} = \begin{Bmatrix} \max\{b_{11}\mathbf{s}_1, b_{12}\mathbf{s}_2\} + \epsilon_1(t) \\ \max\{b_{21}\mathbf{s}_1, b_{22}\mathbf{s}_2\} + \epsilon_2(t) \end{Bmatrix}.$$

Using MaxICA, Figure 2.1 gives the fitting result. Red lines are simulated observations, blue lines are fitted lines. We can tell from the Figure that the fitted lines follow the shape and the trends of red lines. The fitting result looks good. This result demonstrates that MaxICA with the augmented genetic algorithm also works for MaxICA model with (2.11).

Note that althogh MaxICA and ICA are both aimed to find hidden components, they have different application domains. One should not think to use MaxICA to replace ICA or vice versa. In chapter 4, we will demonstrate their performance when they are applied to simulated data under various scenarios.

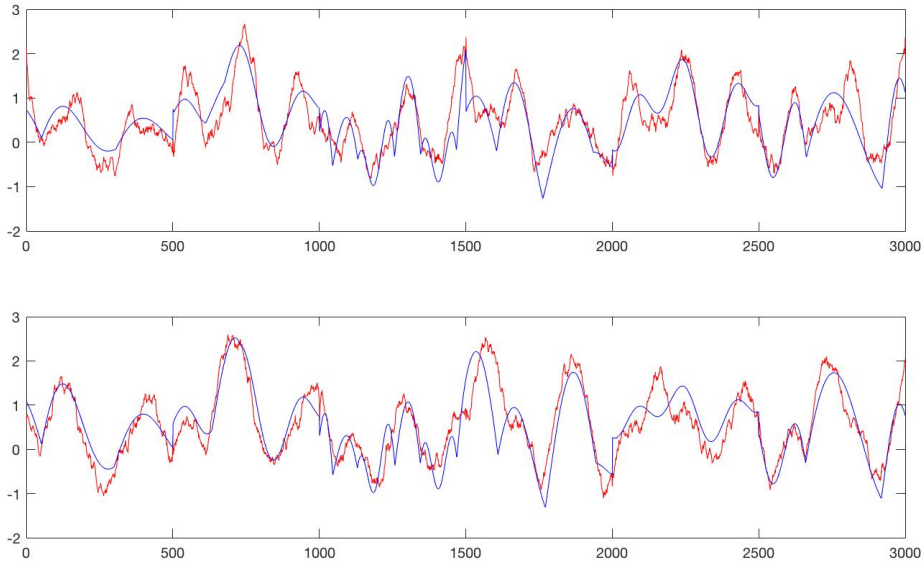


Figure 2.1: *Fitting result for simplified MaxICA model with random amplitudes*

Chapter 3

Genetic Algorithm Based Approach For Maximum Independent Component Analysis

Constrained by the formulation of MaxICA model, it's difficult to estimate the values of parameters using classical estimation approaches, e.g., maximum likelihood estimation, methods of moments, and any popular estimation methods in time series studies. We turn to use a mathematical optimization method to solve the problem. In the area of evolutionary computation, a family of algorithms for global optimization was inspired by biological evolution, and it is a rapidly growing area of artificial intelligence nowadays. Evolutionary programming was introduced by Lawrence J. Fogel in the US, and John Henry Holland called his method a genetic algorithm in his book *Adaption in Natural and Artificial Systems*(1975).

In this chapter, we first review the popular genetic algorithm. Then we introduce an augmented genetic algorithm with elite weighted sum selection, integrated crossover and dynamic mutation rate. We will also give a mathematical proof for the convergence of the augmented genetic algorithm for MaxICA.

3.1 A brief review of genetic algorithms

Genetic algorithm (GA) is a search method based on principles of natural selection and genetics. The algorithm was generated based on Darwin's theory. While using genetic algorithm for optimization problems, the solutions are evolving. Some basis can be found in (Obitko et al., 1998). The algorithm starts with a set of randomly generated solutions in a prespecified region for parameter called population. Once a population is generated, one takes paired solutions from the population to form a new population based on the selection rule. Hoping that the newly generated population can be a better one. By checking solutions' fitness values, better solutions are more possible to be selected to generate new offsprings. Here fitness is a scalar computed from the objective function, indicating how close is the given solution to the preselected goal, if one solution is more close to the goal, it will be more possible to reproduce. This procedure is repeated to direct the population to the global optimum.

The outline of the basic genetic algorithm is listed in Table 3.1.

Table 3.1: *The Basic Genetic Algorithm.*

1.[**Start**] Randomly generate population of size n;
2.[**Fitness**] Compute the fitness of all chromosomes in the population;
3.[**New population**] Generate a new population by repeating following steps:
i.[**Selection**] Select parent chromosomes based on the fitness;
ii.[**Crossover**] Cross over the parents to generate a new offspring. If no crossover was made, offspring is exactly the copy of parents;
iii.[**Mutation**] Mutate new offspring with a mutation probability;
iv.[**Accepting**] Place newly generated offspring in a new population;
4.[**Replace**] Use the newly generated population for a further run of the algorithm;
5.[**Test**] If the end condition is satisfied, stop, and return the best solution;
6.[**Loop**] Go to step 2.

We note that the basic GA is very general, and it can be implemented in various problems. For example, (Haupt et al., 1998) listed some sophisticated GAs, such as hybrid GA, messy GA, i.e., (Paes et al., 2017) introduced a DRQ genetic algorithm for solving the facility layout problem (FLP), (Amirjanov, 2015) introduced a dynamical adjustment of a search space size to analytically establish the setting of a parameter of a GA, (Driss et al., 2015) introduced a new genetic algorithm for job-shop scheduling problem (JSP), amongst others.

In the following sections, we present an augmented algorithm to the basic genetic

algorithm for solving the MaxICA inference problems. The augmented algorithm can also be used in many other global optimization problems.

3.2 MaxICA objective function

To formulate the objective function for the MaxICA model, let row vector $\mathbf{x}_k(t), k = 1, 2, ..., p$ be the kth observed signal. Define row vector

$$\mathbf{f}_k(\theta_k, t) = \max\{\mathbf{s}_{k1}(t), \mathbf{s}_{k2}(t), \cdots, \mathbf{s}_{kN}(t)\} + \boldsymbol{\gamma}, k = 1, 2, ..., p,$$

where row vector $\mathbf{s}_{ki}(t)$ defined in (2.10), $\max\{\mathbf{s}_{k1}(t), \mathbf{s}_{k2}(t), \cdots, \mathbf{s}_{kN}(t)\}$ is elementwise maximum of vectors $\mathbf{s}_{ki}(t)$. $\boldsymbol{\gamma} = (\gamma, \gamma, ..., \gamma)$ is a row vector, every element of it is the same scalar γ. θ_k is a parameter set including $b_{ik}, \alpha_{ij}, \beta_{ij}, \gamma$, and γ is the horizontal location offset parameter used for adjusting the horizontal location of the model, $n_i, i = 1, 2, ..., N$, N are priori decided values. The objective function is:

$$f(\theta, t) = \sum_{k=1}^{p} ||\mathbf{x}_k(t) - \mathbf{f}_k(\theta_k, t)||_2 \tag{3.1}$$

where $\theta = \{\theta_1, \theta_2, ..., \theta_p\}$. Our goal is to minimize $f(\theta)$ which is the summation of the l^2 norm of differences between observed signal and the modeled signal.

3.3 MaxICA chromosome representation

Chromosome in the genetic algorithm is the set of unknown parameters. Our goal is to minimize function (3.1), and solving for the optimum of parameters. Let the parameter set θ contain $b_{ik}, \alpha_{ij}, \beta_{ij}, \gamma$, totally $Np + 2\sum_{i=1}^{N} n_i + 1$ parameters. Hence MaxICA chromosome has $Np + 2\sum_{i=1}^{N} n_i + 1$ elements, the order of these parameters is shown below:

$$\begin{aligned}\text{Chromosome} = \{&\alpha_{11}, \beta_{11}, ..., \alpha_{1n_1}, \beta_{1n_1}, ..., \alpha_{N1}, \beta_{N1}, ..., \alpha_{Nn_N}, \beta_{Nn_N}, \\ &b_{11}, ..., b_{1N}, ..., b_{p1}, ..., b_{pN}, \gamma\}.\end{aligned} \tag{3.2}$$

The choice of order is flexible, in this thesis we follow the parameter order given in (3.2).

3.4 MaxICA selection operator

In the selection step of genetic algorithms, solutions are selected from the population to be parents to crossover. How to appropriately select solutions is a problem of interest. There exist many selection methods. A very popular selection method is called Roulette Wheel Selection, more details and its mathematical structure of the roulette wheel selection can by found in (Jebari and Madiafi, 2013). In this method, one selects parents based on their fitness. Solutions with better fitness would be more possible to be selected. Like the method's name, one can imagine a roulette wheel with all solutions placed on the wheel, and the area of each solution is proportional to its fitness value like a pie chart. The selection procedure starts like

$$f_1 \xrightarrow{d_1} f_2 \xrightarrow{d_2} \dots \xrightarrow{d_{n-2}} f_{n-1} \xrightarrow{d_{n-1}} f_n$$

Figure 3.1: *Sorted f_i sequence with differences*

randomly throwing a marble on the wheel to pick the solution, and solutions with better fitness scores will be selected more easily. Table 3.2 gives the outline of this method.

Table 3.2: *Outline for Roulette Wheel Selection.*

1.[**Sum**] Calculate sum S of all chromosome fitnesses in population;
2.[**Select**] Generate random number r from interval $(0, S)$;
3.[**Loop**] Go through the population and sum fitnesses from 0, define the sum to be s. When s is greater than r, stop and return the chromosome where you are.

Another selection method is called rank selection. Rank selection will rank the solutions by fitness. As a result, each solution has a chance to be selected. The disadvantages for these methods are obvious. For roulette wheel selection, if fitness values of some solutions are too large or too small, then some other solutions will have very small chances to be selected, the algorithm will then select those good solutions too often, and can hardly get any new solutions. For the rank selection, it will lead to a slower convergence because the best solution does not differ very much from others, all solutions are equally likely to be selected, no matter how good it is or how bad it is. More analysis about selection schemes can be found in (Goldberg and Deb, 1991).

To get over these disadvantages, we introduce a new selection method called elite weighted sum selection. Define sorted scores of population members to be $f_i, i = 1, 2, ..., n$. If the population size is n, assume $f_1 \leq f_2 \leq ... \leq f_n$. f_i are computed by pluging in solutions into the objective function (3.1). Smaller f_i means a better solution, it also has a higher fitness value. Define differences between neighboured f_i's as $d_j = f_{j+1} - f_j, j = 1, 2, ..., n-1$. This relation is shown in Figure 3.1. Define a total D:

$$D = \sum_{j=1}^{n-1} d_j + \sum_{j=2}^{n-1} d_j + ... + d_{n-1}. \tag{3.3}$$

Define $p_i, \quad i = 0, 1, ..., n-1$:

$$p_0 = 0, p_1 = \frac{\sum_{j=1}^{n-1} d_j}{D}, p_2 = \frac{\sum_{j=1}^{n-1} d_j + \sum_{j=2}^{n-1} d_j}{D}, ..., p_{n-1} = 1. \tag{3.4}$$

Rewrite (3.4):

$$p_0 = 0, p_i = \frac{\sum_{k=1}^{i} \sum_{j=k}^{n-1} d_j}{D}, i = 1, 2, ..., n-1. \tag{3.5}$$

p_i are cumulated probabilities. This new selection starts by generating a random real number r from $(0, 1]$, if $p_{i-1} < r \leq p_i$, then select the ith solution. The solution with the worst fitness value is dropped from the population and will not be used to generate offsprings. We call this new selection method the Elite Weighted Sum Selection. Compared to the existing selection methods, the elite weighted sum selection can guarantee that all solutions besides the worst one can have some chance to be selected even if some solutions' fitness values are much larger than others'. This overcomes the disadvantage of roulette wheel selection. Furthermore, compared to the rank

selection, chances for solutions to be selected are not all the same, better solutions can have higher chances to be selected. Moreover, what's most important is that solutions having similar fitness values will have similar chances to be selected. For optimization problems, the elite weighted sum selection method can be more efficient. For example, suppose we have a set of ordered solution scores computed based on the objective function: $\{12.5, 210, 220, 310, 375\}$ corresponding to solution 1, solution 2, solution 3, solution 4, and solution 5 respectively. Figure 3.2 compares different selection methods. In these solutions, solution 1 has the best score and it's quite different from other solution scores. In the figure, the left pie chart uses the roulette wheel selection method, the middle pie chart uses the roulette wheel selection method with the worst solution dropped, and the pie chart on the right uses our elite weight sum selection method. From the left and the middle charts, one can see that solution 1 has a very large area compared with other solutions. This will make the algorithm select solution 1 very often, other solutions have very small chances to be selected, the genetic algorithm can hardly find better solutions. On contrary for the chart on the right using elite weighted sum selection method, solution 1 does not take too much area, and all solutions are possible to be selected, the areas for solutions 2, 3 and 4 are all larger than that in the left two charts. Moreover, with elite weighted sum selection method, better solution will always have higher chance to be selected. In the right chart of Figure 3.2, the areas of solutions 1, 2, 3, 4 are in descent order. Another advantage of this elite weighted sum selection is that solutions with similar scores will have similar chances to be selected. From the right chart of Figure 3.2, solution 2 has a score of 210, solution 3 has a score of 220, and their areas in the

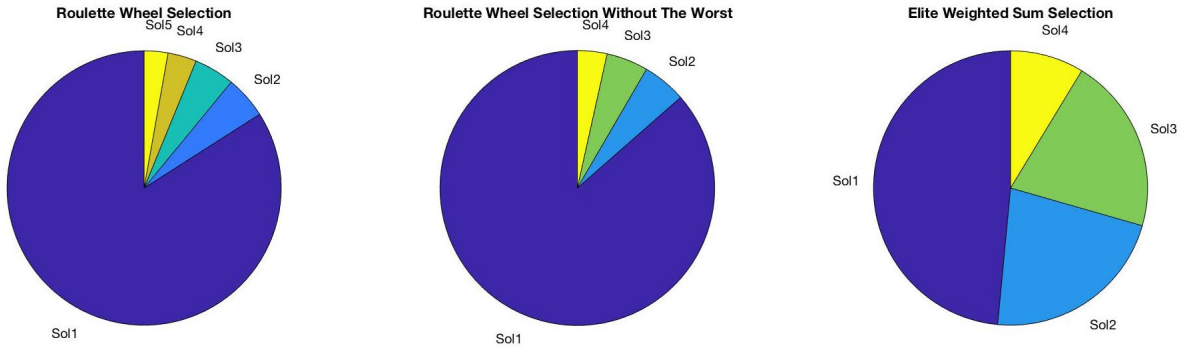


Figure 3.2: *Selection Methods Comparation*

chart are also of similar sizes. Given these empirical evidence, the elite weighted sum selection method can be more efficient for complicated optimization problems.

3.5 MaxICA crossover operator

Crossover is a basic operator of genetic algorithm. The performance of genetic algorithm depends on it a lot. There are many crossover techniques that are built to get the optimum solution as fast as possible in minimum generations, see examples in (Umbarkar and Sheth, 2015). The selection of crossover operator has large impact on the performance of the algorithm. In this section, we will combined different crossover method together. Below are some classic crossover methods.

1. The first crossover method is called 1-point crossover. It is one of the simplest crossover technique. This method uses the single point fragmentation of the parents and combines the parents at the crossover point to create the offspring. It first selects two parents used for crossover and then randomly selects a

crossover point. Two offsprings are created by combining the parents at the crossover point, and the crossover points from parents will be averaged. An example is shown below.

Parent 1: 1 1 1 1 1 1

Parent 2: 0 0 0 0 0 0

Offspring 1: 1 1 0.5 0 0 0

Offspring 2: 0 0 0.5 1 1 1

In this example, the 3rd point is selected to be the crossover point.

2. The second crossover method is called 2-point crossover. Similar to 1-point method, we select 2 crossover points, and the offspring is created by combining parents at 2 crossover points. An example is shown below.

   Parent 1: 1 1 1 1 1 1 1 1

   Parent 2: 0 0 0 0 0 0 0 0

   Offspring 1: 1 0.5 0 0 0 0.5 1 1

   Offspring 2: 0 0.5 1 1 1 0.5 0 0

   In this example, the 2nd and the 6th points are selected to be crossover points.

3. The third crossover method is called reduced surrogate crossover. This method minimizes the unwanted crossover operations in case of the parents having same genes. In this method, first check for the genes in the parents and create a list of all possible crossover points where the genes of the both parents are different. After the check, if no crossover point is there then no action is taken. But if parents are differing in more than 1 gene then keeps the list of all crossover

points and randomly selects one crossover point from the list and perform 1-point crossover.

4. The forth crossover method is called uniform crossover. This method provides the uniformity in combining the bits of both parents. It will first choose a uniform random real number u (between 0 and 1). It creates two offsprings of genes selected from both of the parents uniformly. The random real number decides whether the first child select the genes from first or second parent. Here we let the threshold be 0.5, if $u < 0.5$ select gene from the first parent, otherwise from the second parent, and repeat this process for all genes.

5. The fifth crossover method is called average crossover. Average crossover only creates one offspring from two parents. The way it creates offspring is taking average of the two parents. Each gene in a child is taken by averaging genes from both parents.

6. The sixth crossover method is called discrete crossover. Similar to the uniform crossover, but discrete crossover only creates one child from two parents. It first choose a random real number and use the number to decide from which parent to take the genes for the child.

In our crossover method, we are not selecting specific one crossover method, instead, we will combine these methods together. This is motivated by the concept of genetic algorithm. The genetic algorithm is a simulation of chromosome evolution. In the process of evolution, various crossover methods could happen. So stick with one crossover method could slow down the speed of convergence especially for complicated

optimization problems. Thus, in each iteration of the program, we can randomly choose a number from one to six, and use the number to decide which crossover method we mentioned above to be used in that iteration. This method is called the Combined Crossover Operator, the idea was proposed by (Hassan, 2015). The crossover methods don't have to be the six above, it's very flexible, they could also be other crossover methods depending on the problem.

3.6 MaxICA mutation operator

Mutation is also a basic operator of genetic algorithm. It's necessary and important. Mutation is to randomly choose some genes from the parent chromosome and replace them by new random numbers. Sometimes the mutation could be beneficial sometimes it could be harmful, so we usually will keep a low mutation rate like 0.05. While performing the algorithm, we found that changing mutation rate while the converging process get stucked could increase the convergence speed. With this observation, we propose a dynamic mutation operator. If the program is converging fast, it doesn't matter a lot to keep a high mutation rate. But when the program get stucked, and cannot find better solutions for a number of iterations, lower down the mutation rate can help avoid harmful mutations, thus will also increase the speed a little bit. As a result, the mutation rate is changing dynamically, while no better solutions are found for a number of iterations, the mutation rate will decrease until it reaches the lower bound, otherwise if the convergence speed is getting faster, mutation rate will increase until it reaches the upper bound. More information about dynamic mutation

rate operator can be found in (Thierens, 2002).

3.7 Augmented genetic algorithm and its convergence

For the Genetic Algorithm with elite weighted sum selection, combined crossover and dynamic mutation rate, we will call this method ECDGA for convenience. In this section we will give a mathematical illustration of the effectiveness and convergence of ECDGA. The structure of the process is inspired by ((Bhandari et al., 1996)), and the paper gives a proof for GA in discrete situation. In this section, we will consider the continuous situation.

Strings of lenght L are considered over an infinite alphabet $\mathcal{A} = \{\alpha_1, \alpha_2, ...\}$ representing a solution. The number of possible strings is infinity. Let $\mathcal{S}$ be the collection of all such strings S. Each string has a fitness value in terms of *fit*. So there exists ordered relation between the fitness of two strings.

In the following four sections, we will discuss the classification of strings, partitioning of populations, transition representations between populations and demonstrate the convergence of augmented genetic algorithm based on Markov chain.

3.7.1 Classification of Strings

Let $\mathcal{C} = \{fit(S) : S \in \mathcal{S}\}$. The number of elements in $\mathcal{C}$ is infinity, $\mathcal{C} = \{F_1, F_2, ...\}$, where $F_1 \geq F_2 \geq ....$ One can partition $\mathcal{S}$:

$$\mathcal{S}_i = \{S : S \in \mathcal{S} \quad \text{and} \quad F_i \geq fit(S) > F_{i+1}\} \quad \forall i = 1, 2, ...$$

Let the number of elements of $\mathcal{S}_i$ be u_i.

$$\mathcal{S}_i \neq \emptyset, \quad \forall i = 1, 2, ...; \mathcal{S}_i \cap \mathcal{S}_j = \emptyset, \quad \cup \mathcal{S}_i = \mathcal{S}.$$

Hence for any two strings $S_1 \in \mathcal{S}_i$ and $S_2 \in \mathcal{S}_j$,

$$fit(S_1) > fit(S_2) \quad if \quad i < j,$$
$$fit(S_1) = fit(S_2) \quad if \quad i = j,$$
$$fit(S_1) < fit(S_2) \quad if \quad i > j.$$

3.7.2 Partitioning of Populations

Population Q is a collection of M strings. Define Q:

$$Q = \{S_1, S_1, ..., (\sigma_1 \quad \text{times}), S_2, S_2, ..., (\sigma_2 \quad \text{times},), ..., S_\xi, S_\xi, ..., (\sigma_\xi \quad \text{times}) : S_i \in \mathcal{S}; \sigma_i \geq 1;$$
$$\text{for all } i = 1, 2, ..., \xi; S_i \neq S_j, \text{for all } i \neq j; \sum_{i=1}^{\xi} \sigma_i = M\}.$$

Two populations are equal if both of them contain the same number of copies of the same string. Define: $fit(Q) = \max_{S \in Q} fit(S)$. Thus $fit(Q) \leq F_1$ for all $Q \in \mathcal{Q}$. Define:

$$E_i = \{Q : Q \in \mathcal{Q} \quad \text{and} \quad fit(Q) = F_i\} \quad i = 1, 2, ...$$

$E_i \cap E_j = \emptyset$, for $i \neq j$ and $\cup E_i = \mathcal{Q}$. Let e_i be the number of populations in E_i, so that $\sum e_i = \mathcal{N}$. Let Q_{ij} be the jth population of E_i, $j = 1, 2, ..., e_i$ and $i = 1, 2, ...$

3.7.3 Transitions Between Two Populations

With ECDGA, a new population $Q_{kl}, l = 1, 2, ..., e_k; k = 1, 2, ...$ generated from Q_{ij}, the best solution in Q_{ij} will be passed into Q_{kl}, the fitness value of the new population is at least $fit(Q_{ij}) = F_i$. This process is called a transition from Q_{ij} to Q_{kl}.

