Statistical Science
2020, Vol. 35, No. 1, 145–157
https://doi.org/10.1214/19-STS763

Maximum Independent Component Analysis with Application to EEG Data

Ruosi Guo, Chunming Zhang and Zhengjun Zhang

Abstract. In many scientific disciplines, finding hidden influential factors behind observational data is essential but challenging. The majority of existing approaches, such as the independent component analysis (ICA), rely on linear transformation, that is, true signals are linear combinations of hidden components. Motivated from analyzing nonlinear temporal signals in neuroscience, genetics, and finance, this paper proposes the "maximum independent component analysis" (MaxICA), based on max-linear combinations of components. In contrast to existing methods, MaxICA benefits from focusing on significant major components while filtering out ignorable components. A major tool for parameter learning of MaxICA is an augmented genetic algorithm, consisting of three schemes for the elite weighted sum selection, randomly combined crossover, and dynamic mutation. Extensive empirical evaluations demonstrate the effectiveness of MaxICA in either extracting max-linearly combined essential sources in many applications or supplying a better approximation for nonlinearly combined source signals, such as EEG recordings analyzed in this paper.

Key words and phrases: Blind source separation, genetic algorithm, image analysis, nonlinear time series, optimization.

1. INTRODUCTION

In many scientific fields, such as neuroscience, genetics, and finance, observational data in the form of nonlinear temporal signals are commonly recorded and require advanced analysis approaches. A commonly encountered but highly under-determined problem is known as "blind source separation," which aims to separate a set of source signals from a set of mixed signals, with very little information about either the source signals or the mixing mechanism. Exploring hidden sources or components from the collected data is essential for analyzing and forecasting, and thus of great interest, but also poses significant challenges.

For instance, in neuroscience and statistics, much effort has been put into modeling the recorded time series of brain activity signals and learning brain activity patterns associated with certain cognitive tasks and brain disorders. Yet, many problems and issues still remain unsolved. This paper explores a newly proposed multivariate time series model with applications to learn hidden patterns and dynamics of brain activities in analyzing datasets in Section 5 on visualization and epilepsy.

The advancement of independent component analysis (ICA) [4, 15, 19, 20, 28] has made it possible to discover statistically independent and non-Gaussian components, based on their linear representations. There exists a rich literature on ICA applied to blind source separation for complex data sets in many research areas. For a detailed review, refer to the most recent work in analytical chemistry [22], gene expression time series [23], brain image analysis [2, 18], among others and references therein. Besides ICA, there are other methods for blind source separation, that is, principal components analysis, singular value decomposition, among others.

A basic assumption of ICA is that the component sources are linearly combined, as in the well-known motivating example of the cocktail-party effect. The effect says that if someone goes to a cocktail party and has conversations with other people, his brain will automatically filter out all other noises in the background, so that he can focus on individual(s) he's talking with. See [3] for some related studies.

Ruosi Guo is a Ph.D. graduate, Department of Statistics, University of Wisconsin, Madison, Wisconsin 53706, USA (e-mail: rguo25@wisc.edu). Chunming Zhang is Professor, Department of Statistics, University of Wisconsin, Madison, Wisconsin 53706, USA (e-mail: cmzhang@stat.wisc.edu). Zhengjun Zhang is Professor, Department of Statistics, University of Wisconsin, Madison, Wisconsin 53706, USA (e-mail: zjz@stat.wisc.edu).

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 we can use ICA to recover the two original speech signals. Such an approach, via a simple linear combination of the original source signals, can capture the essential structure of the data in many applications, but is not sufficient for fully solving the problem to be described next. The estimated source signals separated from the recorded signals by using existing methods, such as ICA, contain all information, as the method was designed to separate the data as precisely as possible, as explained in [20]. In some situations, this is useful, precise and fast as well, but in some other situations we might neither need nor access that much information. Let us continue with the cocktail party example under a different scenario: there were two groups of people talking to each other within each group. Suppose that 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 of their interest, they would be attracted to a 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 us further consider an experiment: Suppose that 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 linear combinations of the original source signals, that is, 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.

1.1 Motivating Example for MaxICA

This paper intends to develop a new time series model for observed signals obtained by taking the maximum of the source signals. These kinds of output signals are more like the recorded signals in some practical situations, where the major information is focused on and minor information is filtered out. In nature, this phenomenon is closer to the way the human brain functions. The following numerical example illustrates our idea. Suppose that two component signals $S_1(t)$ and $S_2(t)$, plotted in Figure 1, are defined by

$$S_1(t) = \sin(1.3t + 2.5), \qquad S_2(t) = \sin(0.5t + 2), \tag{1.1}$$

where t denotes a generic time point in the interval [0, T]. In Figure 2, the top panel plots the two source signals $1.2S_1(t_i)$ and $2.0S_2(t_i)$, the middle panel depicts the sum of these two source signals, that is, $1.2S_1(t_i) + 2.0S_2(t_i)$, whereas the bottom panel draws the maxima of these two source signals, that is,

$$\max\{1.2S_1(t_i),\ 2.0S_2(t_i)\}, \tag{1.2}$$

at time points $\{t_i : i = 1, \dots, 1000\}$ equally spaced in [0, T] with T = 50. The linear combination contains all information from the two component signals, that is, all details will be reflected in the combined signals; even trivial perturbation will affect the combined signal. For the max-linear combination, the resulting curve consists of fragments exactly coming from one of the two component signals; some smaller waves will be covered by larger waves. As a result, taking the maximum operation can reduce the effects generated from small turbulence, while it mainly focuses on the major information.

The maximum operation applied to source signals inspires us to call the procedure "maximum independent component analysis" (MaxICA). Analogous to ICA, MaxICA aims to recover the component signals from the output signals in the bottom panel of Figure 2. Let us further consider two sequences of true signals,

$$\begin{aligned} \text{signal}_1(t_i) &= \max\{1.2S_1(t_i),\ 2.0S_2(t_i)\},\\ \text{signal}_2(t_i) &= \max\{2.5S_1(t_i),\ 0.5S_2(t_i)\}, \end{aligned} \tag{1.3}$$

where $\text{signal}_1(t_i)$ is also used in (1.2). The left panels of Figure 3 display $\text{signal}_1(t_i)$ and $\text{signal}_2(t_i)$. After adding random noises $\epsilon_1(t_i) \overset{\text{i.i.d.}}{\sim} 0.2\mathbb{N}(0, 1)$ and $\epsilon_2(t_i) \overset{\text{i.i.d.}}{\sim} 0.2\mathbb{N}(0, 1)$, the resulting observed signals $X_j(t_i) = \text{signal}_j(t_i) + \epsilon_j(t_i)$, $j = 1, 2$, are plotted in the right panels of Figure 3. A natural issue is to recover the true components $\{S_1(t_i), S_2(t_i)\}$ in (1.1), from the observed signals $X_1(t_i)$ and $X_2(t_i)$ in Figure 3. As seen from Figure 4, the components recovered via the MaxICA method resemble the true components $\{S_1(t_i), S_2(t_i)\}$ (in red dashed lines), whereas the ICA method fails to recover the true components. See also Remark 1 for the order of recovered components.

REMARK 1. Both ICA and MaxICA methods focus on separating the set $\{S_1(t), \dots, S_N(t)\}$ of "hidden components" from "observed signals" $X_1(t), \dots, X_p(t)$, but in general, could not uniquely identify the original ordering of hidden components $\{S_1(t), \dots, S_N(t)\}$.

It is readily noticed that we are dealing with inferences for multivariate nonlinear time series. In the time series literature, there has been much development in modeling nonlinear time series. Refer to [11] for an extensive review of theory and methods, and [31] and references therein for applications to neuronal functional connectivity. On the other hand, the max-autoregressive and moving-maxima models, in [24] and the recent literature, can be applicable to modeling time series of maxima within a single brain voxel or among multiple voxels.