Let $p_{ij.kl}$ be the probability that $Q_{kl} \in E_k$ generated from $Q_{ij} \in E_i$. Let $p_{ij.k}$ denote the probability of generating a population in E_k from Q_{ij}. Thus

$$p_{ij.k} = \sum_{l=1}^{e_k} p_{ij.kl}; \quad j = 1, 2, ..., e_i; \quad i, k = 1, 2, ...$$

So,

$$\sum_k p_{ij.k} = 1; \forall \quad j = 1, 2, ..., e_i, \quad i = 1, 2, ...$$

Recall the theroem from ((Bhandari et al., 1996)),

Theorem 3.1. *For all* $j = 1, 2, ..., e_i$ *and* $i = 1, 2, ...$*:*

$p_{ij.k} > 0 \quad$ *if* $k \leq i$,

$p_{ij.k} = 0 \quad$ *if* $k > i$.

Thus there is always a positive probability to result in one of the populations Q_{kl} of E_k from $Q_{ij} \in E_i$, if $k \leq i$; and zero probability if $k > i$.

Considering the chance of reaching to a population of E_k in iteration $t+1$ from a population Q_{ij} of E_i in tth iteration. This process of transitions can be viewed as a Markov chain.

3.7.4 Markov Chain and ECDGA

Let η_{ij} be the probability of occurrence of the population Q_{ij} at the initial step. Then for ECDGA,

$$\eta_{ij} \geq 0; \forall j = 1, 2, ..., e_i; i = 1, 2, ... \quad \text{and} \sum_i \sum_j \eta_{ij} = 1.$$

The populations of a ECDGA together with $\{\eta_{ij}\}$ and the conditional probabilities $p_{ij.kl}$ constitute a Markov chain. Let's order the states as $Q_{11}, Q_{12}, ..., Q_{1e_1}, Q_{21}, Q_{22}, ...,$ $Q_{2e_2}, ...$ The number of rows in the transition probability matrix $\mathcal{P}$ is $\mathcal{N}$. Every element of $\mathcal{P}$ is $p_{ij.kl}$. Let $p^{(n)}_{ij.kl}$ be the probability that ECDGA results in Q_{kl} at the nth step given that the initial state is Q_{ij}, then the n-step transition probability

matrix $\mathcal{P}^{(n)}$ is $\mathcal{P}^n$. Let $p^{(n)}_{ij.k} = \sum_l p^{(n)}_{ij.kl}$. $p^{(n)}_{ij.k}$ denotes the probability of reaching a population in E_k in n steps with the starting population as Q_{ij}.

In a Markov chain, a set C of states is said to be closed if no state outside C can be reached from any stat in C. E_1 is closed.

To show the convergence of ECDGA to a global optimal solution it is sufficient to show that starting from any state Q_{ij}, ECDGA eventually result in one of the states $Q_{1l}, l = 1, 2, ..., e_1$ of E_1. In other words, $p^{(n)}_{ij.kl} \to 0 \quad \forall k \geq 2$, as $n \to \infty$, and for all i and j. Based on the theorem from (Bhandari et al., 1996),

Theorem 3.2. *For an ECDGA with the probability of mutation* $0 < q \leq \frac{a-1}{a}$,
$\lim_{n\to\infty} p^{(n)}_{ij.k} = 0$ *for* $2 \leq k, \forall j = 1, 2, ..., e_i, i = 1, 2, ...$
Hence, $\lim_{n\to\infty} p^{(n)}_{ij.1} = 1, \quad \forall j = 1, 2, ...e_i, i = 1, 2, ...$

With the established theorem, the following conclusions can be made:

- From $Q_{1j} \in E_1$, one can at most reach to $Q_{1j_1} \in E_1$.
- For any state $Q_{ij}, i \geq 2$, $p^{(n)}_{ij.k} \to 0$ as $n \to \infty, \forall k \geq 2$.
- mutation rate $q > 0$. In each iteration, the knowledge of the best string obtained in the previous iteration is preserved within the population.
- Transition probabilities from one population to another depend on the process of generation of the mating pool.
- The structures of E_is depend on the characteristics of the fitness function. The transition probabilities vary for different fitness functions even if the string length and the set of alphabet $\mathcal{A}$ is kept constant.

- Q_{ij} will eventually result in a population containing the best string for $i \geq 2$.

3.8 MaxICA algorithm

The data for MaxICA algorithm should be recorded simultaneously at the same time period from different locations. The reason is that in practice, the hidden components might be changing with time. As an example, if we record the electroencephalogram (EEG) signals of patients with epilepsy, the brain components activity during seizure period and non-seizure period might be different. Also recording from different locations can offer us more information about the hidden components, which might help us find more accurate result.

In the data, there usually exist unwanted noises. Noises could be caused by environment, electronic experimental machine or body movements. So statistical smoothing methods can be applied to the observed data. As mentioned in chapter 2, we need to decide the value of N and $n_1, n_2, ..., n_N$. N is the number of hidden components, $n_1, n_2, ..., n_N$ are the numbers of sine waves in each one of N components. For N, use principle component analysis (PCA) to decide the number of components that account for the majority variation of the data. For $n_1, n_2, ..., n_N$, based on simulations and experiments, a good choice is taking $n_1 = n_2 = ... = n_N = 5$, this applys to most of the problems. After these values being decided, we can apply the algorithm to the data. Considering varying components, we need to seperate the data into smaller pieces beforehand. Figure 3.3 is an example of seperating observations into three parts. We then apply the genetic algorithm with elite weighted sum

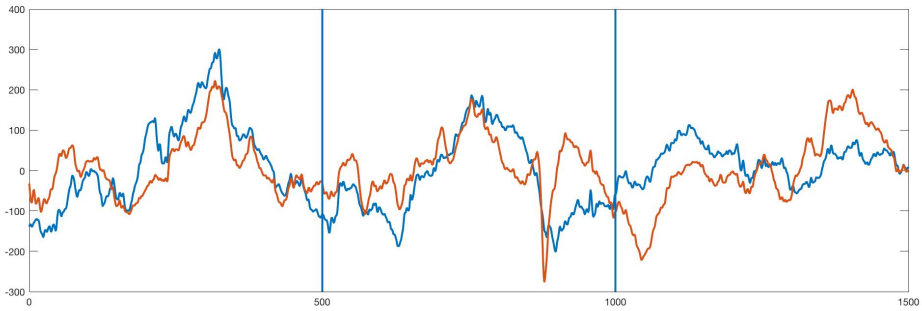


Figure 3.3: *Seperate the observed signals into three parts*

selection, combined crossover and dynamic mutation as mentioned in chapter 3 to each piece respectively. Table 3.3 lists the outline for MaxICA algorithm.

Table 3.3: *Outline for MaxICA.*

1.[**Smoothing**] Apply statistical smoothing[1]to the data;
2.[**PCA**] Use PCA to decide the number of hidden components;
3.[**Piecewise Optimization**] Seperate the data into pieces and apply the GA[2]on each piece respectively.

[1] Moving average.

[2] Genetic Algorithm with elite weighted sum selection, combined crossover, dynamic mutation. The stopping criterion is: no better solution can be found for 10000 iterations.

Chapter 4

Simulation Studies

We use four simulation examples to demonstrate our MaxICA algorithm and its performance. In Example 1, we simulated five original signals, obtained five observed signals, then use MaxICA to fit and recover the original signals. We will also compare the results generated by different methods. In Example 2, we simulated signals based on components generated in section (5.1), and apply MaxICA and ICA respectively. In Example 3, we simulated signals from ICA model, apply MaxICA method to see its performance. In Example 4, brain images are used for comparing different methods.

Example 1. Let $s_i, i = 1, 2, ..., 5$ be the five original signals. s_i is defined as follows:

$$s_1(t) = \sin(1.5t + 2) + \sin(0.3t + 1),$$

$$s_2(t) = \sin(0.7t + 1.5) + \sin(2t + 2.5),$$

$$s_3(t) = \sin(3t + 2) + \sin(t + 0.1),$$

$$s_4(t) = \sin(2t + 1.5) + \sin(1.2t + 0.6),$$

$$s_5(t) = \sin(t + 0.4) + \sin(3t + 2).$$

Let $\mathbf{s} = (s_1(t), s_2(t), s_3(t), s_4(t), s_5(t))'$,

$$\text{coefficient matrix } B = \begin{pmatrix} b_{11} & \cdots & b_{15} \\ \vdots & \ddots & \vdots \\ b_{51} & \cdots & b_{55} \end{pmatrix} = \begin{pmatrix} 0.19407 & 2.143 & 2.3734 & 2.1781 & 3.3999 \\ 3.6811 & 0.7596 & 4.5468 & 2.805 & 2.5759 \\ 0.36604 & 4.543 & 3.8124 & 4.7688 & 2.939 \\ 3.1243 & 3.4297 & 3.7292 & 1.8777 & 0.085087 \\ 1.6402 & 3.9494 & 2.7078 & 1.3471 & 2.8531 \end{pmatrix},$$

$$\text{let } \mathbf{x} = \begin{pmatrix} \max\{b_{11}s_1(t), b_{12}s_2(t), ..., b_{15}s_5(t)\} + \epsilon_1(t) \\ \max\{b_{21}s_1(t), b_{22}s_2(t), ..., b_{25}s_5(t)\} + \epsilon_2(t) \\ \vdots \\ \max\{b_{51}s_1(t), b_{52}s_2(t), ..., b_{55}s_5(t)\} + \epsilon_5(t) \end{pmatrix},$$

where $\epsilon_i(t), i = 1, 2, ..., 5$ are normal errors. $\mathbf{x}$ is the simulated observation. The values in the coefficient matrix B are randomly generated from uniform distribution over $[0, 5]$.

Applying MaxICA to $\mathbf{x}$, from PCA, fraction of total variance retained can be the cumulative sum of sorted eigenvalues divided by the total sum of eigenvalues. Figure 4.1 plots the fraction of total variance retained on the vertical axis vs. the number of eigenvalues on the horizontal axis.

From Figure 4.1, five components account for nearly 95 percent of the variance.

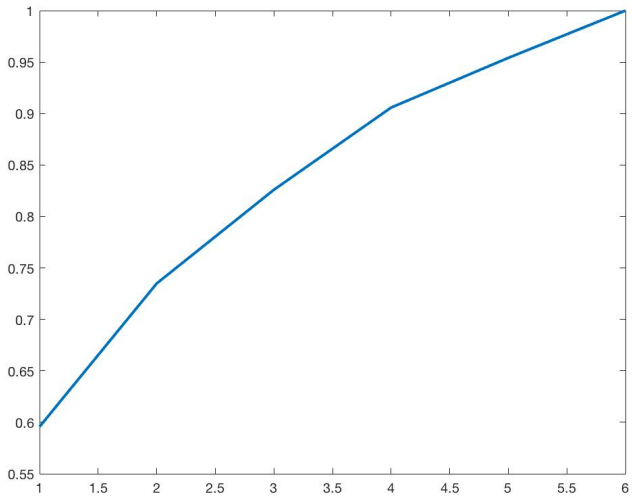


Figure 4.1: *Fraction of total variance retained vs. number of eigenvalues.*

We take five as the number of hidden components. In the data separation step, observations are separated into smaller parts, each part ranges over 500 observation points. Apply MaxICA piecewise, the fitting result is shown in Figure 4.2, red lines are original signals combined by taking maximum, blue lines are fitted lines. This is a good fitting result, blue lines fit almost all of the waves, and for some part of the data, fitting is nearly perfect. Figure 4.3 plots all five original signals $s_1, s_2, ..., s_5$ from top to bottom in order, Figure 4.4 are recovered signals from MaxICA, because of the Maximum operation, there's no order for these five components. The recovered components have similar frequencies and similar amplitudes. As a comparation, Figure 4.5 shows components recovered by ICA, the shape and frequencies of ICA components don't match with the original signals.

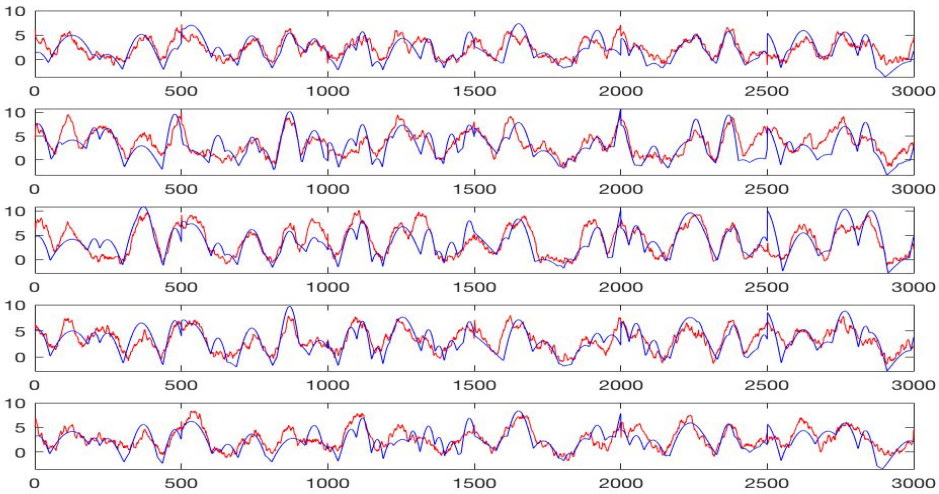


Figure 4.2: *MaxICA fitting result assuming five hidden components.*

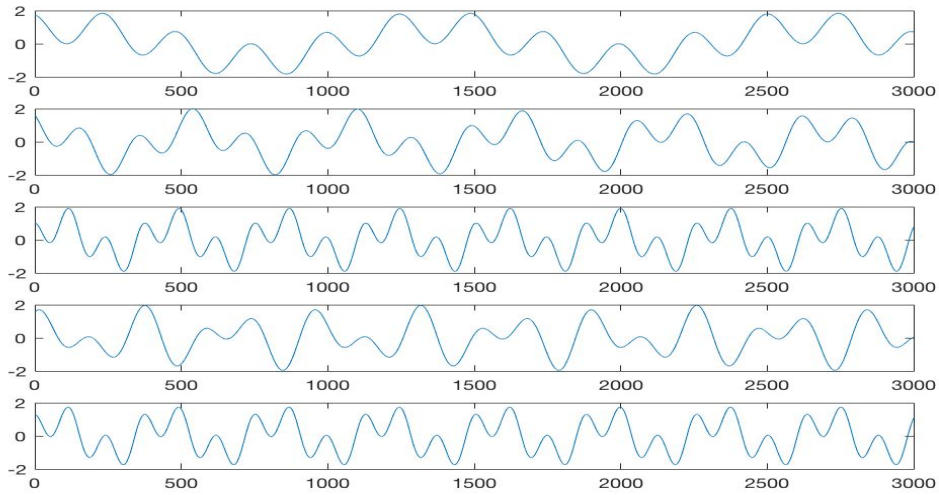


Figure 4.3: *The original components:* $s_1, s_2, ..., s_5$.

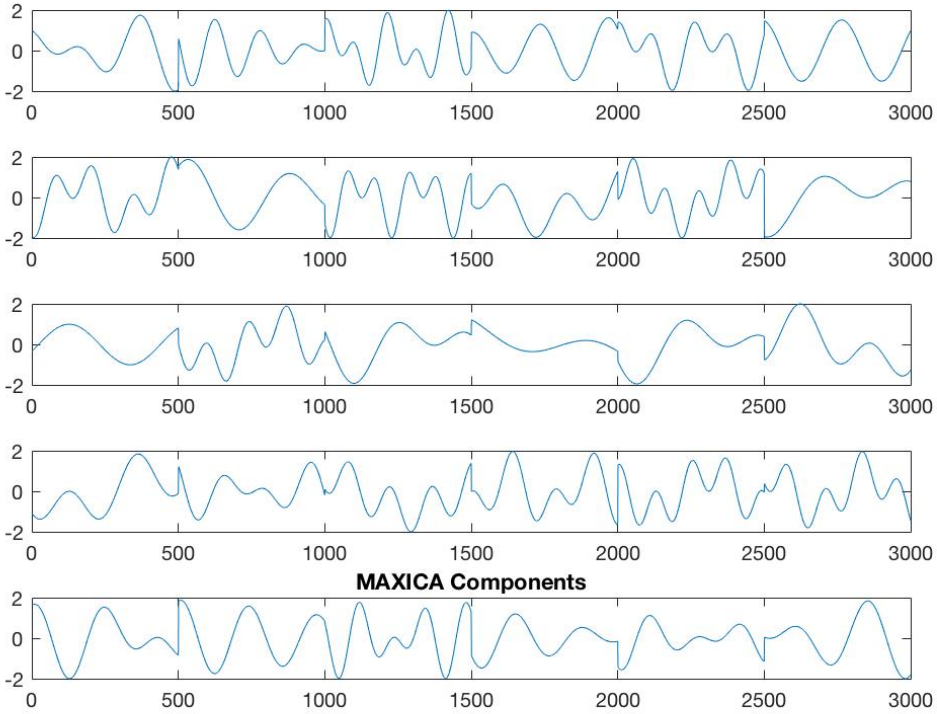


Figure 4.4: *Recovered components from MaxICA.*

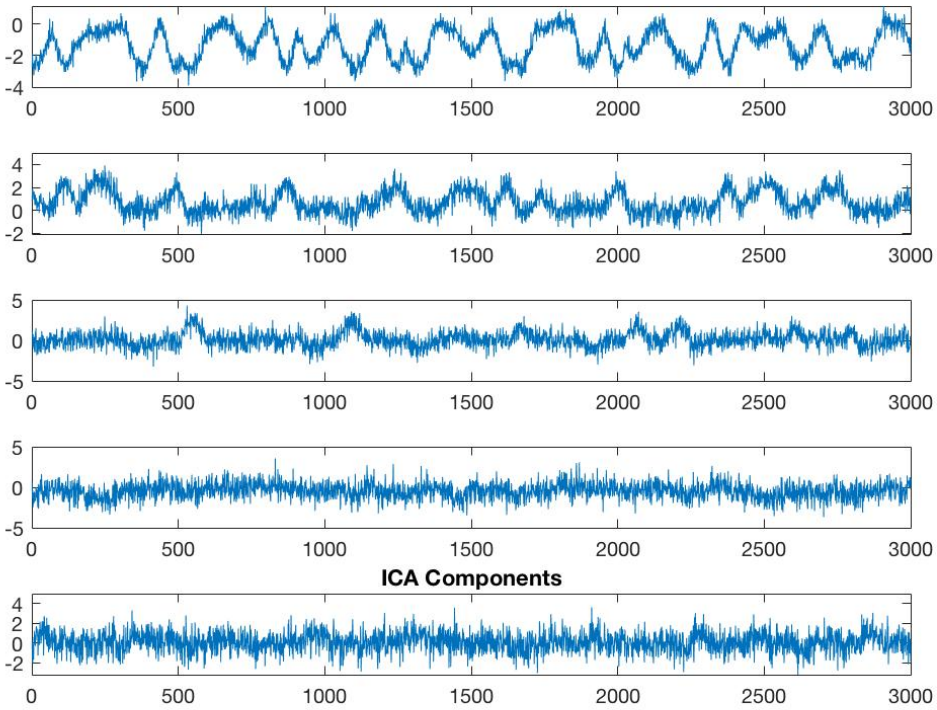


Figure 4.5: *Recovered components from ICA.*

The above results were found based on the assumption that there are five hidden components. If we assume four hidden components, Figure 4.6 shows the fitting result, and if we assume only two hidden components, Figure 4.7 shows the fitting result. By assuming four components, we still can see a roughly good result, but the fitting result of assuming two hidden components doesn't look vey well. From Figure 4.1, four components still account for almost 90% of the variance, but two components account for about 75% of the variance. The fitting matches with this result.

MaxICA uses a genetic algorithm with elite weighted sum selection, combined crossover and dynamic mutation methods. In contrast, the popular genetic algorithm is generaly applied with roulette wheel selection, single point crossover and fixed mutation rate. Figure 4.8 gives the fitting result by the commonly used genetic algorithm. This optimization result was generated by running the same number of iterations as to that by MaxICA in Figure 4.2. With the same number of iterations, the GA with roulette wheel doesn't perform as good as the one with MaxICA. The regular GA needs much more iterations to get a good fitting result. Moreover, the simulated annealing (SA) is also a well known optimization algorithm. Figure 4.9 gives the fitting result by SA. Apparently this result is not as good as the result by MaxICA. Also, the SA algorithm is much more slower than GA, it will take too much time to get a good fitting. Repeating MaxICA 100 times, in Figure 4.10 blue lines are fitted lines given by 100 MaxICA fittings, white lines are the observed signals. Fitted lines all follow the shape and trends of the observed signals.

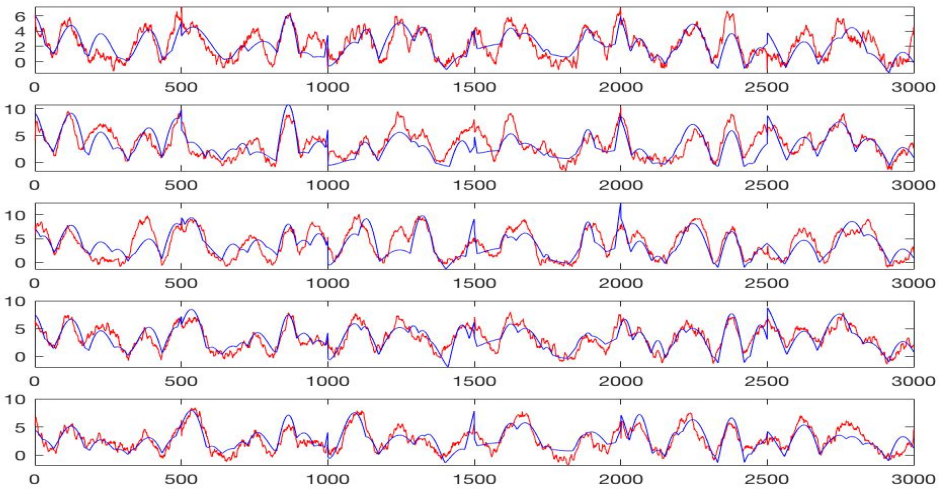


Figure 4.6: *MaxICA fitting result assuming four hidden components.*

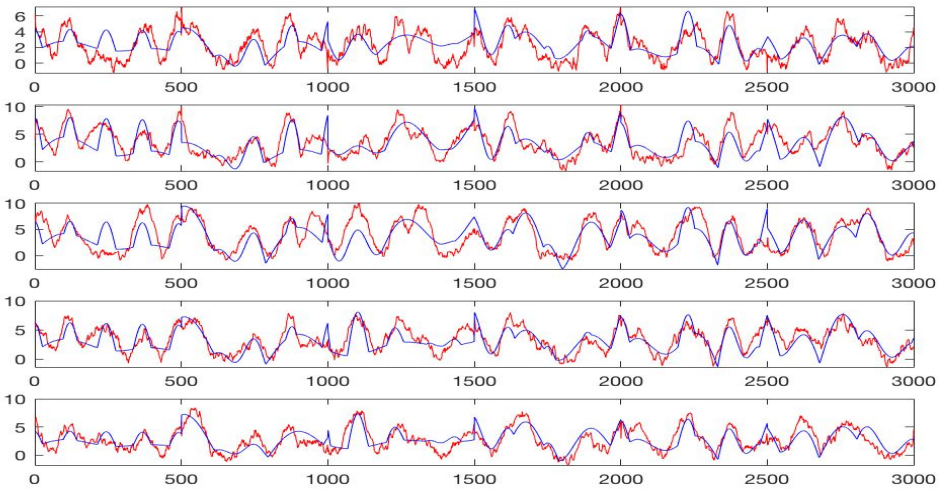


Figure 4.7: *MaxICA fitting result assuming two hidden components.*

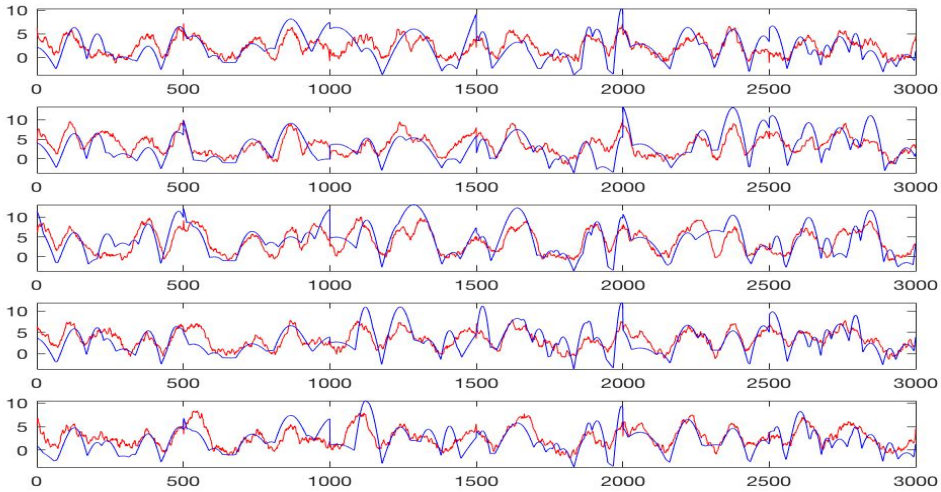


Figure 4.8: *Fitting result using GA with roulette wheel selection, single point crossover and fixed mutation rate.*

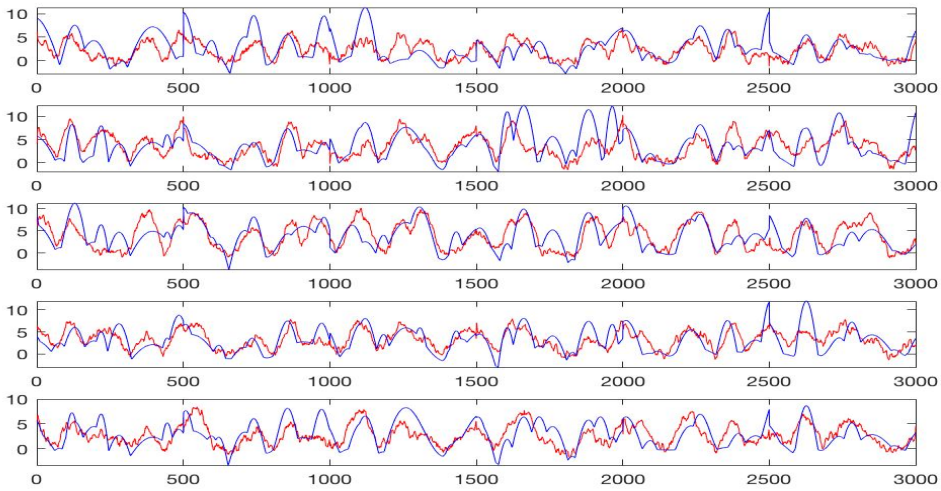


Figure 4.9: *Simulated Annealing(SA) fitting result assuming five hidden components.*

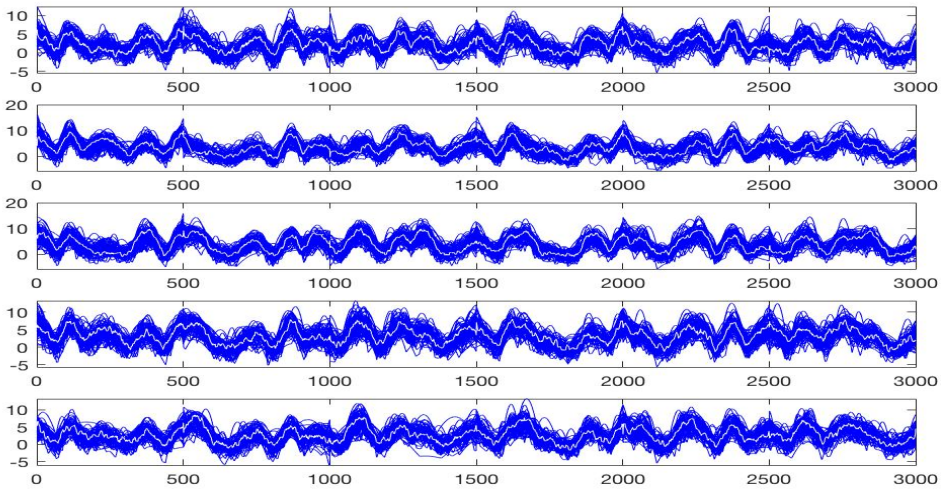


Figure 4.10: *MaxICA fitting 100 times.*

Example 2. In this example, we borrow the results from section (5.1). The **s** in equation (5.1) contains all five hidden components discovered by MaxICA. We randomly generate coefficient matrix B from standard normal distribution to set:

$$B = \begin{pmatrix} b_{11} & \cdots & b_{15} \\ \vdots & \ddots & \vdots \\ b_{51} & \cdots & b_{55} \end{pmatrix} = \begin{pmatrix} 0.7186 & 0.47555 & -0.68076 & -1.7432 & -0.46776 \\ -1.1563 & -0.72816 & -0.93876 & -0.38086 & -0.92032 \\ 0.9143 & -0.020308 & -0.54847 & 0.03899 & 1.5747 \\ 0.84468 & 1.0166 & -0.57345 & -0.39733 & 1.3367 \\ -0.11342 & -0.48642 & -1.3024 & 0.43596 & 0.62172 \end{pmatrix}.$$

The simulated signals: $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_5 \end{pmatrix} = \begin{pmatrix} \max\{b_{11}s_1(t), b_{12}s_2(t), ..., b_{15}s_5(t)\} + \epsilon_1(t) \\ \max\{b_{21}s_1(t), b_{22}s_2(t), ..., b_{25}s_5(t)\} + \epsilon_2(t) \\ \vdots \\ \max\{b_{51}s_1(t), b_{52}s_2(t), ..., b_{55}s_5(t)\} + \epsilon_5(t) \end{pmatrix}$.

We run ICA algorithm on $\mathbf{x}$ to see if ICA can recover the hidden components $\mathbf{s}$. In Figure 4.11, red lines are the hidden components $\mathbf{s}$, blues lines are ICA components generated by FastICA algorithm. Obviously ICA components fail to recover max-linear components. Next, we run MaxICA method on $\mathbf{x}$ to see its performance. Figure 4.12 gives the result, and blue lines are MaxICA components, while red lines are the hidden components. One can see that the fitted components and the original components have similar frequency trend and similar amplitude trend. This result is better than that from ICA algorithm, and the linear model oriented ICA method doesn't work for max-linear model, but from MaxICA, one can recover many significant information about hidden components, such as frequency and amplitude.

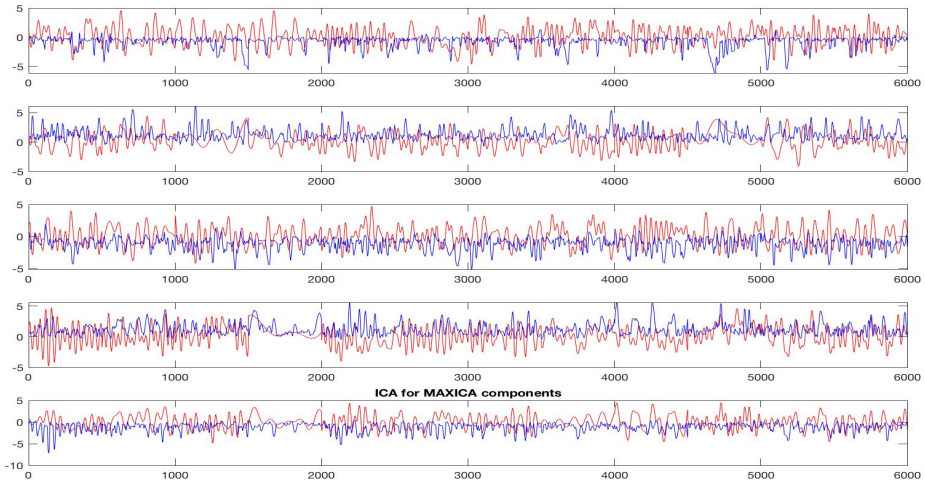


Figure 4.11: *Components given by ICA for simulated signals.*

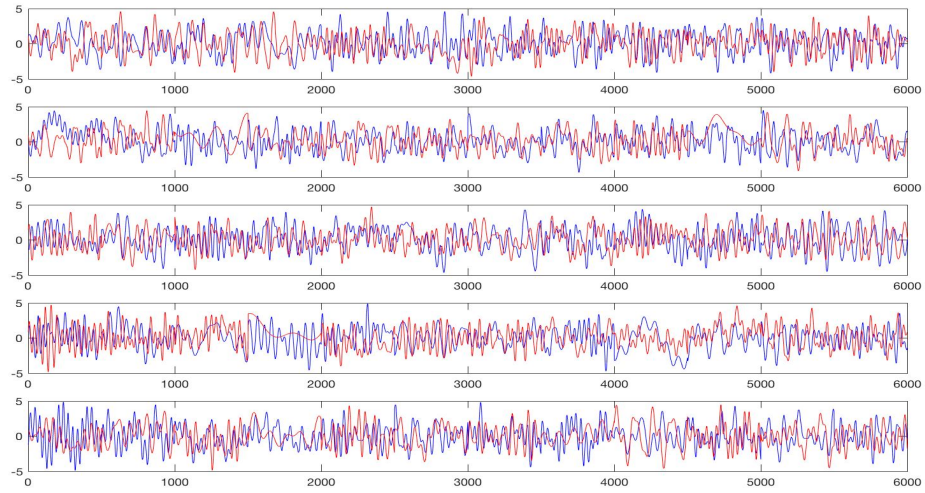


Figure 4.12: *Components given by MaxICA for simulated signals.*

Example 3. Let

$$s_1 = \sin(2t + 0.3),$$

$$s_2 = \sin(0.5t + 1).$$

$e_1, e_2 \sim 0.2N(0,1)$, $\boldsymbol{\epsilon_1} = (e_1, e_2)'$, $e_3, e_4 \sim 0.8N(0,1)$, $\boldsymbol{\epsilon_2} = (e_3, e_4)'$.

$$\mathbf{A_1} = \begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}, \quad \mathbf{A_2} = \begin{pmatrix} 5 & 2 \\ 2 & 8 \end{pmatrix}, \quad \mathbf{S} = (s_1, s_2)'.$$

$$\mathbf{X_1} = \mathbf{A_1 S} + \boldsymbol{\epsilon_1}, \tag{4.1}$$

$$\mathbf{X_2} = \mathbf{A_1 S} + \boldsymbol{\epsilon_2}, \tag{4.2}$$

$$\mathbf{X_3} = \mathbf{A_2 S} + \boldsymbol{\epsilon_1}. \tag{4.3}$$

In this example, s_1, s_2 are linearly combined. Apply ICA and MaxICA respectively to each one of $\mathbf{X_1}$, $\mathbf{X_2}$ and $\mathbf{X_3}$.

Figure 4.13 plots the simulated signals $\mathbf{X_1}$. Figure 4.14 plots the ICA components for $\mathbf{X_1}$, red lines are s_1, s_2, blue lines are ICA components. ICA recovered the components. Figure 4.15 is the fitting result by MaxICA for $\mathbf{X_1}$, red lines are observed signals, blues lines are fitted signals. Figure 4.16 plots the MaxICA components, red lines are s_1, s_2, blues lines are MaxICA components. In this case, MaxICA had a good performance in finding hidden components. The MaxICA components are:

$$\widehat{s_1} = \sin(1.9414t + 1.1308),$$

$$\widehat{s}_2 = \sin(0.50032t + 0.98659).$$

For $\mathbf{X_2}$, error term in equation (4.2) is larger comparing to that in equation (4.1). Figure 4.17 plots the $\mathbf{X_2}$ signals. Figure 4.18 gives ICA components, red lines are s_1, s_2, blue lines are ICA components, one can see ICA components recovered the trends of original signals, but if we give a closer look at the top plot of Figure 4.18, ICA component doesn't follow the shape of the original component. Figure 4.19 gives the fitting result by MaxICA, red lines are observed signals, blues lines are MaxICA fitted signals. Figure 4.20 gives the MaxICA components, red lines are s_1, s_2, blues lines are MaxICA components, MaxICA gave a very good result. The MaxICA components are:

$$\widehat{s}_1 = \sin(2.0033t + 0.23858),$$
$$\widehat{s}_2 = \sin(0.51712t + 0.80748).$$

Signal $\mathbf{X_3}$ are plotted in Figure 4.21. Equation (4.3) shows that $\mathbf{X_3}$ was generated based on $\mathbf{A_2}$, consider the amplitude of s_1, s_2, there are some large values (5 and 8) in $\mathbf{A_2}$ comparing to $\mathbf{A_1}$. Figure 4.22 gives the ICA components, red lines are s_1, s_2, blue lines are ICA components, ICA performs very well. Figure 4.23 gives the fitting result by MaxICA, red lines are observed signals, blue lines are fitted signals, the fitting is not good. Figure 4.24 shows the MaxICA components, red lines are s_1, s_2, blue lines are MaxICA components, on the top plot, MaxICA failed to recover the original component.

As a result, ICA performs very well for linearly combined signals. MaxICA can also recover the information about linearly combined components, even with big errors. But if there's more complex mixing matrix, MaxICA doesn't perform as well as ICA. The application domains for ICA and MaxICA are different, one should not replace ICA by MaxICA, nor replace MaxICA by ICA.

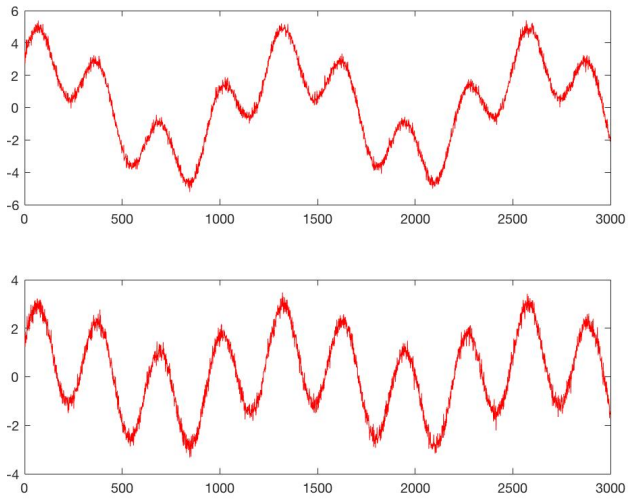


Figure 4.13: *Simulated signals* $\mathbf{X_1}$.