Different from the existing work, a new approach called “the maximum independent component analysis (MaxICA)” for studying time series data is introduced in this paper. Our new framework can be thought as an analogue to the spectral analysis in the classical time series. In the spectral analysis, observed signals are decomposed into linear combinations of sine waves or functions. In our new framework, observed signals are decomposed into max-linear combinations of sine waves or functions. A major tool for parameter learning of MaxICA is an “augmented genetic algorithm,” called ERD_GA, consisting of schemes for the **e**lite weighted sum selection, **r**andom combined crossover and **d**ynamic mutation, respectively. Extensive empirical evaluations demonstrate the effectiveness of the MaxICA method in either extracting max-linearly combined essential sources in many applications or supplying a better approximation for nonlinearly combined source signals, such as EEG recordings analyzed in this paper.

The rest of the paper is structured as follows. Section 2 proposes the MaxICA model and the estimation method. Section 3 develops the ERD_GA algorithm which augments the classical genetic algorithm for estimating parameters in our MaxICA model. Section 4 presents simulation examples to assess the performance of MaxICA. Section 5 analyzes two sets of EEG data, that is, visual processing data and epilepsy data. Section 6 concludes. The online supplementary file [14] collects notations and all figures in the paper.

2. PROPOSED MaxICA MODEL AND ITS ESTIMATION

Before introducing our proposed model for maximum independent component analysis (MaxICA), we first briefly review two relevant time series models widely used in the literature.

2.1 Review of Spectral Analysis

The spectral analysis is widely used for time series data in geophysics, oceanography, atmospheric science, astronomy, and engineering, among others. It is usually used in cases where several cyclical patterns are simultaneously presented in a time series. Different from analyzing correlation properties of time series in the time domain, the spectral analysis is analyzing frequency properties of time series, and said to be working in the frequency domain. Following [6], we consider a time series represented by

$$Y_t = \sum_{j=1}^{\mathrm{m}} \{A_j \cos(2\pi \mathrm{f}_j t) + B_j \sin(2\pi \mathrm{f}_j t)\}, \tag{2.1}$$

where frequencies $0 < \mathrm{f}_1 < \cdots < \mathrm{f}_\mathrm{m} < 1/2$ are nonrandom, whereas amplitudes A_j and B_j are independent Gaussian random variables with zero mean and $\mathrm{var}(A_j) = \mathrm{var}(B_j) = \sigma_j^2$. Then the sequence $\{Y_t\}$ is stationary with zero mean and the lag-k auto-covariance equal to $\gamma_k = \sum_{j=1}^{\mathrm{m}} \sigma_j^2 \cos(2\pi k \mathrm{f}_j)$. Refer to [5] for other properties of stationary processes.

2.2 Review of Model for ICA

The independent component analysis (ICA) is a useful statistical and computational technique for finding hidden factors that underlie sequences of observed measurements or signals. The ICA model assumes the observed multivariate data to be linear mixtures of some unknown variables, with unknown mixing coefficients, and requires the hidden variables to be jointly non-Gaussian and mutually independent.

Consider a data matrix $\mathbf{X} = (X_j(t_i))_{j=1,\dots,p;i=1,\dots,m} \in \mathbb{R}^{p\times m}$ whose rows are vectors $\vec{\boldsymbol{X}}_1^T, \dots, \vec{\boldsymbol{X}}_p^T$ corresponding to p sequences of observed signals, where the superscript T denotes transpose, and

$$\vec{\boldsymbol{X}}_j = \big(X_j(t_1), \dots, X_j(t_m)\big)^T \tag{2.2}$$

denote the jth sequence of observed signals, $j = 1, \dots, p$, at time points $t_1, \dots, t_m$. Let $\mathbf{S} \in \mathbb{R}^{N\times m}$ be the component matrix whose rows are vectors $\vec{\boldsymbol{S}}_1^T, \dots, \vec{\boldsymbol{S}}_N^T$ associated with N component signals. Let $\mathrm{A} = (a_{j,k})_{j=1,\dots,p;k=1,\dots,N} \in \mathbb{R}^{p\times N}$ be a mixing matrix of coefficients, with columns $\boldsymbol{a}_1, \dots, \boldsymbol{a}_N$. The ICA model is written as

$$\begin{aligned} X_j(t) &= \sum_{k=1}^{N} a_{j,k} S_k(t), \\ j &= 1, \dots, p, \text{ that is, } \mathbf{X} = \mathrm{A}\mathbf{S} = \sum_{k=1}^{N} \boldsymbol{a}_k \vec{\boldsymbol{S}}_k^T, \end{aligned} \tag{2.3}$$

where t denotes a generic time point, and both A and $\mathbf{S}$ are unknown. If $p = N$, then independent components can be obtained by

$$\mathbf{S} = \mathrm{W}\mathbf{X},$$

where the matrix W is the inverse of the matrix A. Refer to references listed in Section 1 for recent developments and applications of ICA. The FastICA algorithm for estimating the mixing matrix A and recovering the component matrix $\mathbf{S}$ was first given by [20]. Throughout this paper, the numerical implementation of the ICA model adopts the FastICA package for Matlab.

In Section 2.3, we will develop a new modeling framework for some ideas illustrated in Section 1.

2.3 Proposed Model for MaxICA

Consider a sequence of harmonic frequencies, $\{\omega_\ell = \ell/n : \ell = 1, \dots, n/2\}$, with an even integer n. It follows from (2.1) that a component signal at a time point t may be represented as

$$S(t)=\sum_{\ell=1}^{n/2}\left\{\mathrm{a}_1\left(\frac{\ell}{n}\right)\cos(2\pi\omega_\ell t)+\mathrm{a}_2\left(\frac{\ell}{n}\right)\sin(2\pi\omega_\ell t)\right\},$$

where $\mathrm{a}_1(\ell/n)$ and $\mathrm{a}_2(\ell/n)$ are treated as regression coefficients. The identity $\cos(2\pi\omega_\ell t)=\sin(2\pi\omega_\ell t+\pi/2)$ gives

$$S(t)=\sum_{\ell=1}^{n/2}\left\{\mathrm{a}_1\left(\frac{\ell}{n}\right)\sin(2\pi\omega_\ell t+\pi/2)+\mathrm{a}_2\left(\frac{\ell}{n}\right)\sin(2\pi\omega_\ell t)\right\}. \tag{2.4}$$

For N component signals $S_1(t),\ldots,S_N(t)$, they are expressed via (2.4) as

$$S_k(t)=\sum_{\ell=1}^{n_k}\mathrm{a}_{k,\ell}\sin(\alpha_{k,\ell}t+\beta_{k,\ell}),\quad k=1,\ldots,N, \tag{2.5}$$

where n_k is the number of sine waves within the kth component, $\{\mathrm{a}_{k,\ell}\}$, $\{\alpha_{k,\ell}\}$ and $\{\beta_{k,\ell}\}$ are unknown parameters. For p sequences of observed signals, $X_1(t),\ldots,X_p(t)$, we propose the model

$$\begin{aligned} X_1(t) &= \max\{b_{1,1}S_1(t),\ldots,b_{1,N}S_N(t)\}+\epsilon_1(t),\\ \vdots\ &\ \ \vdots \qquad\qquad\qquad \vdots\\ X_p(t) &= \max\{b_{p,1}S_1(t),\ldots,b_{p,N}S_N(t)\}+\epsilon_p(t), \end{aligned} \tag{2.6}$$

for the maximum independent component analysis (MaxICA), where $S_k(t)$ is as defined in (2.5), and $\epsilon_1(t),\ldots,\epsilon_p(t)$ are random noise terms with zero means.