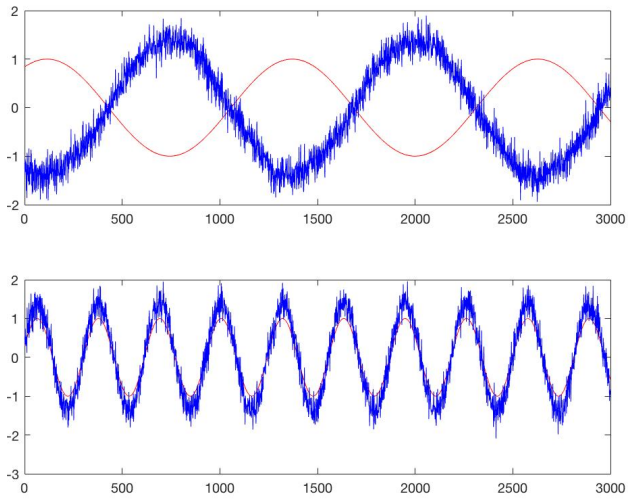


Figure 4.14: *ICA components.*

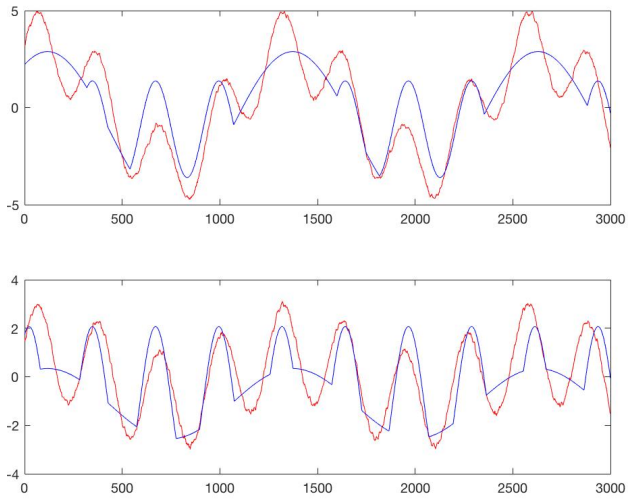


Figure 4.15: *MaxICA fitting result.*

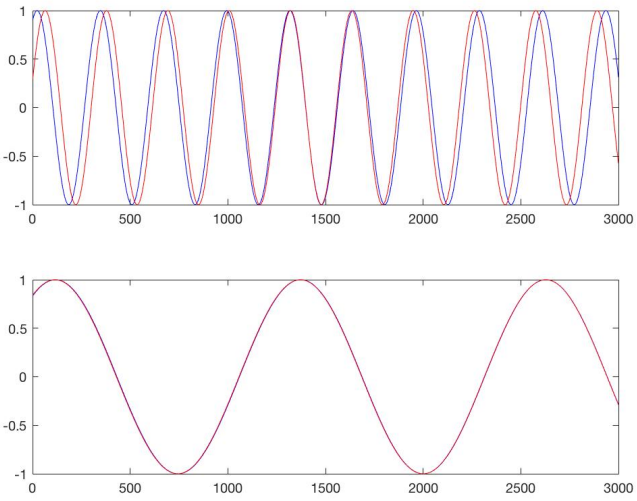


Figure 4.16: *MaxICA components*

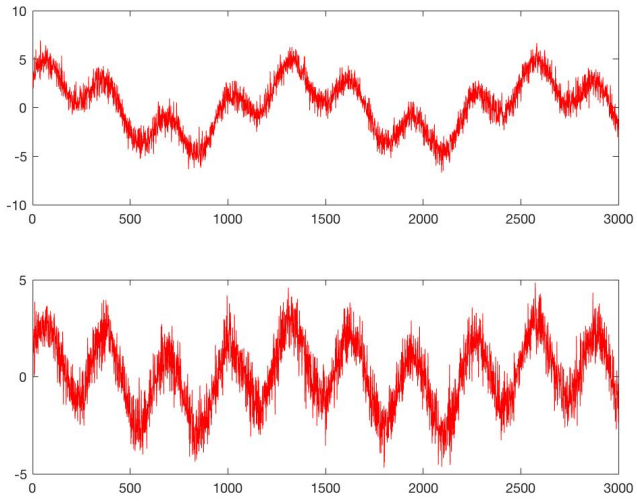


Figure 4.17: *Simulated signals* $\mathbf{X_2}$

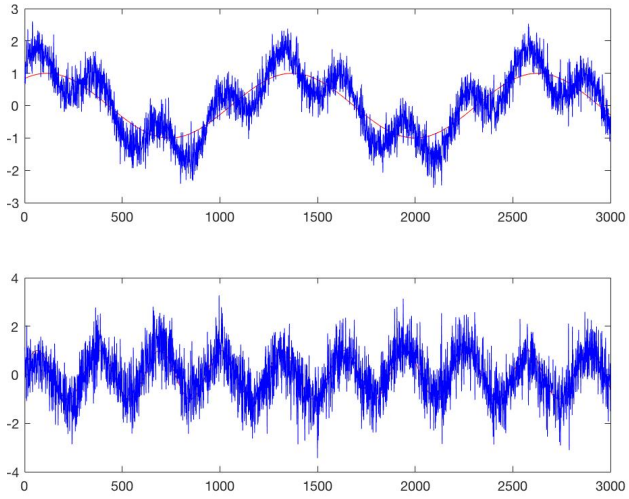


Figure 4.18: *ICA components*

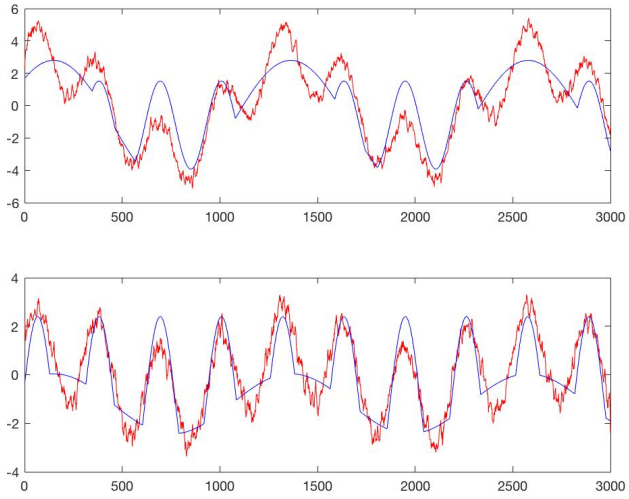


Figure 4.19: *MaxICA fitting result.*

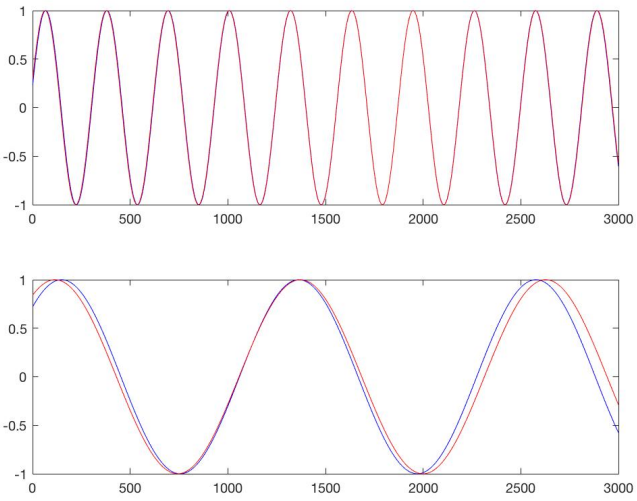


Figure 4.20: *MaxICA components.*

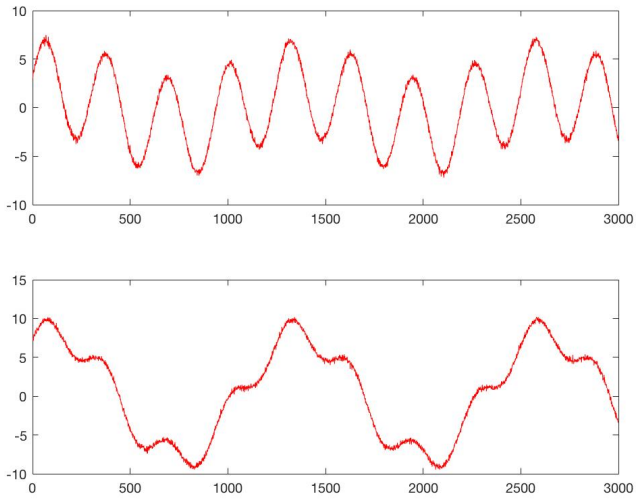


Figure 4.21: *Simulated signals* $\mathbf{X_3}$.

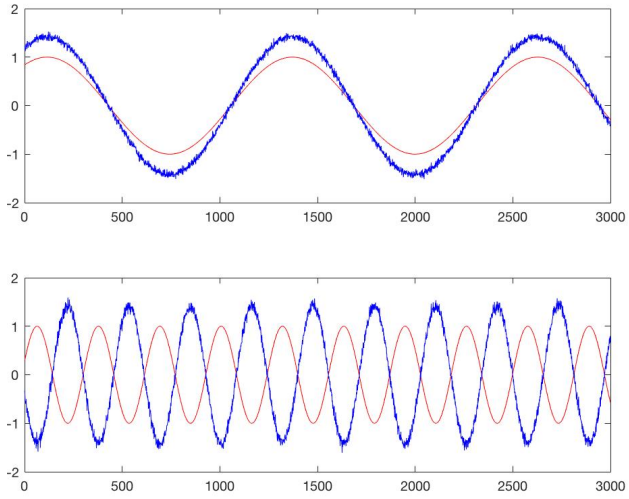


Figure 4.22: *ICA components.*

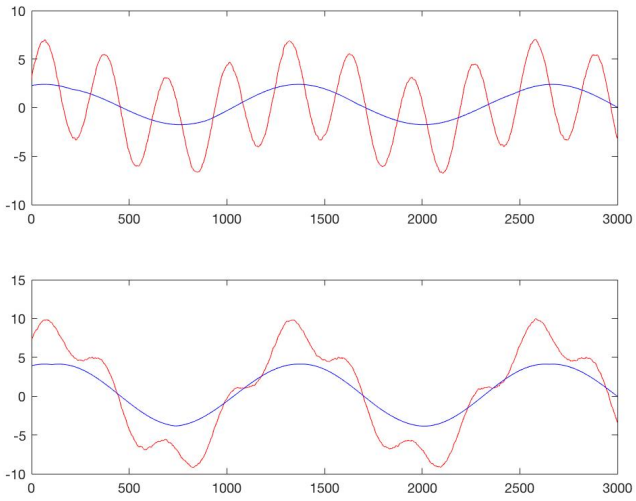


Figure 4.23: *MaxICA fitting result.*

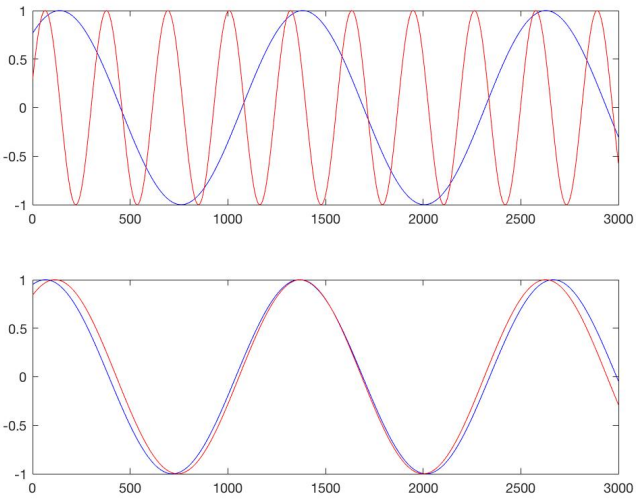


Figure 4.24: *MaxICA components.*

Example 4. Generate five MaxICA component signals based on the data from Section 5.1, Figure 4.25 shows these five signals. In Figure 4.25 from top to bottom, assign these signals to five specific channel locations: FP1, FP2, F3, F4, C3. The locations are shown in Figure 4.26, this figure is a view from top of the head, the circle represents the head, left ear on the left of the circle, right ear on the right of the circle, nose on the top of the circle.

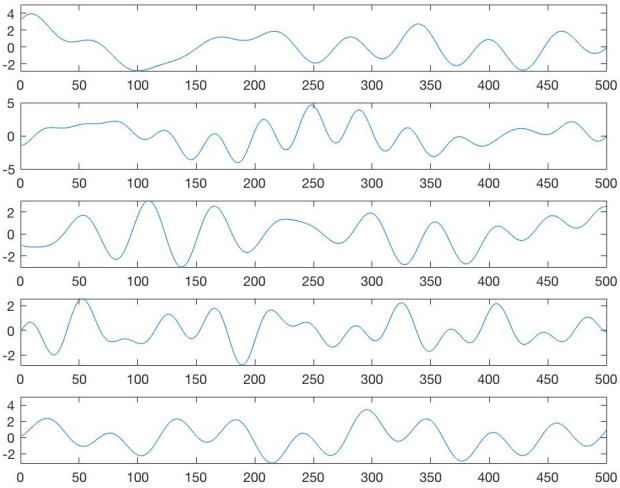


Figure 4.25: *Original components.*

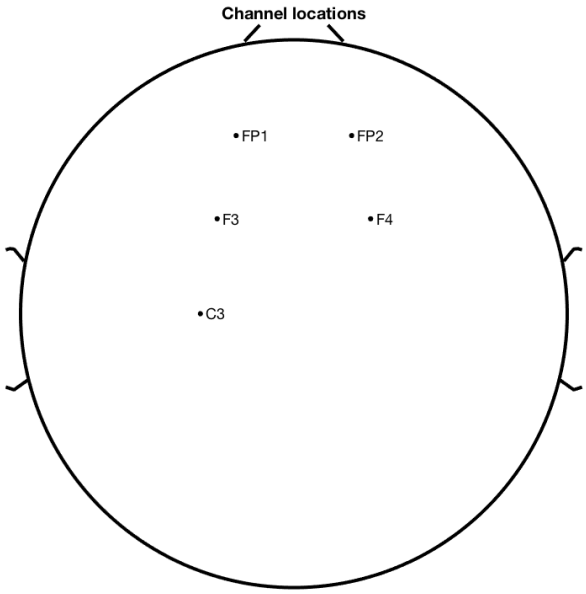


Figure 4.26: *Five channel locations over head.*

To construct brain images for ICA components, let ICA model be $\mathbf{x} = \mathbf{As}$. ICA solve for $\widehat{\mathbf{A}}$ and $\widehat{\mathbf{s}}$ such that $\widehat{\mathbf{A}}\widehat{\mathbf{s}} = \mathbf{x}$, where $\widehat{\mathbf{A}}$ is $N \times N$ mixing matrix, $\widehat{\mathbf{s}} = (\widehat{s}_1, \widehat{s}_2, ..., \widehat{s}_N)^T$ is $N \times m$ matrix, $\widehat{s}_i, i = 1, ..., N$ are estimated source signals. To contrust brain image for component $\widehat{s}_j$, let all other components in $\widehat{\mathbf{s}}$ be zeros. $\widehat{\mathbf{A}}(\mathbf{0}, ..., \mathbf{0}, \widehat{s}_j, \mathbf{0}, ..., \mathbf{0})^T = \mathbf{C}_j$. Brain images can be plotted for any columns of $\mathbf{C}_j$.

To construct brain images for MaxICA components, define notation rMAX, such that

$$\mathbf{x} = rMAX(\mathbf{Bs}) = \begin{pmatrix} \max(b_{11}s_1, \cdots, b_{1N}s_N) \\ \vdots \\ \max(b_{N1}s_1, \cdots, b_{NN}s_N) \end{pmatrix},$$

where $B = \begin{pmatrix} b_{11} & \cdots & b_{1N} \\ \vdots & \ddots & \vdots \\ b_{N1} & \cdots & b_{NN} \end{pmatrix}$ and $\widehat{\mathbf{s}} = (\widehat{s}_1, \widehat{s}_2, ..., \widehat{s}_N)^T$ is $N \times m$ matrix. MaxICA solve for $\widehat{\mathbf{B}}$ and $\widehat{\mathbf{s}}$ such that $rMAX(\widehat{\mathbf{B}}\widehat{\mathbf{s}}) = \mathbf{x}$. To contrust brain image for component $\widehat{s}_j$, let all other components in $\widehat{\mathbf{s}}$ be $b_{ij}\sin(b_{ij})(-\infty)$,

$$rMAX(\widehat{\mathbf{B}}\widehat{\mathbf{s}}) = \begin{pmatrix} \max(\widehat{b}_{11}\text{sign}(\widehat{b}_{11})(-\infty), \cdots, \widehat{b}_{1j}\widehat{s}_j, \cdots, \widehat{b}_{1N}\text{sign}(\widehat{b}_{1N})(-\infty)) \\ \vdots \\ \max(\widehat{b}_{N1}\text{sign}(\widehat{b}_{N1})(-\infty), \cdots, \widehat{b}_{Nj}\widehat{s}_j, \cdots, \widehat{b}_{NN}\text{sign}(\widehat{b}_{NN})(-\infty)) \end{pmatrix} = \mathbf{C}_j.$$

Brain images can be plotted for columns of $\mathbf{C}_j$.

MaxICA component is dynamically changing with time. In order to see how are those component signals changing, plot brain images of each one of the components

at 5 time points: 100ms, 200ms, 300ms, 400ms and 500ms. On the left side of Figure 4.27 is the sequence of brain images for the simulated original signals at 5 different times. On the upper right of Figure 4.27 is the sequence of brain images for observed signals of linearly combined original signals. On the lower right of Figure 4.27 is the sequence of brain images for observed signals of original signals combined based on max-linear assumption.

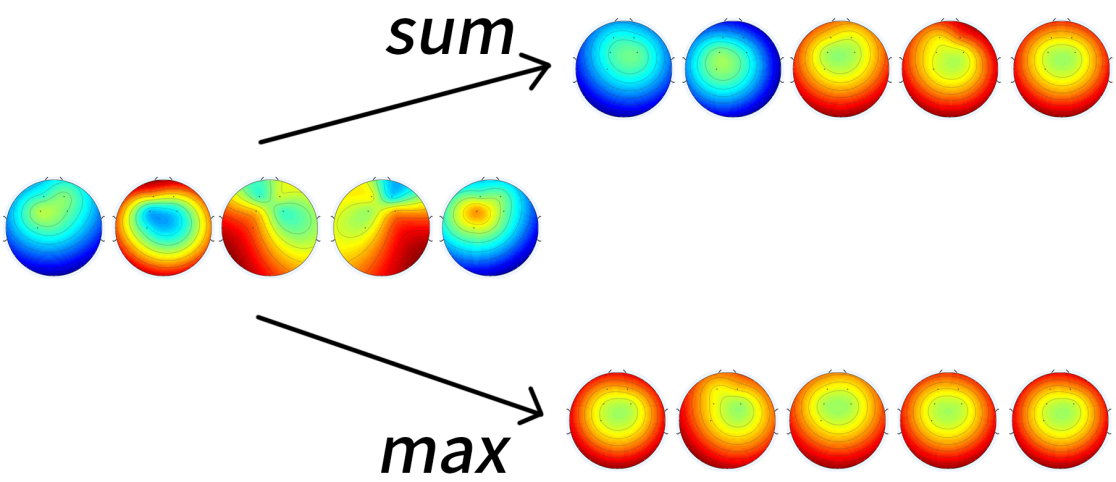


Figure 4.27: *Dynamic brain images of the five components and observed signals based on linear and max-linear assumptions.*

Figure 4.28, 4.29, 4.30 and 4.31 give the process and comparations of MaxICA and ICA based on different assumptions. In Figure 4.28 and Figure 4.29, simulated source signals are on the left hand side. Combine source signals based on max-linear assumption, the observed signals are ploted in the middle with corresponding brain image. In Figure 4.28, we apply ICA to the observations. ICA components are ploted on the right hand side. Brain images of corresponding ICA components are also shown. In Figure 4.29, we apply MaxICA to the observations. MaxICA component

signals with corresponding brain images are shown on the right hand side of the figure. The component recovering result from MaxICA is better than that from ICA. For observations with max-linear assumption, MaxICA performs better than ICA. ICA doesn't work well for max-linear problems.

In Figure 4.30 and Figure 4.31, simulated source signals are on the left hand side. Combine source signals linearly, the observed signals are ploted in the middle with corresponding brain image. In Figure 4.30, we apply ICA to the observations, ICA components with corresponding brain images are plotted on the right hand side. In Figure 4.31, we apply MaxICA to the observations, MaxICA components with corresponding brain images are plotted on the right hand side. The aplitudes and frequencies of ICA components are more similar to the original signals. MaxICA component signals are not as good as that from ICA.

MaxICA and ICA give different results in different situations. They actually have different applications. They should be applied in different areas. MaxICA cannot replace ICA and vice versa.