Finding the exact values of $\mathrm{a}_{k,\ell}$ in (2.5) and $b_{j,k}$ in (2.6) can be avoided, since $\mathrm{a}_{k,\ell}$ can be absorbed into $b_{j,k}$, by introducing $b_{j,k,\ell}=b_{j,k}\mathrm{a}_{k,\ell}$, $j=1,\ldots,p$, $k=1,\ldots,N$, and $\ell=1,\ldots,n_k$. Here $b_{j,k,\ell}$ stands for the coefficient in the ℓth sine wave within the kth component from the jth sequence of observed signals. We can thus redefine the source signals in (2.6) by

$$S_{j,k}(t)=\sum_{\ell=1}^{n_k}b_{j,k,\ell}\sin(\alpha_{k,\ell}t+\beta_{k,\ell}),\quad j=1,\ldots,p,k=1,\ldots,N, \tag{2.7}$$

and accordingly, the MaxICA model (2.6) becomes

$$\begin{aligned} X_1(t) &= \max\{S_{1,1}(t),\ldots,S_{1,N}(t)\}+\epsilon_1(t),\\ \vdots\ &\ \ \vdots \qquad\qquad\qquad \vdots\\ X_p(t) &= \max\{S_{p,1}(t),\ldots,S_{p,N}(t)\}+\epsilon_p(t), \end{aligned} \tag{2.8}$$

where $S_{j,k}(t)$ is defined in (2.7). Since the source signal $S_{j,k}(t)$ includes coefficients $b_{j,k,\ell}$, $\alpha_{k,\ell}$ and $\beta_{k,\ell}$, model (2.7) contains $(p+2)(n_1+\cdots+n_N)$ unknown parameters to be estimated.

For practical convenience, if we assume that $b_{j,k,\ell}$ in (2.7) are identical across all ℓ, that is, coefficients of all sine waves within the kth component from the jth sequence are all equal, then the source signals can be further simplified:

$$S_{j,k}(t)=b_{j,k}\sum_{\ell=1}^{n_k}\sin(\alpha_{k,\ell}t+\beta_{k,\ell}),\quad j=1,\ldots,p,k=1,\ldots,N. \tag{2.9}$$

Regarding model (2.9), the number of unknown parameters reduces to $Np+2(n_1+\cdots+n_N)$, a dramatic decrease from that of model (2.7).

REMARK 2. Analogous to the spectral analysis, $S_{j,k}(t)$ in (2.7) can be generalized to

$$S_{j,k}(t)=\sum_{\ell=1}^{n_k}B_{j,k,\ell}\sin(\alpha_{k,\ell}t+\beta_{k,\ell}),\quad j=1,\ldots,p,k=1,\ldots,N,$$

where $B_{j,k,\ell}$ are independent Gaussian random variables with zero means.

We now present a simulation example to demonstrate that the component signals in models (2.8) and (2.9) can be "recovered" by the proposed ERD_GA algorithm devised in Section 3.2 for our MaxICA method. Let $X_1(t)$ and $X_2(t)$ be observed signals generated from

$$\begin{pmatrix}X_1(t)\\X_2(t)\end{pmatrix}=\begin{pmatrix}\max\{b_{1,1}S_1(t),b_{1,2}S_2(t)\}\\\max\{b_{2,1}S_1(t),b_{2,2}S_2(t)\}\end{pmatrix}+\begin{pmatrix}\epsilon_1(t)\\\epsilon_2(t)\end{pmatrix}, \tag{2.10}$$

with two component signals,

$$\begin{aligned} S_1(t) &= 0.17724\sin(1.2t+2)+0.4609\sin(0.5t+1)\\ &\quad +0.80601\sin(0.34t+23)\\ &\quad -2.4463\sin(1.3t+2)-0.34539\sin(0.6t-3),\\ S_2(t) &= -0.57756\sin(0.5t+1.7)\\ &\quad -1.1414\sin(2t+2.1)-0.044345\sin(1.4t+2)\\ &\quad +0.16376\sin(0.25t+1)\\ &\quad +0.23883\sin(1.67t+2), \end{aligned}$$

mixing coefficients $b_{1,1}=0.21265$, $b_{1,2}=-1.2195$, $b_{2,1}=0.69503$ and $b_{2,2}=-0.55459$ are generated from the $\mathbb{N}(0,1)$ distribution, and noise terms $\{\epsilon_1(t),\epsilon_2(t)\}\overset{\text{i.i.d.}}{\sim}0.25\mathbb{N}(0,1)$. From Figure 5, we observe that the fitted signals (in blue lines), using MaxICA, follow the shape and trends of simulated data signals (in dots), lending support to the ERD_GA algorithm for the MaxICA model (2.9). To provide a unified discussion, this paper focuses on the

MaxICA model (2.8) with source signals $S_{j,k}(t)$ in (2.9), namely,

$$\text{MaxICA model (2.6) with component signals}$$
$$S_k(t) = \sum_{\ell=1}^{n_k} \sin(\alpha_{k,\ell} t + \beta_{k,\ell}). \tag{2.11}$$

Note that although both MaxICA and ICA aim to identify hidden components, they differ in terms of modeling aspects. The conventional ICA model (2.3) assumes nonstructured components and excludes random noises, oftentimes suffering from the "overfitting" problem as seen in Figures 18, 20, 22 and 26, whereas the MaxICA model (2.11) embeds structured components and incorporates noise terms with either known or unknown distributions, ameliorating such "overfitting" problem. It is to be remarked that both ICA and MaxICA have their own merits, and are not treated as interchangeable. Simulation studies in Section 4 will compare their performances under different scenarios.

2.4 Proposed Estimation for MaxICA

To formulate the loss function for parameter estimation in the MaxICA model (2.11), we define

$$\vec{\boldsymbol{f}}_j(\boldsymbol{\theta}) = \big(f_j(\boldsymbol{\theta}, t_1), \dots, f_j(\boldsymbol{\theta}, t_m)\big)^T, \quad j = 1, \dots, p,$$

with

$$f_j(\boldsymbol{\theta}, t) = \max\{b_{j,1} S_1(t), b_{j,2} S_2(t), \dots, b_{j,N} S_N(t)\} + \gamma,$$

where the location parameter γ is used for adjusting the horizontal location of the data, and $\boldsymbol{\theta} = \{\{(\alpha_{k,\ell}, \beta_{k,\ell})\}_{\ell=1,\dots,n_k; k=1,\dots,N}; \{b_{j,k}\}_{k=1,\dots,N; j=1,\dots,p}; \gamma\}$ collects all parameters. Together with the observed signals $\vec{\boldsymbol{X}}_j$ in (2.2), the loss function of $\boldsymbol{\theta}$ is formed by

$$L(\boldsymbol{\theta}) = \sum_{j=1}^{p} \|\vec{\boldsymbol{X}}_j - \vec{\boldsymbol{f}}_j(\boldsymbol{\theta})\|_2, \tag{2.12}$$

which is the overall differences in the L2-norm between observed signals and the modeled signals. Thus, we seek the optimization solution of $\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$ for parameter estimation.

3. PROPOSED ALGORITHM FOR MaxICA ESTIMATION

Due to the nonconvexity and nonsmoothness of the loss function (2.12), classical parameter estimation approaches, for example, maximum likelihood method, methods of moments, and other popular estimation methods in time series studies, are not directly applicable to solve the optimal $\boldsymbol{\theta}$ for the MaxICA model.

For the MaxICA estimation, we will devise an alternative optimization method for parameter learning, motivated from the genetic algorithm (GA). Section 3.1 reviews basics of the GA. Section 3.2 develops our ERD_GA algorithm, containing schemes for the elite weighted sum selection, random combined crossover and dynamic mutation, respectively. Section 3.3 summarizes the ERD_GA algorithm for the estimation in MaxICA.