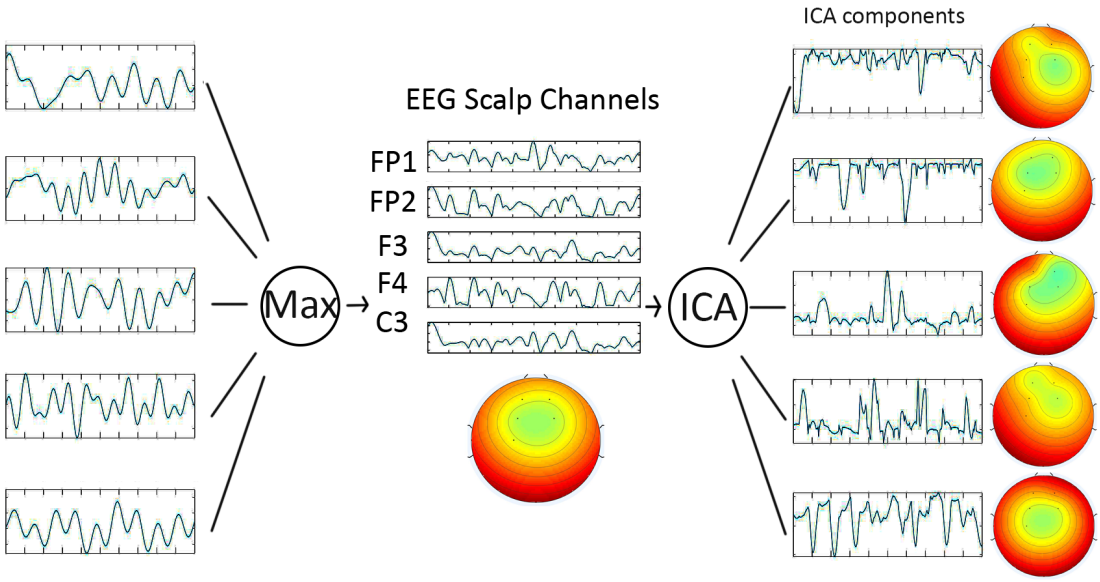


Figure 4.28: *ICA components based on max-linear assumption*

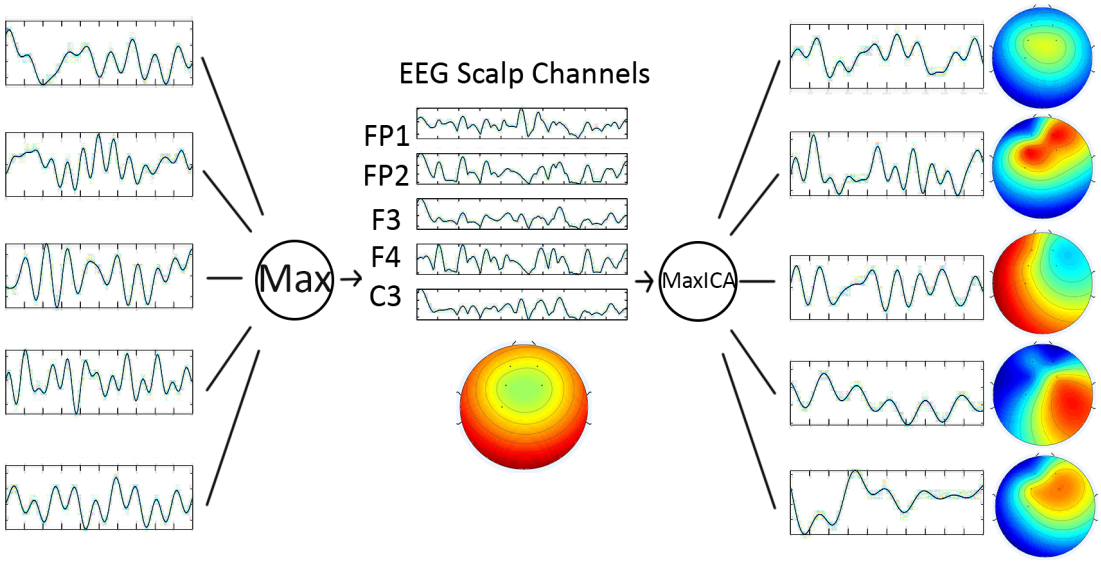


Figure 4.29: *MaxICA components based on max-linear assumption*

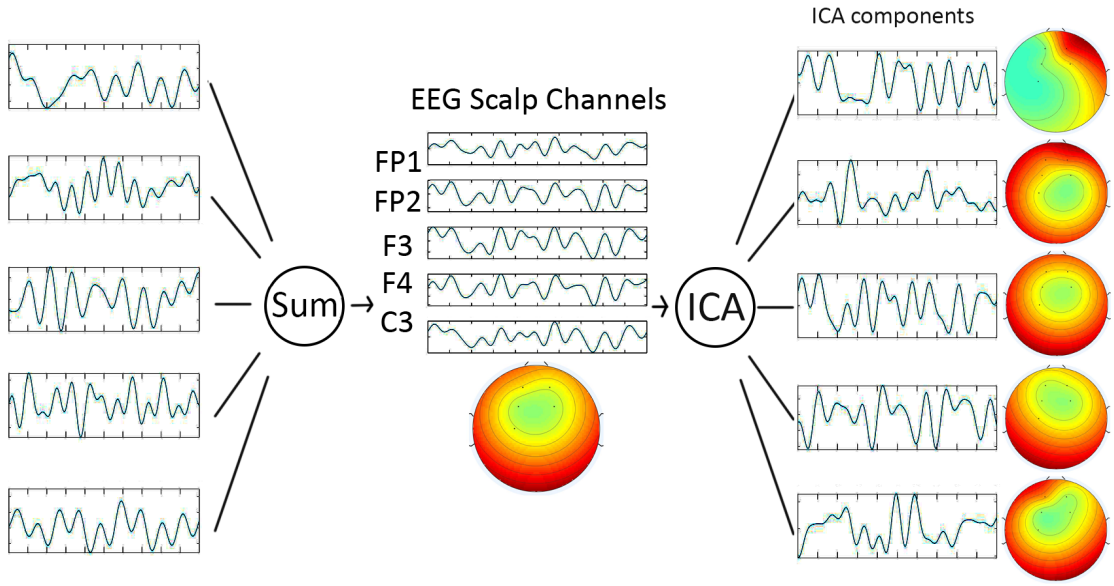


Figure 4.30: *ICA components based on linear assumption*

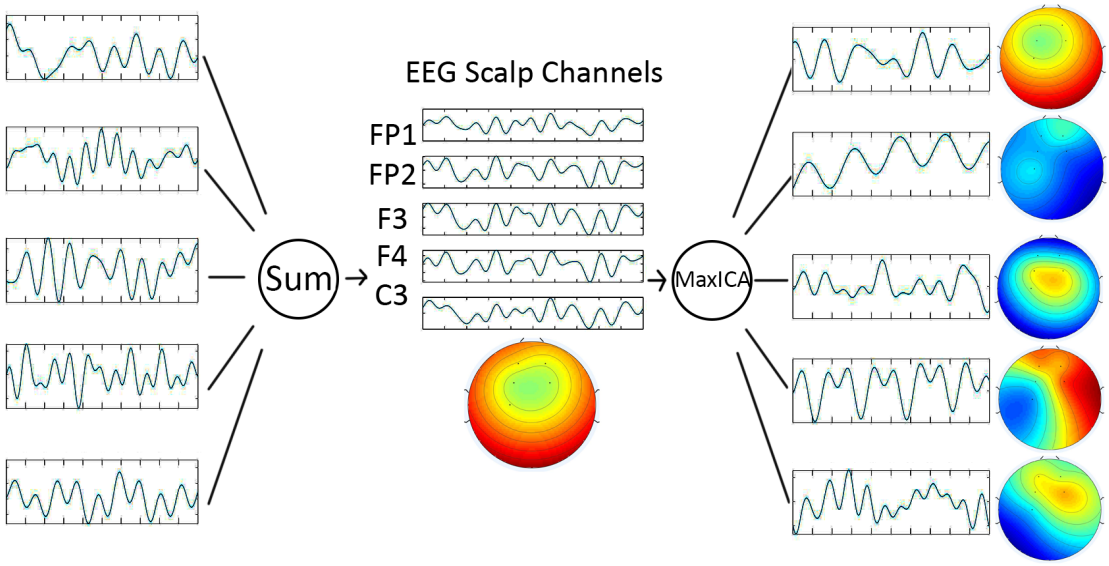


Figure 4.31: *MaxICA components based on linear assumption*

Chapter 5

Real Data Analysis

We present three real data examples. One is visual processing data, another one is epilepsy data and the third one is a motor movement and imagery data. In visual processing data, we apply MaxICA and ICA respectively to see their performances. In epilepsy data, we try to recover hidden brain components to compare different activity patterns of different brain areas. In motor movement and imagery data, we use MaxICA to see brain component activity differences between real movement and imaginary movement.

5.1 Visual Processing Data

This is an EEG (Electroencephalography) data. It's a collection of 32-channel data from 14 subjects (7 males, 7 females) acquired using the Neuroscan software, channel Cz is the reference channel. Subjects participate in two tasks, one is a go-nogo categorization task, the other is a go-no recognition task. Tasks are operated by

presenting natural photographs in front of subjects very briefly (20ms). Each subject responded to 2500 trials. Data is sampled at 1000Hz (Delorme et al., 2002).

In the experiment, participants were seated 110cm away from a computer screen in a dark room. In the two tasks, images were equally likely to be presented to participants. The experiment last two days. There are 13 series on the first day and 12 series on the second day. Each series includes 100 images. The participant need to press a button to start a series. A small point was drawn in the middle of the screen, and the images were presented at the point very fast, participants had to stare at the point. This design can help reduce the effect of eye movement. For each targeted image, participants lift their finger from the button as fast as possible, any missed response was considered as a nogo response.

In the categorization task, for example, participants could be asked to respond to all animals in 100 images mixed by 50 animal images and 50 nonanimal images. All pictures were of natural scenes. Large varieties of pictures were chosen for each image category. The animal category included birds, fishes, reptiles, etc. There were large varieties of non-target images as well, included pictures of natural landscapes, city scenes, food, trees, etc. In the recognition task, participants need to learn the target image in the learning phase in order to recognize it in the following test phase. Participants had no priori information about the pictures. More details about the experiment can be found in (Fabre-Thorpe et al., 2001) and (Delorme et al., 2004).

Electric brain potentials were recorded from 32 electrodes mounted on an elastic cap (Oxford Instruments). Electrode Cz was used as reference and a mastoid electrode was used as ground, so we have 31 EEG observed signals for analysis. Data acquisition

was made at 1000 Hz (corresponding to a sample bin of 1 ms) using a SynAmps recording system coupled with a PC computer.

The electrode locations are shown in Figure 5.1, there are 31 channels in the figure, these 31 signals were used for analysis. The reference electrod Cz is in the center which is not plotted in this figure.

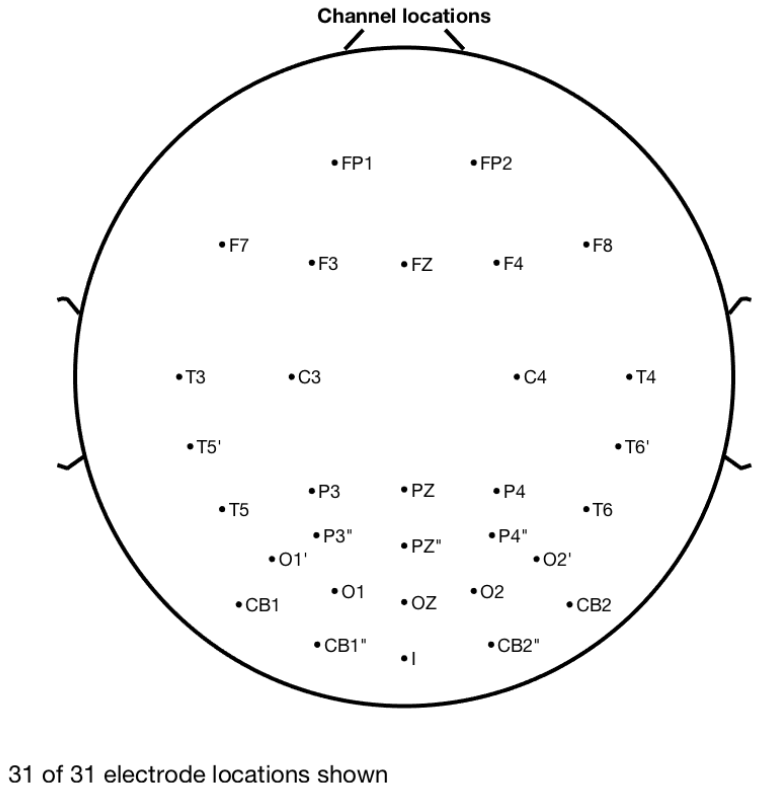


Figure 5.1: *Channel Location*

EEG is person-specific data (Shoeb, 2009), which means EEG of different people varies. Cerebral characteristics could be different between subjects, and Figure 5.2 is the beginning part of all 31 channel data.

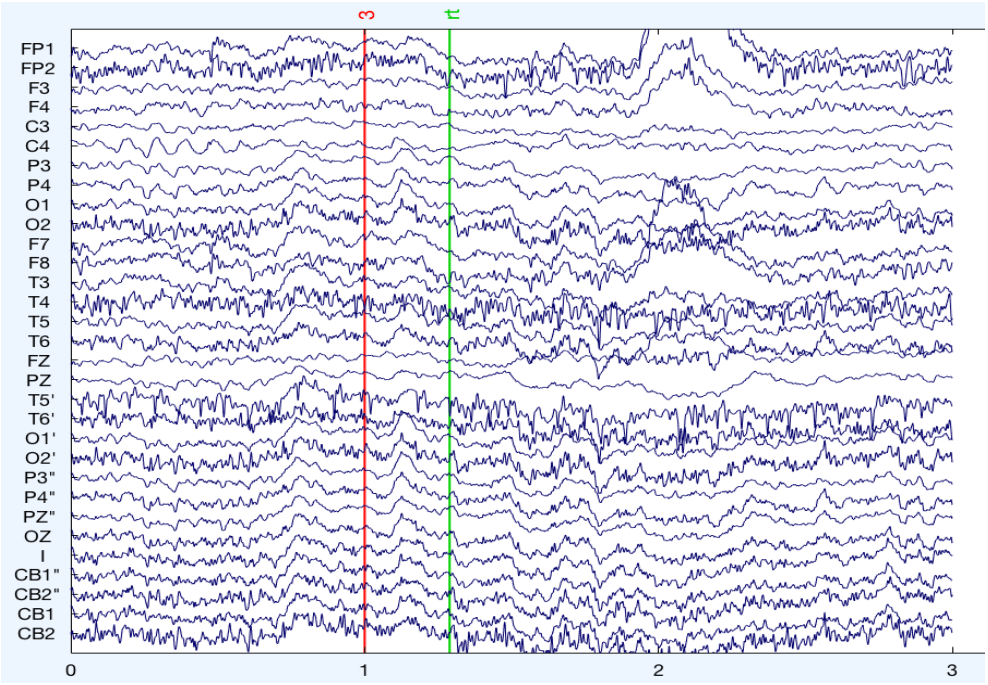


Figure 5.2: *Channel Data*

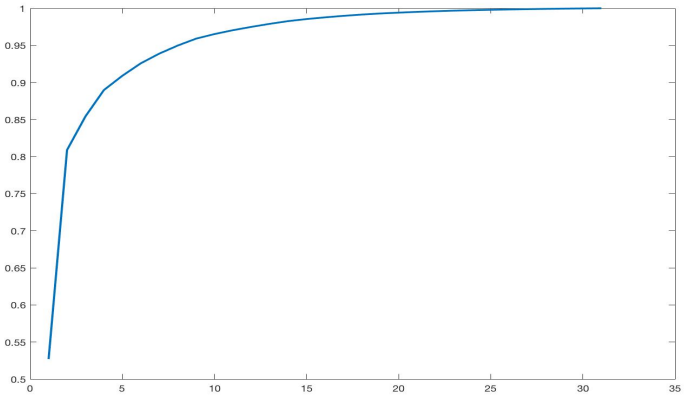


Figure 5.3: *Fraction of total variance vs. number of eigenvalue.*

Apply MaxICA to the data, we first use the categorization task data from the first subject. Figure 5.3 indicates that the first 5 components account for nearly

90% of the variance. So we assume 5 hidden components. We will separate the data into smaller parts, each part ranges 500 observation points, as the data sampled at 1000Hz, so 500 points is 500ms. The fitted model is as follows:

t from 0 to 500ms,

$$
\begin{aligned}
s_1 &= \sin(0.042264t + 185.36) + \sin(0.017198t + 163.07) + \sin(0.091295t + 103.2) \\
&\quad + \sin(0.034458t - 68.199) + \sin(0.17587t - 41.678); \\
s_2 &= \sin(0.091247t - 115.73) + \sin(0.17509t + 148.3) + \sin(0.16988t + 39.285) \\
&\quad + \sin(0.0616t + 167.47) + \sin(0.060204t - 17.111); \\
s_3 &= \sin(0.18037t + 138.79) + \sin(0.039506t + 90.603) + \sin(0.15104t + 38.472) \\
&\quad + \sin(0.11246t + 42.806) + \sin(0.081538t + 110.29); \\
s_4 &= \sin(0.1473t - 34.453) + \sin(0.13189t + 194.5) + \sin(0.16454t - 111.93) \\
&\quad + \sin(0.16945t - 103.56) + \sin(0.17321t - 68.896); \\
s_5 &= \sin(0.020743t - 7.5584) + \sin(0.0008t + 59.59) + \sin(0.0038559t - 104.28) \\
&\quad + \sin(0.1201t + 175.59) + \sin(0.14411t + 20.938).
\end{aligned}
$$

$$
\text{Let } \mathbf{s} = (s_1, s_2, s_3, s_4, s_5)'. \tag{5.1}
$$

$$\text{Coefficient matrix } B = \begin{pmatrix} b_{11} & \cdots & b_{15} \\ \vdots & \ddots & \vdots \\ b_{51} & \cdots & b_{55} \end{pmatrix} = \begin{pmatrix} -2.5923 & 35.433 & 6.6397 & 17.243 & 6.3749 \\ 11.967 & 41.421 & 0.88095 & -13.991 & 11.67 \\ 19.611 & -14.605 & 5.906 & 3.0899 & -20.339 \\ -15.174 & 23.278 & -7.3007 & 2.0725 & 10.633 \\ -8.3165 & -28.103 & -11.426 & 13.544 & -4.045 \end{pmatrix}.$$

For t from 500ms to 6000ms, please refer to appendix A.

$$\text{The fitted model } \mathbf{x} = \begin{pmatrix} \max\{b_{11}s_1(t), b_{12}s_2(t), ..., b_{15}s_5(t)\} + \epsilon_1(t) \\ \max\{b_{21}s_1(t), b_{22}s_2(t), ..., b_{25}s_5(t)\} + \epsilon_2(t) \\ \vdots \\ \max\{b_{51}s_1(t), b_{52}s_2(t), ..., b_{55}s_5(t)\} + \epsilon_5(t) \end{pmatrix}.$$

Figure 5.4 gives a partial fitting result, this plot shows the fitting of the first 6000 points from the first two signals. Blue lines are MaxICA fitted line, red lines are observed signals. The plot indicates a good fit, especially for the big peaks in the signals. MaxICA gives a very good fitting about the big events in the signals. Those peaks are related to important brain activities. The components recovered by MaxICA could be scientifically meaningful. Figure 5.5 gives the fitting result by ICA. Blue lines are ICA fitted lines, red lines are observed signals. ICA also gives good fit. Figure 5.6 plots the five hidden signals recovered by MaxICA. The frequencies and patterns of these components can be helpful for revealing the activity of brain. As a comparation, components recovered by ICA are plotted in Figure 5.7. ICA is based on linear assumption, as in practice, major signals could cover trivial signals, so MaxICA could be more appropriate for max-linear assumption. As a comparation, Figure 5.8 gives the fitting result of this data using Simulated Annealing method. Apparently

this is not as good as the result from MaxICA. Actually, simulated annealing will take much longer time to find the global optimum, for the same time period, MaxICA performs much better than SA. If we take a look at other data, Figure 5.9 gives the fitting result of subject 1's recognition task data, Figure 5.10 gives the fitting result of subject 2's categorization task data, for both of these data, MaxICA fitting performs very well. MaxICA is very good at fitting important events.

To see fitting performance of MaxICA on more data, we apply MaxICA to the recognition task data of the first subject. Figure 5.11 gives the fitting result. Red lines are observed signals, blue lines are MaxICA fitted signals. One can see that the fitted signals follow the major trend of observed signals. And there's a significant turbulence occurs from point 5000, MaxICA fitted lines still can follow the trend even with large noise. Next we apply MaxICA to the categorization task data of the second subject. Figure 5.12 gives the fitting result. Red lines are observed signals, blue lines are MaxICA fitted signals. The fitting is still good, fitted signals not only follow the trend of the original signals, it also gives a good fit for many significant peaks in the original signals. The fitting performance of MaxICA is good.

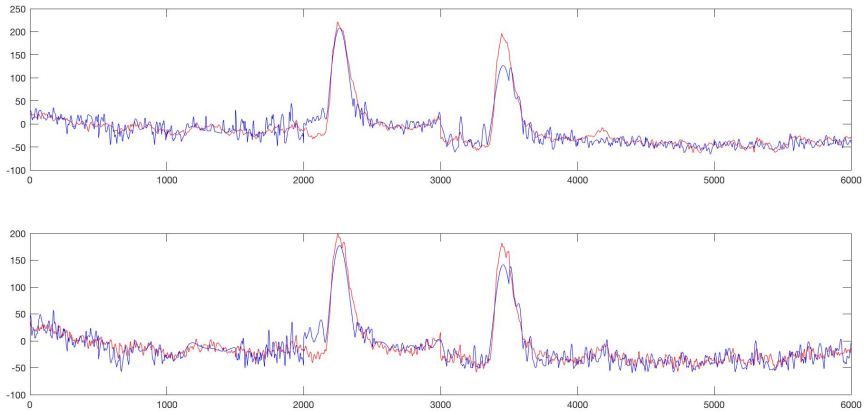


Figure 5.4: *MaxICA Fitting result of subject1's categorization task.*

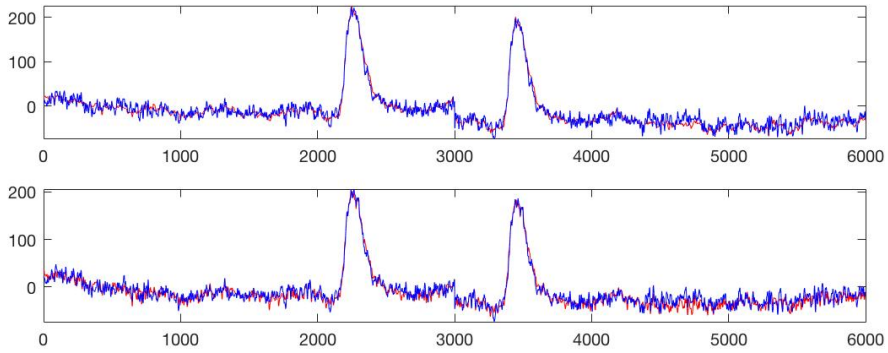


Figure 5.5: *ICA Fitting result of subject1's categorization task.*

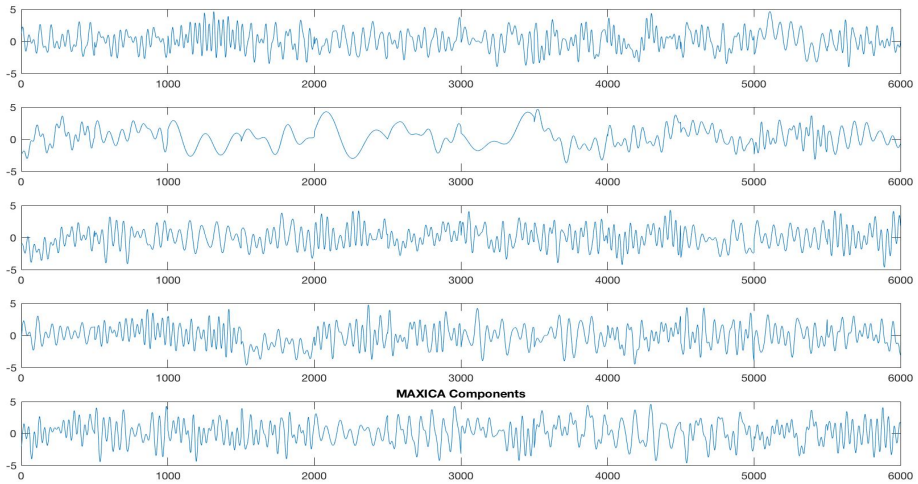


Figure 5.6: *Recovered components by MaxICA*

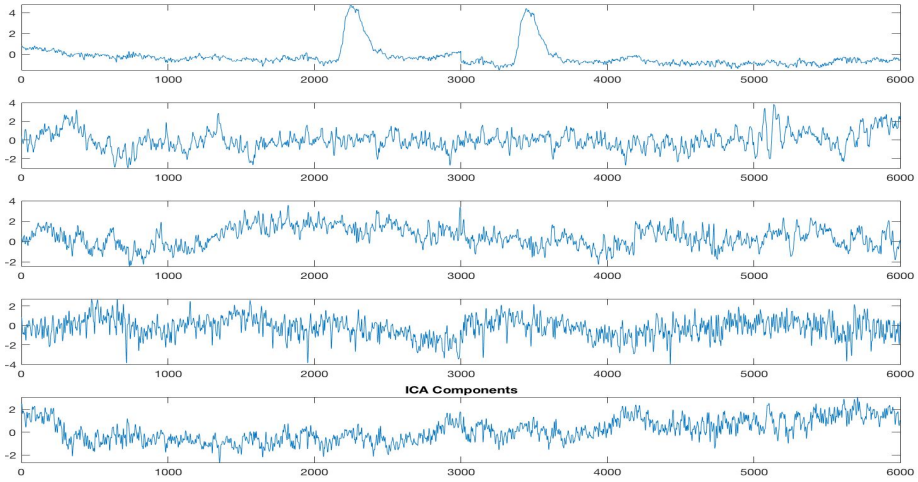


Figure 5.7: *Recovered components by ICA*

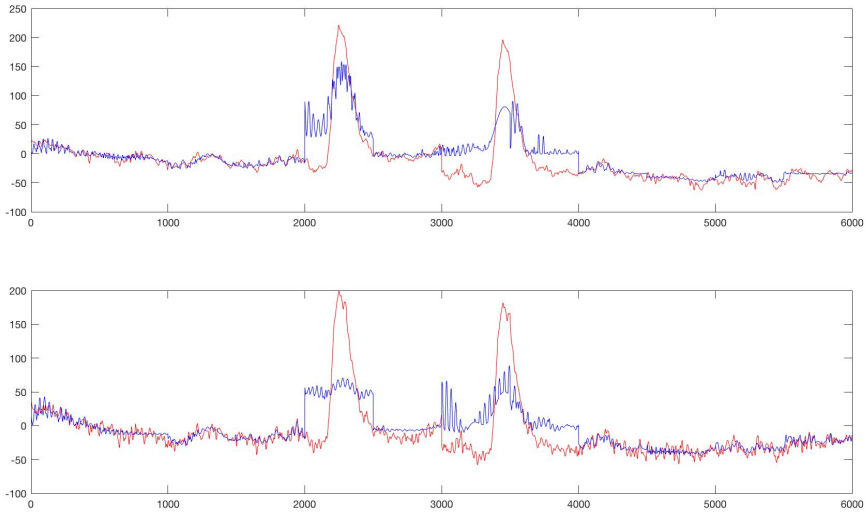


Figure 5.8: *Fitting result of subject1's categorization task using Simulated Annealing.*