3.1 A Brief Review of Genetic Algorithms

The genetic algorithm (GA) is a stochastic search method for optimization based on Darwin's principle of natural selection and genetics. Using GAs for optimization problems, the solutions are evolving with iterations. See [25] for some details. The algorithm starts with a set of randomly generated solutions, called "a population," in a pre-specified region for parameters. Once a population is generated, paired solutions are taken from the population to form a new population based on some selection rule, hoping that the newly generated population can be a better one. By checking fitness values of solutions, better solutions are more possible to be selected to generate new offsprings. Here, the "fitness" is a scalar computed from a loss function, indicating how close the given solution is to the preselected goal. A solution, which is closer to the goal (i.e., associated with a better fitness value), will be more possible to reproduce. This procedure is repeated to direct the population to the global optimum. Algorithm 1 outlines the procedure of the basic GA.

The procedure of the basic GA can be implemented for various generic problems. For example, [17] listed some sophisticated GAs, such as the hybrid GA and messy GA, [26] introduced a DRQ GA for solving the facility layout

Algorithm 1 Outline of the basic GA

```
1: [Start] Randomly generate a population of size $n$;
2: [Fitness] Compute the fitness of all chromosomes in the population;
3: [New population] Generate a new population by repeating the steps below:
     i. [Selection] Select two parent chromosomes based on fitness from the population with replacement;
     ii. [Crossover] Crossover the parents to generate a new offspring. If no crossover was made, the offspring is exactly the copy of parents;
     iii. [Mutation] Mutate the new offspring with a mutation probability;
     iv. [Accepting] Place the 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, then stop, and return the best solution;
6: [Loop] Go to step 2.
```

problem, and [9] presented a new GA for the job-shop scheduling problem, among others.

In the following Section 3.2, we develop an enhanced genetic algorithm, called ERD_GA, for estimating parameters in the MaxICA model.

3.2 Proposed Augmented Genetic Algorithm for MaxICA

3.2.1 *Chromosome representation for* MaxICA. Recall from Algorithm 1 that the chromosome in the GA is the set of unknown parameters. Also, recall from Section 2.4 that our goal is to seek the optimization solution of (2.12) for parameter estimates. Since the set of parameters in our MaxICA model contains $Np + 2\sum_{k=1}^{N} n_k + 1$ parameters $\{(\alpha_{k,\ell}, \beta_{k,\ell}); b_{j,k}; \gamma\}$, the chromosome in MaxICA contains $Np + 2\sum_{k=1}^{N} n_k + 1$ elements, the order of which is arranged as follows:

$$\begin{aligned}\text{chromosome}\\ &= \{(\alpha_{1,1}, \beta_{1,1}), \dots, (\alpha_{1,n_1}, \beta_{1,n_1}); \dots;\\ &\quad (\alpha_{N,1}, \beta_{N,1}), \dots, (\alpha_{N,n_N}, \beta_{N,n_N});\\ &\quad (b_{1,1}, \dots, b_{1,N}); \dots, (b_{p,1}, \dots, b_{p,N}); \gamma\}.\end{aligned}$$

Note that other ways of arranging the order of parameters are also feasible.

3.2.2 *Selection operator for* MaxICA. In the selection step of GAs, solutions are selected from the population to be parents to crossover. How to appropriately select solutions is an interesting issue. Among many others, the "Roulette Wheel Selection" method (RWS) is popularly used; see [21] for more details and mathematical structure. Using this method, parents are selected based on their fitness; solutions with better fitness would be more possible to be selected. 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. The selection procedure is similar to randomly throwing a marble on the wheel to pick up the solution, and solutions with better fitness values will be selected more easily. Algorithm 2 outlines this method. Another selection method is called the "Rank Selection" [13], which will rank the solutions by fitness, and thus each solution has a chance to be selected.

Algorithm 2 Outline of the Roulette Wheel Selection (RWS)

```
[Sum] Calculate the sum S of all chromosome fitness values in the population;
[Select] Generate a random number r from the interval (0, S);
[Loop] Go through the population and sum fitness values from 0. Define the sum to be s*. If s* > r, stop and return the chromosome where you are.
```

The disadvantages for these selection methods are obvious. For the RWS method, if fitness values of some solutions are excessively large, that is, some other solutions will have very small chances to be selected, then RWS will select those good solutions too often, and could hardly get any new solutions. The "Rank Selection," on the other hand, 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 or bad it is.

To ameliorate these disadvantages, we introduce a new selection method called the "Elite Weighted Sum Selection" (EWSS). Compute the loss function $L_s = L(\widehat{\boldsymbol{\theta}}^{(s)})$ in (2.12) from the solution $\widehat{\boldsymbol{\theta}}^{(s)}$, $s = 1, \dots, n$, where

$$L_1 \le L_2 \le \cdots \le L_n. \tag{3.1}$$

Here, a smaller loss L_s corresponds to a better solution indexed by s, and thus a better fitness value. Define by

$$d_j = L_{j+1} - L_j, \quad j = 1, \dots, n-1, \tag{3.2}$$

the differences between successive values of L_s, illustrated by $L_1 \xrightarrow{d_1} L_2 \xrightarrow{d_2} \cdots \xrightarrow{d_{n-2}} L_{n-1} \xrightarrow{d_{n-1}} L_n$. Define cumulative probabilities,

$$\begin{aligned} p_0 &= 0,\\ p_1 &= \frac{\sum_{j=1}^{n-1} d_j}{D},\\ p_2 &= \frac{\sum_{j=1}^{n-1} d_j + \sum_{j=2}^{n-1} d_j}{D},\\ &\dots,\\ p_{n-1} &= 1, \end{aligned} \tag{3.3}$$

where d_j are defined in (3.2), and $D = \sum_{j=1}^{n-1} d_j + \sum_{j=2}^{n-1} d_j + \cdots + d_{n-1}$. The EWSS method starts by generating a random real number r from (0, 1]. If $p_{s-1} < r \le p_s$, then the sth solution is selected. The solution with the worst fitness value is dropped from the population and will no longer be used to generate offsprings. Compared with RWS, EWSS guarantees that all solutions, excluding the worst one, can have some chance to be selected, even if fitness values of some solutions far exceed those of others. Compared with the "Rank Selection," chances for solutions selected by EWSS are not identical, that is, better solutions can have higher chances to be selected. Moreover, the most important aspect of EWSS is that solutions having similar fitness values will have similar chances to be selected.

As an illustration, let us consider an example of (ordered) losses L_s corresponding to solution indices s:

loss functions L_s:	12.5,	210,	220,	310,	375,
solution indices s:	1,	2,	3,	4,	5,

where solution 1 has the best fitness. Figure 6 uses pie charts to compare the roulette wheel selection (RWS) method (in the left panel), the RWS method with the worst solution dropped (in the middle panel), and our EWSS method (in the right panel). Areas associated with solutions are proportional to the selection probabilities, namely, $\{1/L_s : s = 1, \dots, 5\}$ in the left panel, $\{1/L_s : s = 1, \dots, 4\}$ in the middle panel, and $\{(p_s - p_{s-1}) \propto (L_n - L_s) : s = 1, \dots, 4\}$ in the right panel, with p_s defined in (3.3). There, due to the larger area of solution 1 than those of other solutions, the RWS method selects solution 1 much more often than other solutions, and thus can hardly find better solutions. On the contrary, for the EWSS method in the right panel, the following observations are obtained. (i) The area of solution 1 reduces, and all four solutions are possible to be selected. (ii) Moreover, using EWSS, better solutions will always have higher chances to be selected. (iii) Another advantage of the EWSS is that solutions with similar fitness values will have similar chances to be selected. This is visible from solutions 2 and 3, which have fitness values 210 and 220 respectively, and occupy comparable areas. The empirical evidence indicates that the EWSS method can be more efficient than the RWS and "Rank Selection" methods in complicated optimization procedures.

3.2.3 *Crossover operator for* MaxICA. Crossover is another basic operator of the GA. Many crossover techniques [30] have been built to get the optimum solution as fast as possible within minimum generations. Some classic crossover methods are reviewed below.

- The first crossover method is called the "1-point crossover." one of the simplest crossover techniques. This method uses the single point fragmentation of the parents and combines the parents at the selected location to create the offspring. It first selects 2 parents used for crossover and then randomly selects a crossover location. Two offsprings are created by combining the parents at the crossover location, and the values at the crossover location from parents will be averaged. An example is illustrated below, where numerical quantities within a row represent the parameter values of a solution, and the 3rd point is selected to be the crossover location.

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

- The second crossover method is called the "2-point crossover." Similar to the 1-point method, 2 crossover locations are selected, and the offspring is created by combining parents at 2 crossover locations. An example is given below, where the 2nd and 6th points are selected to be crossover locations.

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

- The third crossover method is called the "reduced surrogate crossover." This method minimizes the unwanted crossover operations in case of parents having same genes. In this method, first check for the genes in the parents and create a list of all possible crossover locations where the genes of both parents are different. After the check, if no crossover location is there, then no action is taken. But if parents are differing in more than 1 gene, then one crossover location is randomly selected from the list of all crossover locations and the 1-point crossover is performed.
- The fourth crossover method is called the "uniform crossover." This method provides the uniformity in combining the bits of both parents. It will first choose a random real number u from the uniform distribution between 0 and 1. It creates 2 offsprings of genes selected from both of the parents uniformly. The random real number decides whether the first child selects the genes from the first or the second parent. For a threshold 0.5, select a gene from the first parent if $u < 0.5$, and from the second parent otherwise, followed by repeating this process for all genes.
- The fifth crossover method is called the "average crossover." Average crossover only creates one offspring from 2 parents. The way it creates offspring is taking average of the 2 parents. Each gene in a child is taken by averaging genes from both parents.
- The sixth crossover method is called the "discrete crossover." Similar to the "uniform crossover," the discrete crossover only creates one child from 2 parents. It first chooses a random real number and uses the number to decide the parent whose genes are to be taken for the child.

Selecting a crossover operator has a large impact on the performance of the GA. Our ERD_GA, instead of selecting one specific crossover method, will combine different crossover methods together. This is motivated by the concept of the GA, which simulates the process of chromosome evolution, where various crossover methods are possible and fixing one crossover method may slow down the speed of convergence especially for complicated optimization problems. Thus, in each iteration of the program, we randomly choose one of six crossover methods to be used in that iteration. Our "Random Combined Crossover Operator" is motivated from the "Combined Crossover Operator (CCO)" [16], but differs from CCO as follows.

- In the existing CCO, more than one crossover operator are applied at each generation in a competitive way, and the operator which produces the best result will be finally taken into account. Four crossover operators [16] are applied at each generation, that is, Heuristic crossover, Arithmetic crossover, Simulated binary crossover and Linear BGA crossover, which are distinct from the six crossover methods reviewed above.
- In our "Random CCO," one of the six crossover operators will be randomly chosen to be taken into account at each generation. Choices of crossover operators at each generation are not limited to the six above; other crossover methods are also possible depending on the problem.

3.2.4 *Mutation operator for* MaxICA. The mutation operator also plays a necessary and important role in the GA. Mutation randomly alters each gene with a small probability; usually, a low mutation rate, such as 0.05, will be used. The mutation sometimes could be beneficial, but sometimes may not be. In our numerical experiments, we observe that changing the mutation rate, in case the algorithm gets stuck, could increase the convergence speed. We thus propose a dynamic mutation procedure as follows. If the program is converging fast, a high mutation rate is maintained. On the other hand, if the program gets stuck, and cannot find better solutions for a number of iterations, then reducing the mutation rate can help avoid harmful mutations, and thus will also increase the speed. As a result, the mutation rate is changing dynamically. When no better solutions are found for a number of iterations, the mutation rate will decrease until it reaches the lower bound; otherwise, if the algorithm converges fast, the mutation rate will increase until it reaches the upper bound. See [29] for more information about the dynamic mutation rate operator.

3.3 Summary of the Algorithm for MaxICA

Before applying the ERD_GA algorithm to MaxICA, we address a number of issues, related to the implementation details.

- Similar to the data structure for ICA, the data for the MaxICA 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. For example, if the electroencephalogram (EEG) signals of patients with epilepsy are recorded, components of the brain activity during the seizure period and nonseizure period might differ. Also, recordings from different locations can offer us more information about the hidden components, and thus help us find more accurate results.

Algorithm 3 Summary of the ERD_GA algorithm for MaxICA

```
1: [PCA] Use PCA to decide the number of hidden components;
2: [Piecewise Optimization] Separate the data into pieces and apply the ERD_GA* to each piece respectively.
       *The augmented GA with the elite weighted sum selection, random combined crossover and dynamic mutation. The stopping criterion is: no better solution can be found for 10,000 iterations.
```

- As mentioned in Section 2, the number N of hidden components and numbers $n_1, \dots, n_N$ of sine waves within each of the N components are needed before data analysis. Employing the principle component analysis (PCA), we select the number N of components by the one which accounts for the majority variation of the data. For choices of $n_1, \dots, n_N$, simulation experiments suggest that the choice $n_1 = \cdots = n_N = 5$ applies well to most of the problems.
- To incorporate varying types of components, the data can be separated into smaller pieces beforehand. Our ERD_GA algorithm will be applied to each piece separately.

Algorithm 3 summarizes the ERD_GA algorithm for the MaxICA method. Moreover, the ERD_GA algorithm can be applied to solve other types of global optimization problems, including several benchmark examples listed in [12].

4. SIMULATION STUDIES

We present simulation studies to assess the performance of the MaxICA method in recovering the component signals, as compared with the ICA method.

(a) In the example of Section 4.1, we simulate observed signals from the MaxICA model.

(b) In the example of Section 4.2, we simulate observed signals by adding noise terms to true signals from the ICA model.

4.1 Example 1: Fitting a MaxICA Model

The observed signals $\{X_j(t) : j = 1, \dots, 5\}$ are simulated according to the MaxICA model (2.6), where 5 component signals $\{S_k(t) : k = 1, \dots, 5\}$,

$$\begin{aligned} S_1(t) &= \sin(1.5t + 2) + \sin(0.3t + 1), \\ S_2(t) &= \sin(0.7t + 1.5) + \sin(2t + 2.5), \\ S_3(t) &= \sin(3t + 2) + \sin(t + 0.1), \\ S_4(t) &= \sin(2t + 1.5) + \sin(1.2t + 0.6), \\ S_5(t) &= \sin(t + 0.4) + \sin(3t + 2), \end{aligned} \tag{4.1}$$

are plotted from top to bottom panels in Figure 7, the mixing coefficients $b_{j,k}$ given in the matrix,

$$\begin{pmatrix} b_{1,1} & \cdots & b_{1,5} \\ \vdots & \ddots & \vdots \\ b_{5,1} & \cdots & b_{5,5} \end{pmatrix} = \begin{pmatrix} 2.9821 & 1.794 & 1.8946 & 0.12226 & 1.1915 \\ 1.9397 & 3.4027 & 4.1967 & 3.8037 & 1.1087 \\ 0.31721 & 3.8972 & 3.6087 & 0.97069 & 2.5573 \\ 0.34402 & 1.9707 & 4.9086 & 1.5245 & 3.9098 \\ 3.6735 & 4.1914 & 0.60795 & 4.3392 & 2.8036 \end{pmatrix},$$

are randomly generated from the uniform distribution on $[0, 5]$, and noise terms $\{\epsilon_j(t) : j = 1, \dots, 5\} \overset{\text{i.i.d.}}{\sim} 0.5949\mathbb{N}(0, 1)$.

Before applying MaxICA to the observed data matrix $\mathbf{X}$, we apply PCA to select the number N of hidden components. Figure 8 plots the "fraction of total variance retained," which is the cumulative sum of sorted eigenvalues (of the sample covariance matrix for $\mathbf{X}$) divided by the sum of eigenvalues. There, 5 components account for nearly 95% of the variance, and thus we take $N = 5$ for the number of hidden components.