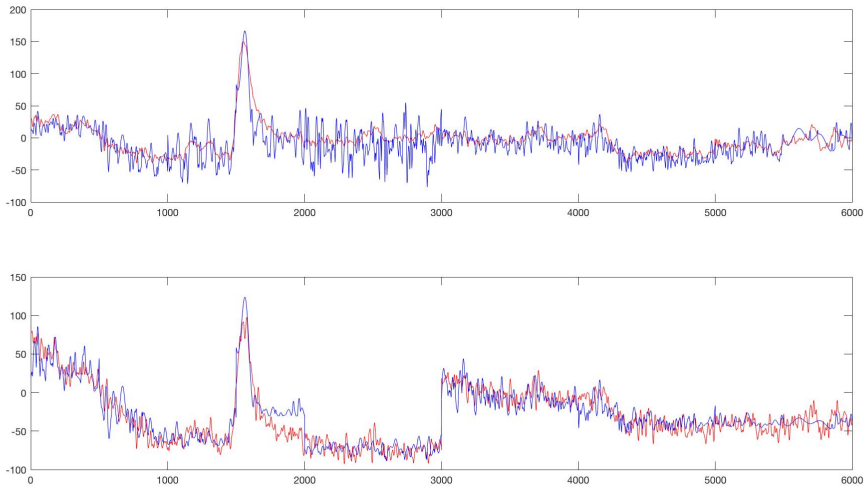


Figure 5.9: *Fitting result of subject1's recognition task.*

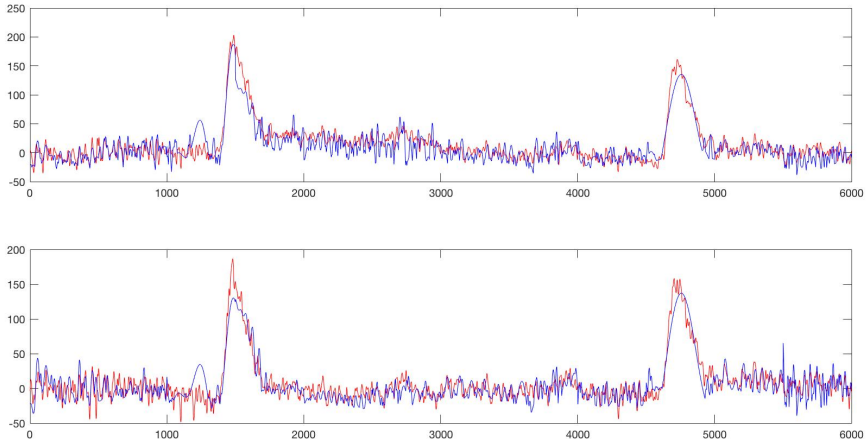


Figure 5.10: *Fitting result of subject2's categorization task.*

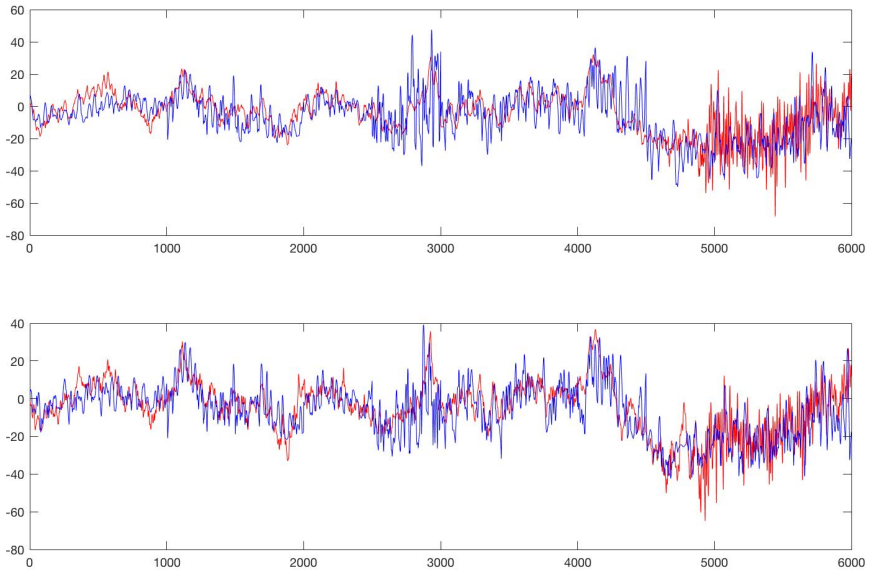


Figure 5.11: *Fitting result of subject1's recognition task.*

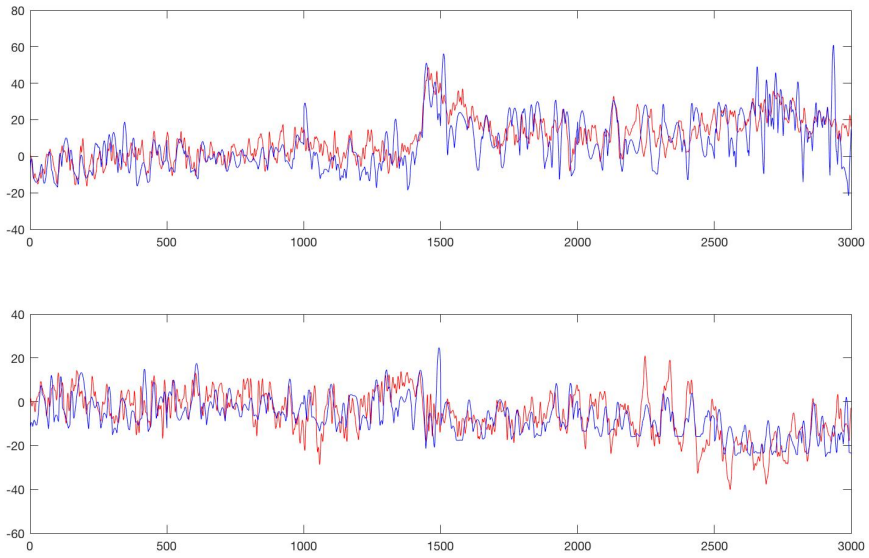


Figure 5.12: *Fitting result of subject2's categorization task.*

5.2 Epilepsy Data

In this section, we will study the intracranial EEG recordings from five epilepsy patients (Andrzejak et al., 2012). These data were performed for the diagnostics of these patients. All patients had longstanding pharmacoresistant temporal lobe epilepsy and they were candidates for epilepsy surgery. All patients underwent long-term intracranial EEG recordings in the Department of Neurology at the University of Bern. The brain areas where seizures started could be localized for all patients. Also these areas were found in the brain that could be surgically resected and all patients had good surgical outcome. Three of them attained complete seizure freedom, the other two only had auras but no other seizures following surgery.

All EEG signals were digitally band-pass filtered between 0.5 and 150 HZ. All channels that detected first ictal EEG signal changes as judged by visual inspection by at least two neurologists who are also board-certified electroencephalographers. Though visual analysis is not a perfect method, but it is still the most important method for helping clinical diagnostic. These channels were classified as focal EEG channels, while all other channels in the recordings were classified as nonfocal EEG channels. Pairs of simultaneously recorded signals were randomly selected from focal and nonfocal EEG channels respectively, the rule of selection refers to (Andrzejak et al., 2012).

First we analyze a signal pair from focal channels. Figure 5.13 indicates that by PCA, we assume two hidden components, actually this is a pair of signals, so this step is not quite necessary. Figure 5.14 gives the fitting result by MaxICA, we can see a good fit, blue lines are fitted lines, red lines are observed signals. The fitting recovers almost all major peaks. Figure 5.15 plots the components recovered by MaxICA and Figure 5.16 plots the components recovered by ICA. The components generated by the two methods appear to have similar characteristics, frequencies, amplitudes, and trending. We next use a pair of EEG signals from nonfocal signals, Figure 5.17, gives the fitting result by MaxICA, and Figure 5.18 and Figure 5.19 show components revealed by MaxICA and ICA respectively. Similar to focal data, for nonfocal data we also see very good fitting, and two different methods both give similar components.

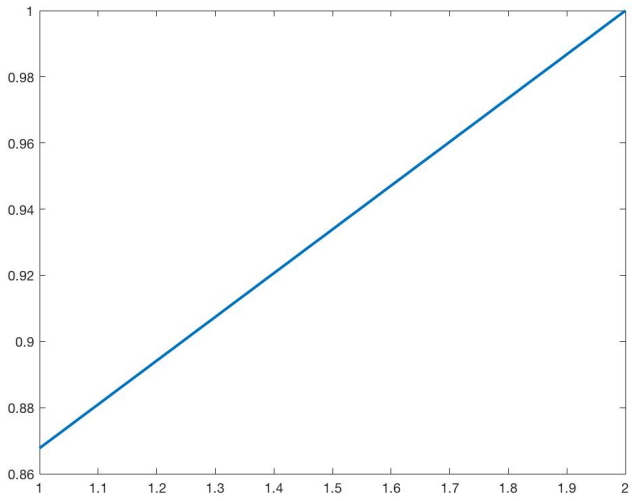


Figure 5.13: *Fraction of total variance vs. number of eigenvalue*

Next we apply MaxICA to other pairs. For another signal pair from focal channels, using MaxICA, the fitting result is given in Figure 5.20. For another signal pair from nonfocal channels, using MaxICA, Figure 5.21 gives the fitting result. In both figures, red lines are observed signals, blue lines are fitted signals. One can see that the fitting results are all very good. Blue lines follow the trends of observed signals, and recover the most of the major peaks.

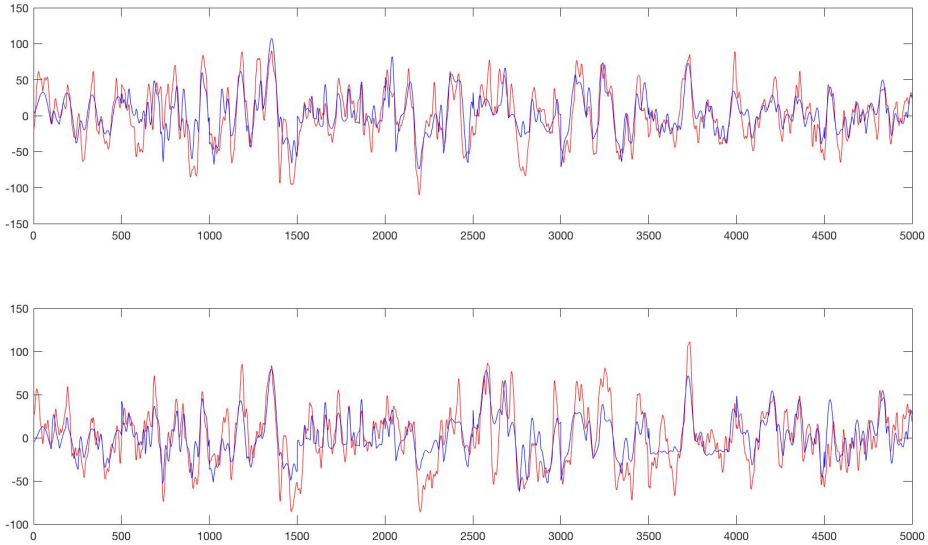


Figure 5.14: *Fitting result of EEG pair from focal channels using MaxICA*

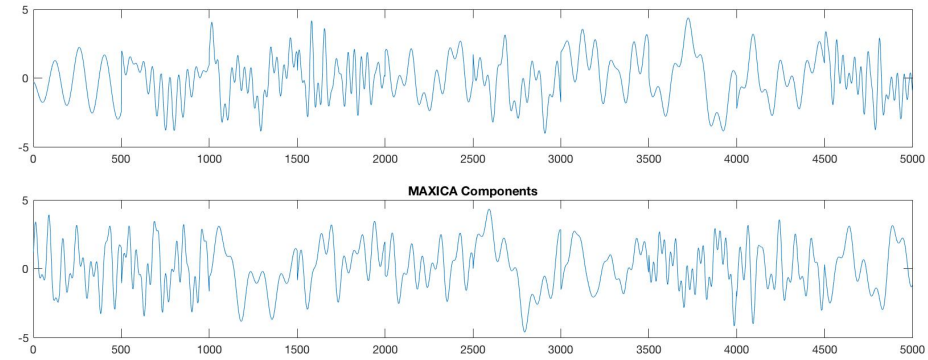


Figure 5.15: *MaxICA components*

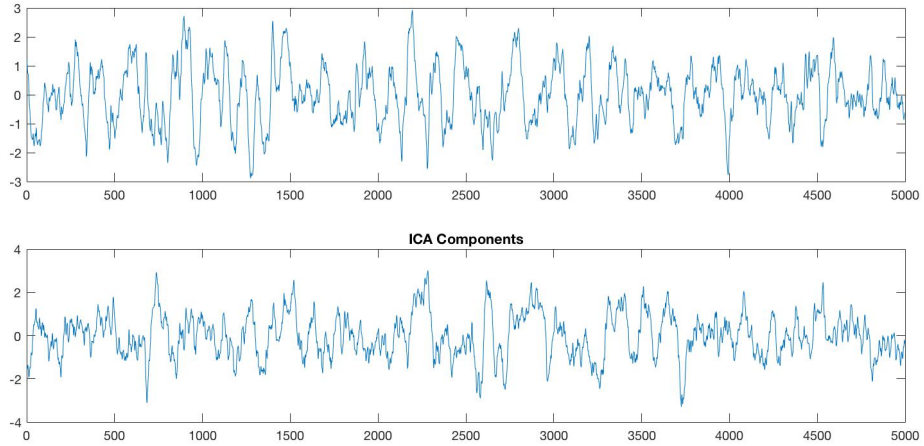


Figure 5.16: *ICA components*

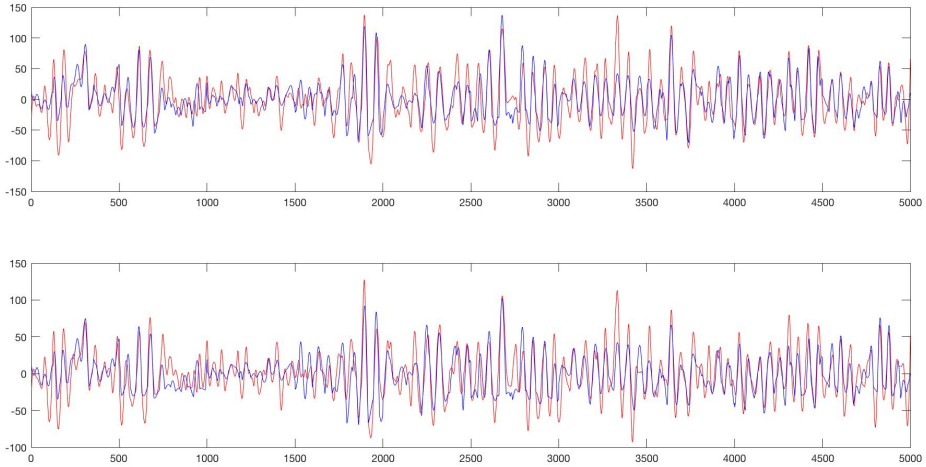


Figure 5.17: *Fitting result of EEG pair from nonfocal channels using MaxICA*

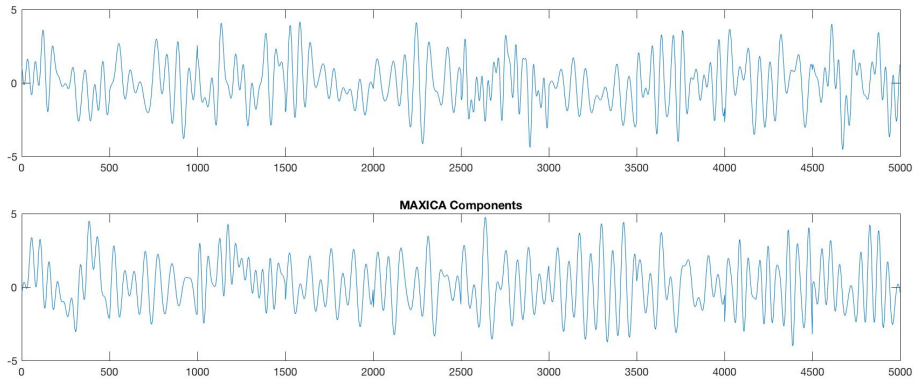


Figure 5.18: *MaxICA components*

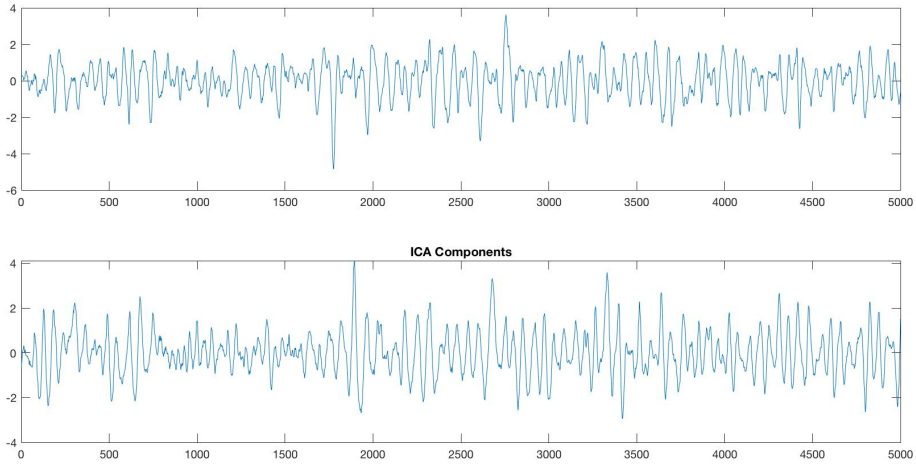


Figure 5.19: *ICA components*

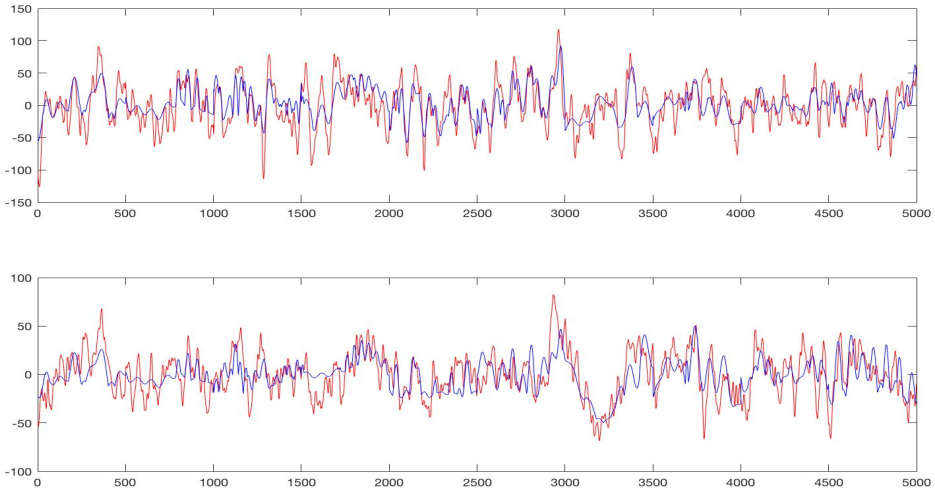


Figure 5.20: *Fitting result*

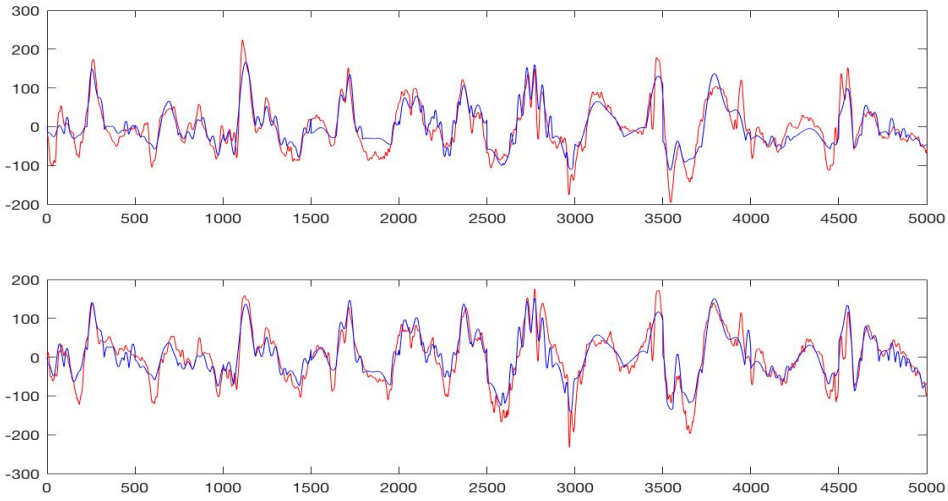


Figure 5.21: *Fitting result*

5.3 Motor Movement and Imagery Data

This is a dataset created by the developers of the BCI2000 instrumentation recording system. This system is described in (Schalk et al., 2004). The dataset contains EEG signals of volunteers. The experiment includes multiple tasks. One task is that the subject opens and closes one of the fists during a two-minute period, this is called the movement group. Another task is that the subject imagines opening and closing one of the fists during a two-minute period without actually open and close the fist, this is called the imagery group. A baseline signal is recorded as the control group while subject relaxes without any movement. More details about the data can be found in (Goldberger et al., 2000).