In the data separation step, the sequence of observations is separated into shorter parts, each part consisting of 500 observation points. The fitted signals (in blue lines) by MaxICA in Figure 9 fit reasonably well the observed signals $X_j(t_i)$ (in dots). As seen from Figure 10, frequencies and amplitudes of the recovered components by MaxICA are similar to those of the true components, though the order may not match, as mentioned in Remark 1. On the other hand, it is visible in Figure 11 that the shape and frequencies of components recovered by ICA don't match well with those of the true components.

The above results obtained from MaxICA were based on the assumption of 5 hidden components. As a comparison, the fitted signals in Figure 12 by MaxICA, assuming 4 hidden components, are comparable to those assuming $N = 5$, but appear to outperform those in Figure 13 assuming 2 hidden components. Indeed, this agrees with Figure 8, where 4 and 2 components account for about 90% and 75% of the variance, respectively.

Recall that MaxICA utilizes the ERD_GA algorithm which contains the EWSS, random combined crossover and dynamic mutation; in contrast, the classical_GA algorithm applies the RWS, single-point crossover and fixed mutation rate. Figure 14 plots the fitted signals via the classical_GA, whose optimization result was generated by running the same number of iterations as that by MaxICA in Figure 9. It is observed that the classical_GA combined with RWS performs as well as ERD_GA, although the classical_GA needs much more iterations before getting a good numerical fit. For another comparison, Figure 15 plots the fitted signals by the "Simulated Annealing" (SA) algorithm (also a well-known optimization method), which performs comparably well. On the other hand, the SA algorithm is slower than the classical_GA, particularly for the visual processing data in Section 5.1, as SA will take longer time before getting a good solution. Moreover, replicating the MaxICA for 100 sets of simulated data signals yields 100 sets of fitted signals (in blue lines) in Figure 16, which follow the shape and trends of true signals (in white lines) in (4.1).

4.2 Example 2: Recovering Components in Noisy ICA Models

In this example, the observed signals in data matrices $\mathbf{X}_1$, $\mathbf{X}_2$ and $\mathbf{X}_3$,

$$\mathbf{X}_1 = \mathrm{A}_1\mathbf{S}_1 + \boldsymbol{\epsilon}_1, \tag{4.2}$$

$$\mathbf{X}_2 = \mathrm{A}_1\mathbf{S}_1 + \boldsymbol{\epsilon}_2, \tag{4.3}$$

$$\mathbf{X}_3 = \mathrm{A}_2\mathbf{S}_2 + \boldsymbol{\epsilon}_1, \tag{4.4}$$

are simulated from noisy ICA models, with linearly combined component signals,

$$S_1(t) = \sin(2t + 0.3), \qquad S_2(t) = \sin(0.5t + 1), \tag{4.5}$$

for $\mathbf{S}_1$, and

$$S_1(t) = \sin^2(2t + 0.3), \qquad S_2(t) = \sin(0.5t + 1),$$

for $\mathbf{S}_2$, where the mixing matrices are

$$\mathrm{A}_1 = \begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}, \qquad \mathrm{A}_2 = \begin{pmatrix} 5 & 2 \\ 2 & 8 \end{pmatrix},$$

and the noise matrix $\boldsymbol{\epsilon}_1$ contain entries $\overset{\text{i.i.d.}}{\sim} 0.2\mathbb{N}(0, 1)$, and the noise matrix $\boldsymbol{\epsilon}_2$ contain entries $\overset{\text{i.i.d.}}{\sim} 0.8\mathbb{N}(0, 1)$.

Figure 17 displays the recovered components (in blue lines) by ICA and MaxICA for $\mathbf{X}_1$. They both well retrieve the set $\{S_1(t), S_2(t)\}$ of true components (in red dashed lines) in (4.5). An examination of the fitted signals (in blue lines) to the observed signals (in dots) in Figure 18 indicates that MaxICA mitigates the overfitting problem of ICA. A similar observation can be made in Figure 20 for $\mathbf{X}_2$ and Figure 22 for $\mathbf{X}_3$.

For the simulated $\mathbf{X}_2$ signals (in dots) plotted in Figure 20, noise terms in model (4.3) have larger variances than those in model (4.2). We observe from the bottom-right panel of Figure 19 that the recovered component (in blue lines) by ICA follows the trend of one of the true components $\{S_1(t), S_2(t)\}$ (in red dashed lines) in (4.5). But the top-right panel of Figure 19 reveals that the other component recovered by ICA doesn't follow the shape of either $S_1(t)$ or $S_2(t)$. Again, as seen from Figure 19, recovered components by MaxICA mimic true components $S_1(t)$ and $S_2(t)$.

Observed signals (in dots) in $\mathbf{X}_3$ plotted in Figure 22 are generated from model (4.4) with mixing coefficients in A_2, which contains some entries (i.e., 5 and 8) larger than those in A_1. Figure 21 illustrates that extracted components (in blue lines) by ICA can be treated as corresponding to the true components $\{S_1(t), S_2(t)\}$ (in red dashed lines). For MaxICA, the fitted signals (in blue lines) in Figure 22 appear to follow trends of the observed signals (in dots). Nonetheless, the top-left panel of Figure 21 reveals that MaxICA is not effective in tracking patterns of the true component $S_1(t)$, though the other true component $S_2(t)$ is well extracted in the bottom-left panel.

In summary, for linearly combined component signals, ICA performs very well in separating the components from observed signals, whereas MaxICA can also recover the information about linearly combined components, even in the presence of large noises. Nonetheless, for certain more complex components, MaxICA may not extract as well as ICA. Analogously, for max-linearly combined components, refer to Figure 4 for the advantages of MaxICA over ICA. Due to different types of applications to which ICA and MaxICA are suitable, it is not suggested to either replace ICA by MaxICA, or substitute MaxICA by ICA.

5. REAL DATA ANALYSIS

We analyze two real datasets. For the visual processing data, we apply MaxICA and ICA respectively to compare their performances. For the epilepsy data, we try to recover hidden brain components to compare different activity patterns in different brain areas.

5.1 Visual Processing Data

This EEG (Electroencephalography) data collect 32-channels from 14 subjects (7 males, 7 females) acquired using the Neuroscan software. Subjects participated in 2 tasks: one was a go-nogo categorization task, and the other one was a go-nogo recognition task. Tasks were operated by presenting natural photographs in front of subjects very briefly (20 ms). Each subject responded to 2500 trials. The data was sampled at 1000 Hz [7].

In the experiment, participants were seated 110 cm away from a computer screen in a dark room. In the 2 tasks, images were equally likely to be presented to participants. The experiment lasted 2 days, yielding 13 series on the first day and 12 series on the second day. Each series includes 100 images. The participant needed to press a button to start a series. A small point was drawn in the middle of the screen, the images were presented at the point very fast, and participants had to stare at the point. This design can help reduce the effect of eye movement. For each targeted image, participants lifted 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 nontarget images as well, including pictures of natural landscapes, city scenes, food, trees, etc. In the recognition task, participants needed to learn the target image in the learning phase in order to recognize it in the following test phase. Participants had no *a priori* information about the pictures. See [10] and [8] for more details about the experiment.

Electric brain potentials were recorded from 32 electrodes mounted on an elastic cap (Oxford Instruments). Electrode Cz was used as a reference and a mastoid electrode was used as ground, so we have 31 observed EEG 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 the left panel of Figure 23, with 31 channels for 31 sequences of data signals used for analysis. The reference electrode Cz is in the center, but is not plotted in the figure.

The EEG is person-specific data [27], which means EEG of different people varies. Cerebral characteristics could be different between subjects, and the right panel of Figure 23 is the beginning part of data signals at all 31 channels.