Applying MaxICA to the control group, Figure 5.22 indicates that five major components account for about 90 percent of the variation, so we consider five hidden components. Figure 5.23 gives the MaxICA fitting result, the fitting is good. Figure 5.24 gives the MaxICA component plot of the five components, the frequencies and amplitude changes looks stable. Applying MaxICA to the experimental groups. Figure 5.25 gives PCA component plots of the movement group on the left and the imagery group on the right. Both plots show that five is a good choice of the number of hidden components. Figure 5.26 is the MaxICA fitting result of the movement group, Figure 5.27 is the MaxICA fitting result of the imagery group. The fittings are good. Figure 5.28 gives the MaxICA component plot of the movement group and Figure 5.29 gives the MaxICA component plot of the imagery group. These component plots have similar patterns, the frequency changes are both significant. Comparing to the control group, these component plots of experimental groups are

both unstable.

Based on the analysis, one can see that real movement and imaginary movement both have effect on human brain, and the ways they affect the brain signal are possibly similar. One can see that if we want to remove the artificial effect by body movement from the brain signal, the subject not only have to be stationary, but also need to relax the brain without even imagine any movement. More study need to be done in the future such as experiments on big movement and small movement.

As a comparation, use ICA to recover hidden components. Figure 5.30, Figure 5.31 and Figure 5.32 give ICA components for control group, movement group and imagery group respectively. Comparing with the control group, ICA components from movement group and imagery group have more unstable pattern, but not very significant. MaxICA might be a better choice for this data analysis.

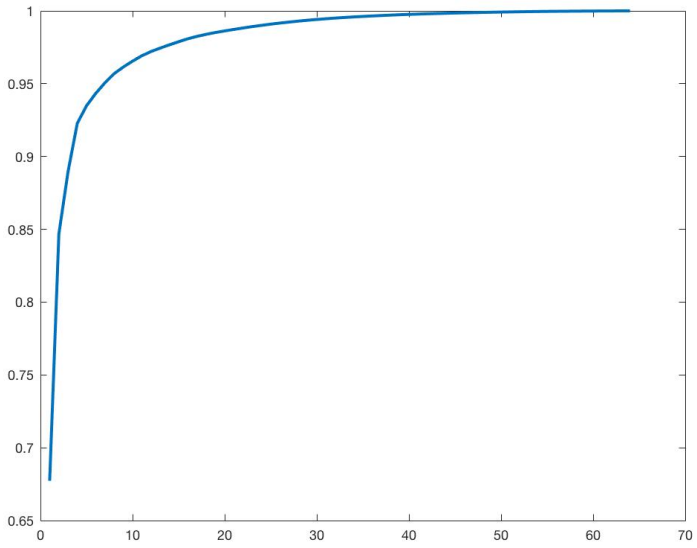


Figure 5.22: *PCA component plot of the control group*

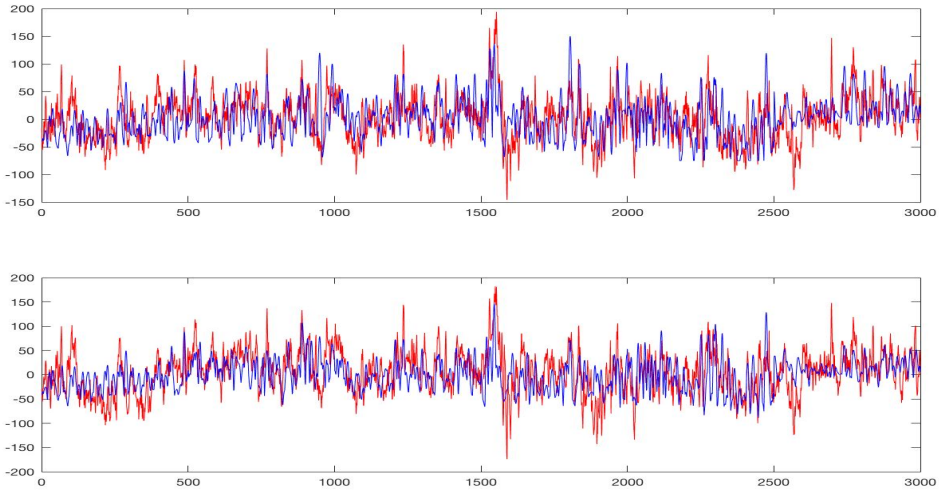


Figure 5.23: *Fitting result of the control group*

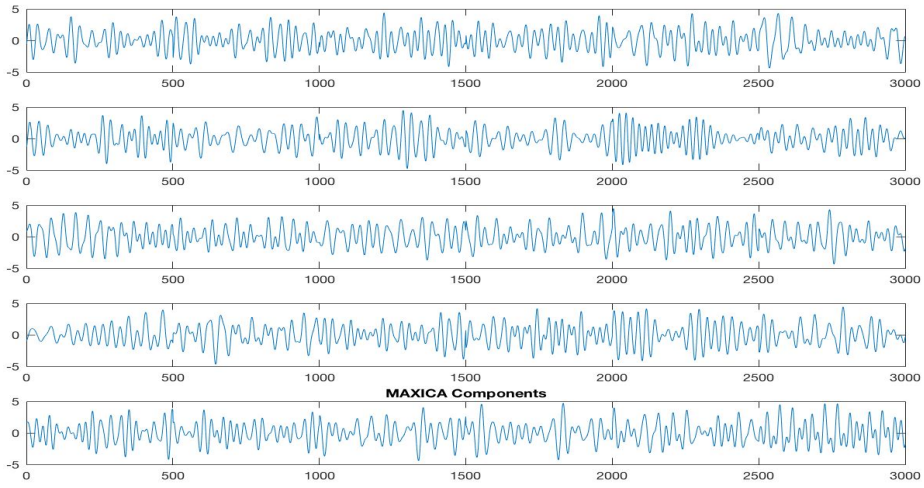
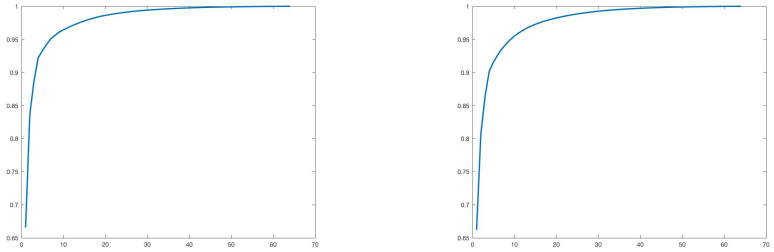


Figure 5.24: *MaxICA components of the control group*



(a) Movement group (b) Imagery group

Figure 5.25: PCA component plot of different groups

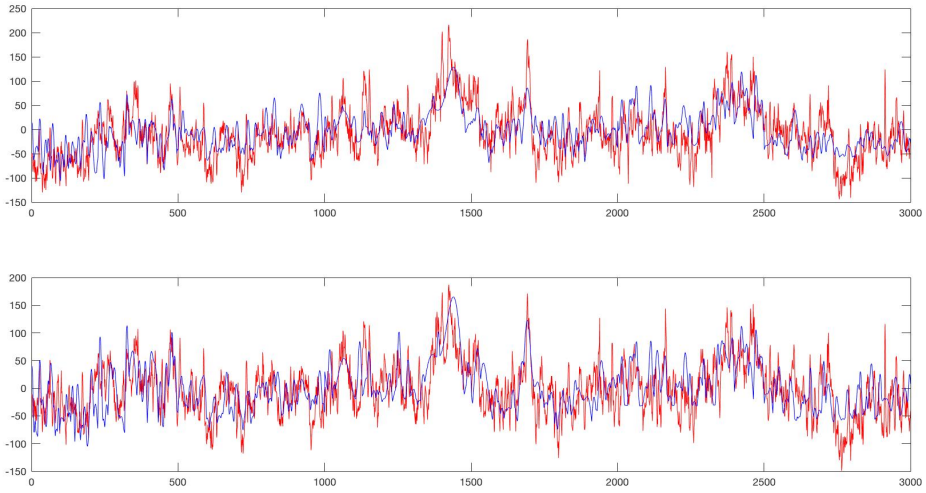


Figure 5.26: *Fitting result of the movement group*

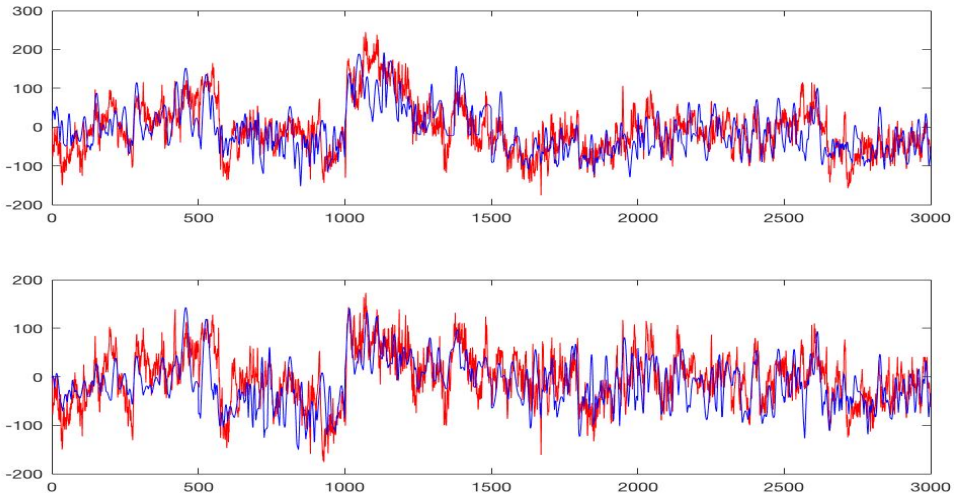


Figure 5.27: *Fitting result of the imagery group*

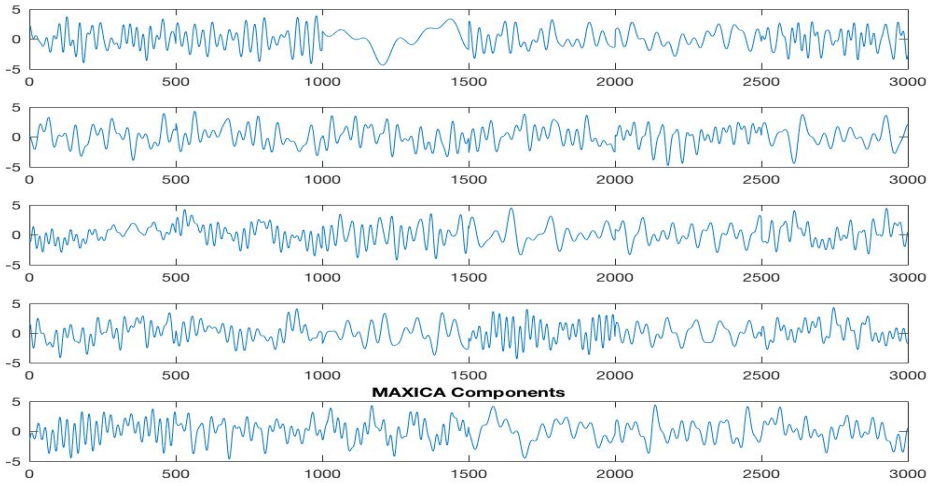


Figure 5.28: *MaxICA components of the movement group*

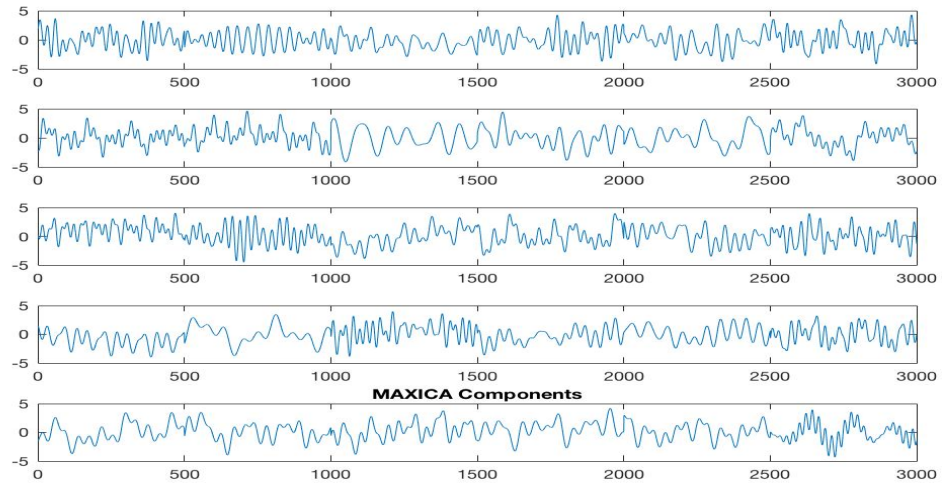


Figure 5.29: *MaxICA components of the imagery group*

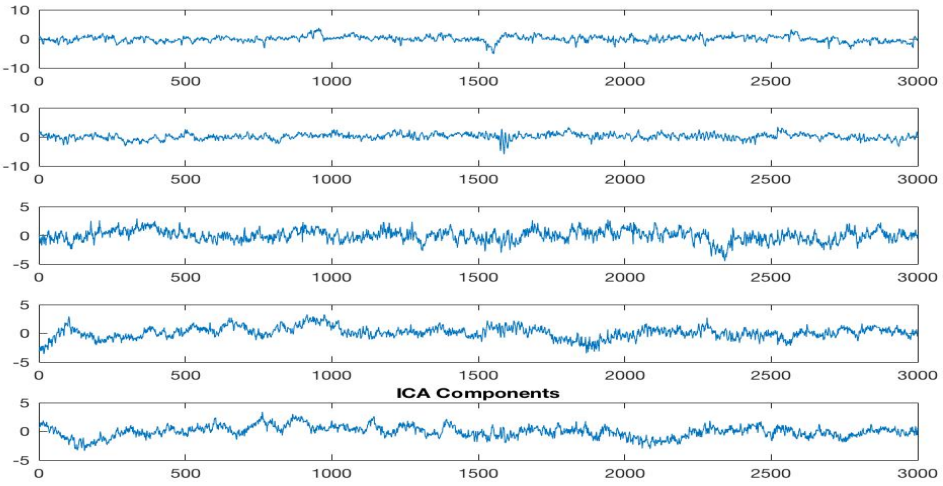


Figure 5.30: *ICA components of the control group*

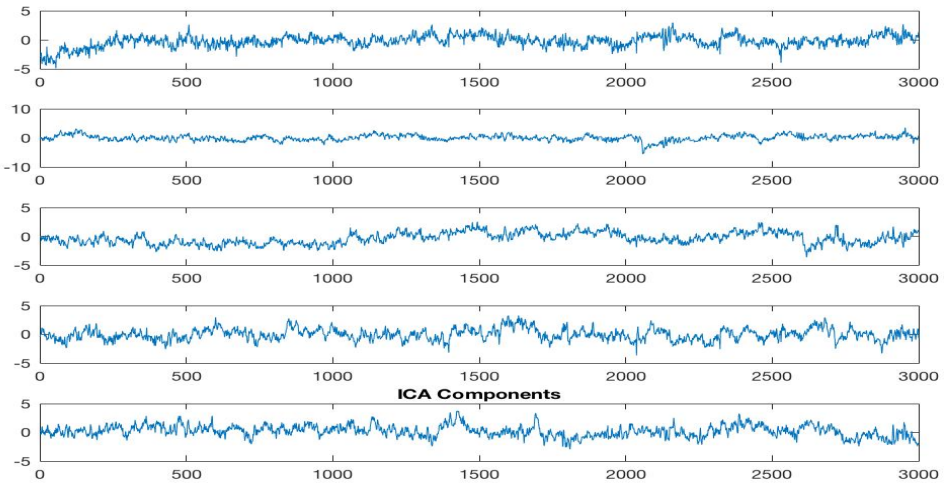


Figure 5.31: *ICA components of the movement group*

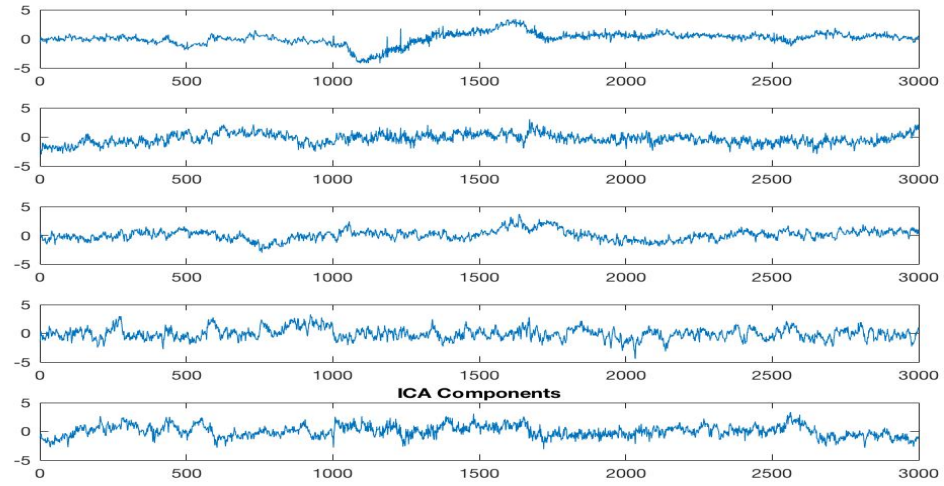


Figure 5.32: *ICA components of the imagery group*

Chapter 6

Conclusions

In this thesis, we focused on a new model for detecting blind sources, and proposed a genetic algorithm with new operators for this model.

The existing methods for finding blind sources are widely used but still have some limits respectively, for example, ICA cannot be well applied if the number of signals is less than the number of hidden components as mentioned in (Hyvärinen and Oja, 2000). Even with many new ICA methods developed recently found in (Chen and Bickel, 2006), (Guo and Tang, 2013), it's still hard to get rid of the restriction, related works can be found in (Sokol et al., 2014). The MaxICA model is more widely applicable. It's not restricted by the number of signals we observed.

Because of the characteristics of MaxICA model, many optimization methods couldn't be applied or cannot give good result. We use an augmented genetic algorithm. The method performs better than the regular GA.

The MaxICA performs well in simulations as mentioned in Chapter 4 and real

data mentioned in Chapter 5. For the real data analysis, MaxICA method found the major hidden components, and comparing to ICA, MaxICA stated a model for hidden sources that ICA cannot provide.