We first analyze the categorization task data from the first subject. Figure 24 indicates that the first 5 components account for nearly 90% of the variance, and thus we assume 5 hidden components for MaxICA. We separate the data into parts, each part covering 500 observation points. As the data was sampled at 1000 Hz, the range of 500 points is 500 ms.

Figure 25 presents a partial result of fitted signals, displaying the fitted signals (in blue lines) by MaxICA of the first 6000 points from the first 2 observed signals (in dots). The two peaks in the observed signals can be clearly spotted from the fitted signals by MaxICA. Those peaks are related to important brain activities. Thus, the components recovered by MaxICA could be scientifically meaningful. As a comparison, the fitted signals (in blue lines) by ICA in Figure 26 reproduce the observed signals (in dots), exemplifying the overfitting problem. Figure 27 plots the 5 hidden components recovered by MaxICA. The frequencies and patterns of these components can be helpful for revealing the activity of brain. As a comparison, components recovered by ICA are plotted in Figure 28. One component recovered by ICA, displayed in the top panel of Figure 28, basically mimics (with an opposite sign) the observed trend in the real data; the other recovered components by ICA in the other four panels contain many high-frequency oscillations. As another comparison, Figure 29 plots the fitted signals to this data using the simulated annealing (SA) algorithm, which is not as good as the ERD_GA. Actually, simulated annealing will take much longer time to find the global optimum. Figure 30 and Figure 31 plot the fitted signals by MaxICA to observed signals of subject 1's recognition task and subject 2's categorization task, respectively. These figures reveal that MaxICA preforms well in the fits of signals containing important events.

5.2 Epilepsy Data

In this section, we study the intracranial EEG recordings from five epilepsy patients [1]. 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, while the other two only had auras but no other seizures following surgery.

All EEG signals were digitally band-pass filtered between 0.5 Hz 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, were classified as focal EEG channels, while all other channels in the recordings were classified as nonfocal EEG channels. Though visual analysis is not a perfect method, it is still the most important method for helping clinical diagnosis. Pairs of simultaneously recorded signals were randomly selected from focal and nonfocal EEG channels respectively; refers to [1] for the selection rule.

First, we analyze a pair of signals from focal channels. We assume 2 hidden components as indicated by PCA in Figure 32. We observe from Figure 33 that MaxICA provides good fits (in blue lines) of the observed signals (in dots); the fitted signals track most of major peaks. Figure 34 and Figure 35 plot the components recovered by MaxICA and ICA, respectively. The components generated by these two methods appear to have similar characteristics, frequencies, amplitudes, and trending. We next use a pair of EEG signals from nonfocal channels. Figure 36 plots the fitted signals by MaxICA, and Figure 37 and Figure 38 present components recovered by MaxICA and ICA respectively. Again, we observe that two different methods extract similar components for the nonfocal data.

Next, we apply MaxICA to other signal pairs. For another signal pair from focal channels, the fitted signals (in blue lines) by MaxICA are given in Figure 39. Likewise, for another signal pair from nonfocal channels, Figure 40 gives the fitted signals by MaxICA. In both figures, dots are observed signals. It is observed that fitted signals follow the trends of observed signals, and catch most of the major peaks.

6. CONCLUSIONS

In this paper, we developed a new MaxICA method for blind source separation, potentially capable of discovering some dominating components or characteristics hidden in observed signals, especially the EEG data. The tomography structured blind source separation of MaxICA may lead to scientific discoveries. We have demonstrated in simulation experiments and real data analysis that MaxICA can be widely applicable in scientific research. For example, the MaxICA performs well in simulations of Section 4.1 and real data analysis of Section 5. For the visual processing data, the fitted signals by ICA in Figure 26 reproduce the observed signals; one recovered component by ICA displayed in the top panel of Figure 28 basically mimics (with an opposite sign) the observed trend in the real data, and the other recovered components by ICA in the other four panels contain many high-frequency oscillations. A similar phenomenon can be observed from Figure 35 for the Epilepsy data, though not as obvious as those in Figure 28. These numerical evidences together with the overfitting problem of ICA, from Figures 4, 18, 20 and 22, suggest that ICA may not suit some applications. In contrast, the MaxICA method found the major hidden components and stated a model for hidden sources that ICA cannot provide.

The existing methods for finding blind sources are widely used but have some limitations. For example, ICA cannot be well applied, if the number p of observed signals is fewer than the number N of hidden components [4, 15, 20, 28]. Moreover, ICA assumes mutual independence and joint non-Gaussianity for the component variables. The MaxICA model removes these constraints and thus is more flexible in applications.

In practice, the choice between ICA and MaxICA may be made on a case-by-case basis. In the visual processing application, to decide whether or not a subject is an animal, multiple neurons/regions may simultaneously process the signal (picture), and a dominatedly (strongest) processed signal from a region may be recorded. Comparing the component signals recovered by the MaxICA and ICA methods, one can notice from Figure 28 that one component signal attained from ICA contains two jumps, while none of the recovered components by MaxICA does. It may be hard to understand a hidden component signal to have two jumps in visual processing experiments. In the epilepsy study, it is desirable to find a hidden dominant factor (brain activity) which causes the disorder.

In practice, when the dataset exhibits some abnormality at some time points or there are some prior knowledge that some hidden components can be dominating factors, the MaxICA model can be a good candidate. Nonetheless, as an exploratory analysis tool, MaxICA is widely applicable.

Due to the nonlinear formulation and special features of the MaxICA model, many existing optimization methods could neither be directly applicable to parameter learning nor deliver expected results. The developed ERD_GA algorithm for the MaxICA model is a feasible approach. Our numerical experiments indicate that ERD_GA is more adaptive and flexible than the classical_GA in fitting time series data, and outperforms the SA algorithm with better fits and faster implementation. Issues on investigating the algorithmic convergence of the ERD_GA and developing other more efficient algorithms in the optimization task are desirable for future research. Moreover, models (2.6) and (2.8) can be extended to incorporate functional regression components and dependent noise processes.

The MaxICA model proposed in this paper can be used to explore and address many scientific questions. In future studies, questions include which brain regions are most responsive to visual processing and which brain regions and their connectivity are relevant to the epilepsy. We wish to collaborate with neuroscientists and medical doctors on some brain science projects and try to find dominating hidden component factors, and brain regions.

ACKNOWLEDGMENTS

The authors greatly appreciate Professor David Siegmund and an anonymous referee for insightful comments. C. Zhang's research is supported by the U.S. NSF Grant DMS-1712418 and Wisconsin Alumni Research Foundation. Z. Zhang's research is supported by U.S. NSF Grants DMS-1505367, NSF CMMI-1536978 and Wisconsin Alumni Research Foundation #MSN215758.

SUPPLEMENTARY MATERIAL

Supplement to "Maximum Independent Component Analysis with Application to EEG **Data."** (DOI: 10.1214/19-STS763SUPP; .pdf). The online supplementary file collects notations and all figures in the paper.

REFERENCES

[1] ANDRZEJAK, R. G., SCHINDLER, K. and RUMMEL, C. (2012). Nonrandomness, nonlinear dependence, and nonstationarity of electroencephalographic recordings from epilepsy patients. *Phys. Rev. E* **86** 046206.

[2] ARTONI, F., DELORME, A. and MAKEIG, S. (2018). Applying dimension reduction to EEG data by principal component analysis reduces the quality of its subsequent independent component decomposition. *NeuroImage* **175** 176–187. https://doi.org/10.1016/j.neuroimage.2018.03.016

[3] BRONKHORST, A. W. (2000). The cocktail party phenomenon: A review of research on speech intelligibility in multiple-talker conditions. *Acta Acust. Acust.* **86** 117–128.

[4] CHEN, A. and BICKEL, P. J. (2006). Efficient independent component analysis. *Ann. Statist.* **34** 2825–2855. MR2329469 https://doi.org/10.1214/009053606000000939

[5] CRAMÉR, H. and LEADBETTER, M. R. (1967). *Stationary and Related Stochastic Processes. Sample Function Properties and Their Applications*. Wiley, New York. MR0217860

[6] CRYER, J. D. and CHAN, K.-S. (2008). Time series regression models. In *Time Series Analysis with Applications in R*, Chapter 11 249–276.