In the future research, we will apply MaxICA for more applications, such as epilepsy seizure monitoring, image processing, etc.

Appendix A

Additional data

Here we list all fitted values of $s_1, s_2, ..., s_5$ and B from section (5.1) for t from 500ms to 6000ms. For convenience, define following notations:

$$s_1 = \sin(\alpha_{11}t + \beta_{11}) + \sin(\alpha_{12}t + \beta_{12}) + \sin(\alpha_{13}t + \beta_{13}) + \sin(\alpha_{14}t + \beta_{14}) + \sin(\alpha_{15}t + \beta_{15});$$

$$s_2 = \sin(\alpha_{21}t + \beta_{21}) + \sin(\alpha_{22}t + \beta_{22}) + \sin(\alpha_{23}t + \beta_{23}) + \sin(\alpha_{24}t + \beta_{24}) + \sin(\alpha_{25}t + \beta_{25});$$

$$s_3 = \sin(\alpha_{31}t + \beta_{31}) + \sin(\alpha_{32}t + \beta_{32}) + \sin(\alpha_{33}t + \beta_{33}) + \sin(\alpha_{34}t + \beta_{34}) + \sin(\alpha_{35}t + \beta_{35});$$

$$s_4 = \sin(\alpha_{41}t + \beta_{41}) + \sin(\alpha_{42}t + \beta_{42}) + \sin(\alpha_{43}t + \beta_{43}) + \sin(\alpha_{44}t + \beta_{44}) + \sin(\alpha_{45}t + \beta_{45});$$

$$s_5 = \sin(\alpha_{51}t + \beta_{51}) + \sin(\alpha_{52}t + \beta_{52}) + \sin(\alpha_{53}t + \beta_{53}) + \sin(\alpha_{54}t + \beta_{54}) + \sin(\alpha_{55}t + \beta_{55}).$$

$$B = \begin{pmatrix} b_{11} & \cdots & b_{15} \\ \vdots & \ddots & \vdots \\ b_{51} & \cdots & b_{55} \end{pmatrix}.$$

Let $\mathbf{p} = (\alpha_{11}, \beta_{11}, ..., \alpha_{15}, \beta_{15}, \alpha_{21}, \beta_{21}, ..., \alpha_{25}, \beta_{25}, ..., \alpha_{51}, \beta_{51}, ..., \alpha_{55}, \beta_{55},$

$b_{11}, ..., b_{15}, b_{21}, ..., b_{25}, ..., b_{51}, ..., b_{55})$.

For t from 500ms to 1000ms,

$\mathbf{p} = (0.021644, 175.44, 0.11196, 143.74, 0.075108, -51.828, 0.10666, 95.25, 0.13404, 78.551, 0.10817,$

$-151.54, 0.0058686, -87.797, 0.06662, -82.307, 0.16763, -50.47, 0.045462, -5.3814, 0.019327, -131.43,$

$0.031834, 17.975, 0.095658, -184.24, 0.10882, -125.99, 0.093896, -91.032, 0.19338, -64.005, 0.12804,$

$39.846, 0.18439, -183.91, 0.19095, -41.532, 0.061924, -76.379, 0.10583, -50.995, 0.099848, -135.96,$

$0.060105, -48.396, 0.0057568, 28.947, 0.042124, -91.223, -0.45764, 9.6994, 7.9232, 17.905, -3.8968,$

$-5.8816, 7.4563, 8.1642, -13.579, -5.6182, -10.526, -3.834, 14.848, 10.885, 5.739, -12.364,$

$9.8011, 10.458, -15.586, -8.7427, -8.2629, -29.341, 7.4393, 10.49, 4.06)$.

For t from 1000ms to 1500ms,

$\mathbf{p} = (0.15926, 165.4, 0.10479, -156.9, 0.0758, -120.41, 0.070432, 90.641, 0.059517, -169.75, 0.035668,$

$-2.6943, 0.020875, 59.746, 0.00031336, 58.597, 0.027154, -104.64, 0.011934, 89.812, 0.067901, -66.618,$

$0.15384, 77.648, 0.14893, 171.68, 0.1256, 134.42, 0.018314, 90.091, 0.0057752, 75.714, 0.015288,$

$-116.6, 0.16538, -146.65, 0.10028, -117.48, 0.19309, -128.05, 0.14817, -165.59, 0.080568, 28.054,$

$0.0065556, 78.376, 0.14189, -13.539, 0.18691, 57.324, -21.014, 46.891, 8.3897, -30.809, 27.255,$

$1.8564, 42.797, -20.22, -20.018, 16.929, 1.4062, 22.131, 11.185, -6.1593, 21.25, 20.263,$

$25.692, 13.734, 3.2535, 24.223, -15.795, 15.034, 9.8305, -3.3366, 10.605)$.

For t from 1500ms to 2000ms,

$\mathbf{p} = (0.077951, 119.79, 0.13586, 109.81, 0.08923, 130.94, 0.080992, -48.456, 0.049041, 42.43, 0.16808,$

$34.845, 0.16585, 151.14, 0.093473, -53.934, 0.026176, -49.726, 0.022457, -139.37, 0.025859, 136.25,$
$0.1556, -126.21, 0.02258, -10.273, 0.10468, 52.865, 0.074288, -58.54, 0.020225, -86.522, 0.010021,$
$-86.898, 0.0003295, -62.037, 0.0030185, 144.99, 0.0083184, 134.01, 0.054992, 156.92, 0.024615, -100.75,$
$0.019668, 124.71, 0.042809, 59.305, 0.027603, 176.51, -42.104, 15.511, 40.353, 44.104, 1.6549,$
$13.505, -20.121, 19.409, 44.283, -15.172, -4.5369, -4.319, -23.246, 15.574, 28.42, -14.639,$
$- 37.688, -24.726, 13.038, 16.119, -13.86, -8.572, 8.6526, -39.008, 15.011).$

For t from 2000ms to 2500ms,

$\mathbf{p} = (0.18577, 52.507, 0.14273, -128.02, 0.19661, -189.16, 0.1831, 2.089, 0.12804, 71.533, 0.12208,$
$102.36, 0.039655, 147.15, 0.15768, 63.29, 0.028699, -5.9883, 0.059338, -196.6, 0.108, 109.21,$
$0.0096159, 48.781, 0.13782, 85.913, 0.024644, 189.16, 0.17377, -51.575, 0.0039399, 185.91, 0.039634,$
$-144.8, 0.091139, -100.65, 0.16929, -100.54, 0.17691, 94.453, 0.18319, 173.89, 0.17128, 157.78,$
$0.014735, 173.99, 0.10809, -187.2, 0.11288, 98.621, -1.1377, 12.282, -31.944, 34.559, -27.624,$
$-6.4072, -14.572, -19.603, -6.9487, -6.5381, -15.732, 14.974, -9.0342, 16.501, -16.642, -7.0828,$
$- 26.093, -0.052027, -22.085, -9.3258, -8.3793, -5.6672, -12.281, -12.215, 11.712).$

For t from 2500ms to 3000ms,

$\mathbf{p} = (3.8062e{-}05, -101.86, 0.071251, 144.08, 0.034138, 179.67, 0.010615, -155.21, 0.10641, -129.47,$
$0.17727, -103.24, 0.08297, 14.072, 0.078598, -13.817, 0.053948, 187.67, 0.081343, -143.42, 0.18789,$
$-159.31, 0.044573, -172.21, 0.053745, 157.77, 0.046708, 18.643, 0.076449, -196.09, 0.15716, 141.9,$
$0.16433, 149.12, 0.14109, 102.26, 0.14318, 55.788, 0.1198, 158.6, 0.049082, -163.24, 0.012127, 15.891,$
$0.13623, -193.08, 0.024672, -144.5, 0.059229, -37.909, -15.842, -26.976, -11.896, 15.418, 27.958,$
$-14.873, 2.139, 7.6714, 20.808, 14.041, -24.501, -27.23, -2.5208, -27.807, -24.167, -22.924,$
$21.098, 32.391, 0.47441, -24.016, -1.1753, -10.164, -6.2839, 5.619, 3.6339).$

For t from 3000ms to 3500ms,

$\mathbf{p} = (0.043343, 172.52, 0.053561, 128.89, 0.17098, 164.3, 0.18716, -174.92, 0.04335, -178.54, 0.060447, -169.26, 0.11148, 143.48, 0.11114, -22.836, 0.13251, 146.45, 0.10285, 98.179, 0.087272, 33.766, 0.090431, 42.139, 0.083251, -70.582, 0.014865, 77.803, 0.11639, 1.9719, 0.088907, 171.64, 0.091835, -145.91, 0.090983, -137.02, 0.17021, 113.54, 0.19222, 99.311, 0.11842, -65.057, 0.12614, 105.79, 0.16947, 57.006, 0.10881, -152.71, 0.12589, -55.94, -9.2022, 5.5654, 22.212, 12.472, -17.611, 11.987, 22.847, -7.2433, 12.134, -24.927, 21.12, -17.377, 4.1589, -17.917, -19.752, 17.035, 11.693, -16.131, -13.587, 16.822, -8.8057, 16.643, 16.721, 13.543, -21.054)$.

For t from 3500ms to 4000ms,

$\mathbf{p} = (0.0020066, 176.14, 0.19623, 169.31, 0.041177, 74.347, 0.058135, -47.767, 0.16336, 109.72, 0.092678, 198.87, 0.13219, 38.877, 0.025854, 67.22, 0.011009, 169.61, 0.11881, 144.06, 0.17796, -55.512, 0.021331, -182.56, 0.071081, -35.025, 0.024556, -36.615, 0.026497, -172.09, 0.053939, 2.3442, 0.043936, -57.33, 0.13068, 3.6961, 0.018749, 91.257, 0.056914, -51.893, 0.0054332, -40.01, 0.0049302, -112.95, 0.068333, -107.13, 0.06373, -196.22, 0.070496, 121.93, 14.423, 9.4492, -13.591, -13.129, 15.258, -1.0307, -4.4353, 2.9352, -5.6562, -14.162, -1.2219, -17.301, -17.833, -10.016, 21.266, 5.9807, 3.2895, -15.204, -18.781, 23.735, -16.183, 10.863, -19.86, -0.75056, -25.221)$.

For t from 4000ms to 4500ms,

$\mathbf{p} = (0.18241, 46.848, 0.12483, -1.7814, 0.10157, 26.578, 0.16493, -178.62, 0.039772, 0.71009, 0.19427, -81.411, 0.14849, -66.097, 0.094939, -86.696, 0.14908, -177.12, 0.14814, 138.96, 0.0058804, -38.51, 0.14557, -112.27, 0.17052, -109.73, 0.18327, -29.031, 0.15226, 32.921, 0.10836, -152.28, 0.0030348, 197.54, 0.10674, -41.871, 0.15956, 198.64, 0.063394, 59.544, 0.128, 156.58, 0.10256, 195.62, 0.048385, -131.1, 0.051422, -178.21, 0.049952, -24.17, -8.3522, 5.5933, -2.5045, 25.69, -4.9878, -0.57536, -21.158, -14.341, 3.1662, 5.8009, -15.321, -18.02, 14.934, 30.415, -11.73, -4.4716, -15.347, 10.442, -17.259, 9.4541, 3.2928, -26.927, 10.789, 12.219, 13.341)$.

For t from 4500ms to 5000ms,

$\mathbf{p} = (0.029348, 26.755, 0.16379, -133.03, 0.089708, -74.636, 0.00108, -92.017, 0.032099, 10.937,$
$0.0019973, -194.23, 0.027588, 57.894, 0.013039, 111.92, 0.0076651, -144.98, 0.034781, 89.861, 0.037328,$
$34.324, 0.072782, 5.4532, 0.12134, -77.824, 0.16514, 9.9827, 0.10424, 190.22, 0.16636, -98.375, 0.040669,$
$194.81, 0.0049371, -182.91, 0.0061542, 58.588, 0.057825, 189.64, 0.067832, -13.813, 0.044857, 104.01,$
$0.19116, 46.757, 0.19716, 142.02, 0.031112, 7.4286, -2.1142, 42.706, 1.7582, -35.813, -7.9133,$
$-12.09, 38.82, 12.741, 7.0867, 15.048, -16.681, 12.431, -13.915, 10.703, -14.021, 12.53,$
$13.644, -10.574, 10.339, 14.51, -7.1125, -31.692, 2.3382, 5.1788, -13.882).$

For t from 5000ms to 5500ms,

$\mathbf{p} = (0.14772, 193.36, 0.17632, -137.2, 0.056454, -19.306, 0.11337, -113.31, 0.19462, 44.784, 0.043585,$
$-81.792, 0.12735, -103.86, 0.051878, 162.32, 0.043883, -41.38, 0.10804, 102.7, 0.17716, -156.11,$
$0.12629, 11.637, 0.069375, 51.804, 0.13773, -150.39, 0.066684, -49.377, 0.12203, -123.13, 0.18012,$
$133.62, 0.062057, 81.636, 0.02296, 96.034, 0.056877, 176.55, 0.15211, 122.51, 0.028855, -47.191,$
$0.071682, 153.02, 0.063847, 31.854, 0.037666, 87.22, -13.126, -21.259, -20.823, -18.309, 19.245,$
$13.247, -27.209, -18.847, -21.698, 14.107, -6.4349, -11.078, -10.286, -20.837, 1.0563, -17.248,$
$- 13.439, 1.0184, 10.954, -8.0053, 0.99192, 11.869, -2.1385, -7.2434, 9.09).$

For t from 5500ms to 6000ms,

$\mathbf{p} = (0.18259, 171.9, 0.19496, 30.864, 0.088141, -175.26, 0.029034, 11.267, 0.13063, -98.069, 0.17602,$
$-36.352, 0.052259, 116.32, 0.048102, 89.569, 0.13063, -71.886, 0.011553, -62.972, 0.13507, 35.642,$
$0.056893, -8.1687, 0.053385, -175.19, 0.096192, -9.5673, 0.12051, 164.57, 0.13354, -112.97, 0.036302,$
$9.6226, 0.1975, 40.417, 0.077752, -196.1, 0.032574, -175.81, 0.088689, -141.28, 0.00060875, -50.618,$
$0.054271, 177.52, 0.13197, -177.23, 0.16148, -189.92, 7.1028, 3.0771, -7.9977, -10.081, 9.4907,$
$-3.8529, 6.9552, -12.325, -7.2994, 3.268, -8.2928, 6.3904, 5.6168, 7.1783, -13.48, 6.1351,$

$2.638, -6.8702, 1.5315, -9.6404, 7.1369, -4.0263, -0.26064, -7.6136, -2.6399)$.

Bibliography

Amirjanov, A. (2015). The parameters setting of a changing range genetic algorithm. *Natural Computing 14*(2), 331–338.

Andrzejak, R. G., K. Schindler, and C. Rummel (2012). Nonrandomness, nonlinear dependence, and nonstationarity of electroencephalographic recordings from epilepsy patients. *Physical Review E 86*(4), 046206.

Artoni, F., A. Delorme, and S. Makeig (2018). Applying dimension reduction to eeg data by principal component analysis reduces the quality of its subsequent independent component decomposition. *NeuroImage*.

Bhandari, D., C. Murthy, and S. K. Pal (1996). Genetic algorithm with elitist model and its convergence. *International journal of pattern recognition and artificial intelligence 10*(06), 731–747.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics 31*, 307–327.

Bronkhorst, A. W. (2000). The cocktail party phenomenon: A review of research

on speech intelligibility in multiple-talker conditions. *Acta Acustica united with Acustica 86*(1), 117–128.

Chen, A. and P. J. Bickel (2006). Efficient independent component analysis. *The Annals of Statistics 34*(6), 2825–2855.

Cramer, H. and M. Leadbetter (1967). *Stationary and Related Stochastic Procesees: Sample Function Properties and Their Applications*. Wiley.

Cryer, J. D. and K.-S. Chan (2008). Time series regression models. *Time series analysis: with applications in R*, 249–276.

Cui, Q. and Z. Zhang (2018). Max-linear competing factor models. *Journal of Business and Economic Statistics 36*, 62–74.

Delorme, A., S. Makeig, M. Fabre-Thorpe, and T. Sejnowski (2002). From single-trial eeg to brain area dynamics. *Neurocomputing 44*, 1057–1064.

Delorme, A., G. A. Rousselet, M. J.-M. Macé, and M. Fabre-Thorpe (2004). Interaction of top-down and bottom-up processing in the fast visual analysis of natural scenes. *Cognitive Brain Research 19*(2), 103–113.

Driss, I., K. N. Mouss, and A. Laggoun (2015). A new genetic algorithm for flexible job-shop scheduling problems. *Journal of Mechanical Science and Technology 29*(3), 1273–1281.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of uk inflation. *Econometrica 50*, 987–1007.

Fabre-Thorpe, M., A. Delorme, C. Marlot, and S. Thorpe (2001). A limit to the speed of processing in ultra-rapid visual categorization of novel natural scenes. *Journal of cognitive neuroscience 13*(2), 171–180.

Goldberg, D. E. and K. Deb (1991). A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of genetic algorithms*, Volume 1, pp. 69–93. Elsevier.

Goldberger, A. L., L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley (2000). Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation 101*(23), e215–e220.

Guo, Y. and L. Tang (2013). A hierarchical model for probabilistic independent component analysis of multi-subject fmri studies. *Biometrics 69*(4), 970–981.

Hassan, I. (2015). Combined crossover operator. *Research Journal of Applied Sciences 10*(3), 75–79.

Haupt, R. L., S. E. Haupt, and S. E. Haupt (1998). *Practical genetic algorithms*, Volume 2. Wiley New York.

Heffernan, J. E., J. A. Tawn, and Z. Zhang (2007). Asymptotically (in)dependent multivariate maxima of moving maxima processes. *Extremes 10*, 57–82.

Huang, H., J. Lu, J. Wu, Z. Ding, S. Chen, L. Duan, J. Cui, F. Chen, D. Kang, L. Qi, W. Qiu, S.-W. Lee, S. Qiu, D. Shen, Y.-F. Zang, and H. Zhang (2018).

Tumor tissue detection using blood-oxygen-level-dependent functional mri based on independent component analysis. *Scientific Reports 8*.

Hyvärinen, A., J. Karhunen, and E. Oja (2001). Independent component analysis.

Hyvärinen, A. and E. Oja (2000). Independent component analysis: algorithms and applications. *Neural networks 13*(4-5), 411–430.

Idowu, T. and Z. Zhang (2017). An extended sparse max-linear moving model with application to high-frequency financial data. *Statistical Theorey and Related Fields 1*, 92–111.

Jebari, K. and M. Madiafi (2013). Selection methods for genetic algorithms. *International Journal of Emerging Sciences 3*(4), 333–344.

Kassouf, A., D. Jouan-Rimbaud Bouveresse, and D. N. Rutledge (2018). Determination of the optimal number of components in independent components analysis. *Talanta 179*, 538–545.

Nascimento, M., F. Silva, T. Sáfadi, A. C. C. Nascimento, T. E. M. Ferreira, L. M. A. Barroso, C. F. Azevedo, S. E. Guimarães, and N. V. Serão (2017). Independent component analysis (ica) based-clustering of temporal rna-seq data. *PLoS One 12*(7), e0181195.

Naveau, P., Z. Zhang, and B. Zhu (2011). An extension of max autoregressive models. *Journal of Econometrics 194*, 231–241.

Obitko, M., P. Slavik, and P. Walter (1998). Introduction to Genetic Algorithms. `http://www.obitko.com/tutorials/genetic-algorithms/index.php`. [Online; accessed 01-May-2018].

Paes, F. G., A. A. Pessoa, and T. Vidal (2017). A hybrid genetic algorithm with decomposition phases for the unequal area facility layout problem. *European Journal of Operational Research 256*(3), 742–756.

Schalk, G., D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw (2004). Bci2000: a general-purpose brain-computer interface (bci) system. *IEEE Transactions on biomedical engineering 51*(6), 1034–1043.

Shoeb, A. H. (2009). *Application of machine learning to epileptic seizure onset detection and treatment*. Ph. D. thesis, Massachusetts Institute of Technology.

Sokol, A., M. H. Maathuis, and B. Falkeborg (2014). Quantifying identifiability in independent component analysis. *Electronic Journal of Statistics 8*(1), 1438–1459.

Thierens, D. (2002). Adaptive mutation rate control schemes in genetic algorithms. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, Volume 1, pp. 980–985. IEEE.

Umbarkar, A. and P. Sheth (2015). Crossover operators in genetic algorithms: A review. *ICTACT journal on soft computing 6*(1), .

Zhang, C., Y. Chai, X. Guo, M. Gao, D. Devilbiss, and Z. Zhang (2016). Statistical learning of neuronal functional connectivity. *Technometrics 58*, 350–359.

Zhang, Y., Z. Dong, P. Phillips, S. Wang, G. Ji, J. Yang, and T.-F. Yuan (2015). Detection of subjects and brain regions related to alzheimer's disease using 3d mri scans based on eigenbrain and machine learning. *Front Comput Neurosci*.

Zhang, Z. (2008). The estimation of m4 processes with geometric moving patterns. *Annals of Institute of Statistical Mathematics 60*, 121–150.

Zhang, Z. and R. L. Smith (2004). The behavior of multivariate maxima of moving maxima processes. *Journal of Applied Probability 41*, 1113–1123.

Zhang, Z. and B. Zhu (2016). Copula structured m4 processes with application to high-frequency financial data. *Journal of Econometrics 194*, 231–241.