[7] DELORME, A., MAKEIG, S., FABRE-THORPE, M. and SEJNOWSKI, T. (2002). From single-trial eeg to brain area dynamics. *Neurocomputing* **44** 1057–1064.

[8] DELORME, A., ROUSSELET, G. A., MACE, M. J.-M. and FABRE-THORPE, M. (2004). Interaction of top-down and bottom-up processing in the fast visual analysis of natural scenes. *Cogn. Brain Res.* **19** 103–113.

[9] DRISS, I., MOUSS, K. N. and LAGGOUN, A. (2015). A new genetic algorithm for exible job-shop scheduling problems. *J. Mech. Sci. Technol.* **29** 1273–1281.

[10] FABRE-THORPE, M., DELORME, A., MARLOT, C. and THORPE, S. (2001). A limit to the speed of processing in ultra-rapid visual categorization of novel natural scenes. *J. Cogn. Neurosci.* **13** 171–180.

[11] FAN, J. and YAO, Q. (2003). *Nonlinear Time Series: Nonparametric and Parametric Methods. Springer Series in Statistics*. Springer, New York. MR1964455 https://doi.org/10.1007/b97702

[12] GAO, S., SHI, L. and ZHANG, Z. (2018). A peak-over-threshold search method for global optimization. *Automatica J. IFAC* **89** 83–91. MR3762035 https://doi.org/10.1016/j.automatica.2017.12.002

[13] GOLDBERG, D. E. and DEB, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms* (*Bloomington, IN*, 1990) 69–93. Morgan Kaufmann, San Mateo, CA. MR1147425

[14] GUO, R., ZHANG, C. and ZHANG, Z. (2020). Supplement to "Maximum independent component analysis with application to EEG data." https://doi.org/10.1214/19-STS763SUPP.

[15] GUO, Y. and TANG, L. (2013). A hierarchical model for probabilistic independent component analysis of multi-subject fMRI studies. *Biometrics* **69** 970–981. MR3146792 https://doi.org/10.1111/biom.12068

[16] HASSAN, I. I. (2015). Combined crossover operator. *Res. J. Appl. Sci.* **10** 75–79.

[17] HAUPT, R. L. and HAUPT, S. E. (1998). *Practical Genetic Algorithms. A Wiley-Interscience Publication*. Wiley, New York. MR1491878

[18] HUANG, H., LU, J., WU, J., DING, Z., CHEN, S., DUAN, L., CUI, J., CHEN, F., KANG, D. et al. (2018). Tumor tissue detection using blood-oxygen-level-dependent functional mri based on independent component analysis. *Sci. Rep.* **8**. Article number: 1223.

[19] HYVÄRINEN, A., KARHUNEN, J. and OJA, E. (2001). *Independent Component Analysis*. Wiley, New York.

[20] HYVÄRINEN, A. and OJA, E. (2000). Independent component analysis: Algorithms and applications. *Neural Netw.* **13** 411–430.

[21] JEBARI, K. and MADIAFI, M. (2013). Selection methods for genetic algorithms. *Int. J. Emerg. Sci.* **3** 333–344.

[22] KASSOUF, A., BOUVERESSE, D. J.-R. and RUTLEDGE, D. N. (2018). Determination of the optimal number of components in independent components analysis. *Talanta* **179** 538–545. https://doi.org/10.1016/j.talanta.2017.11.051

[23] NASCIMENTO, M., SILVA, F., SAFADI, T., NASCIMENTO, A. C. C., FERREIRA, T. E. M., BARROSO, L. M. A., AZEVEDO, C. F., GUIMAR AES, S. E. and SER AO, N. V. (2017). Independent component analysis (ica) based-clustering of temporal rna-seq data. *PLoS ONE* **12** e0181195.

[24] NAVEAU, P., ZHANG, Z. and ZHU, B. (2011). An extension of max autoregressive models. *Stat. Interface* **4** 253–266. MR2812820 https://doi.org/10.4310/SII.2011.v4.n2.a19

[25] OBITKO, M., SLAVIK, P. and WALTER, P. (1998). *Introduction to genetic algorithms*. http://www.obitko.com/tutorials/genetic-algorithms/index.php.

[26] PAES, F. G., PESSOA, A. A. and VIDAL, T. (2017). A hybrid genetic algorithm with decomposition phases for the unequal area facility layout problem. *European J. Oper. Res.* **256** 742–756. MR3549773 https://doi.org/10.1016/j.ejor.2016.07.022

[27] SHOEB, A. H. (2009). Application of machine learning to epileptic seizure onset detection and treatment Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.

[28] SOKOL, A., MAATHUIS, M. H. and FALKEBORG, B. (2014). Quantifying identifiability in independent component analysis. *Electron. J. Stat.* **8** 1438–1459. MR3263128 https://doi.org/10.1214/14-EJS932

[29] THIERENS, D. (2002). Adaptive mutation rate control schemes in genetic algorithms. In *Evolutionary Computation*, 2002. *CEC'*02. *Proceedings of the* 2002 *Congress on* **1** 980–985. IEEE, New York.

[30] UMBARKAR, A. and SHETH, P. (2015). Crossover operators in genetic algorithms: A review. *ICTACT J. Soft Comput.* **06** 1083–1092.

[31] ZHANG, C., CHAI, Y., GUO, X., GAO, M., DEVILBISS, D. and ZHANG, Z. (2016). Statistical learning of neuronal functional connectivity. *Technometrics* **58** 350–359. MR3520664 https://doi.org/10.1080/00401706.2016.1142904

Submitted to Statistical Science

Maximum independent component analysis with application to EEG data

Ruosi Guo Chunming Zhang and Zhengjun Zhang

University of Wisconsin at Madison

APPENDIX A: LIST OF SYMBOLS AND NOTATIONS

- t: scalar denoting a generic time point in the interval $[0, \mathrm{T}]$.
- $j = 1, \dots, p$: p is the number of observed signals;
  $k = 1, \dots, N$: N is the number of hidden components;
  $\ell = 1, \dots, n_k$: n_k is the number of sine waves within the kth component;
  $i = 1, \dots, m$: m is the number of time points;
- $t_1, \dots, t_m$: time points in each of p sequences of observed signals.
- $\mathrm{A} = (a_{j,k})_{j=1,\dots,p;\, k=1,\dots,N} \in \mathbb{R}^{p\times N}$ in the ICA model (2.3):
$$\mathrm{A} = \big(\boldsymbol{a}_1, \dots, \boldsymbol{a}_N\big) \in \mathbb{R}^{p\times N}.$$
- $\mathrm{B} = (b_{j,k})_{j=1,\dots,p;\, k=1,\dots,N} \in \mathbb{R}^{p\times N}$ in the MaxICA model (2.6):
- The matrix of observed signals is
$$\mathbf{X} = \begin{pmatrix} X_1(t_m) & \cdots & X_1(t_m) \\ \vdots & \dots & \vdots \\ X_p(t_m) & \cdots & X_p(t_m) \end{pmatrix} = \big(\boldsymbol{X}(t_1), \dots, \boldsymbol{X}(t_m)\big) = \begin{pmatrix} \vec{\boldsymbol{X}}_1^T \\ \vdots \\ \vec{\boldsymbol{X}}_p^T \end{pmatrix} \in \mathbb{R}^{p\times m},$$
where
$$\boldsymbol{X}(t) = \begin{pmatrix} X_1(t) \\ \vdots \\ X_p(t) \end{pmatrix} \in \mathbb{R}^{p\times 1}, \quad \vec{\boldsymbol{X}}_j = \begin{pmatrix} X_j(t_1) \\ \vdots \\ X_j(t_m) \end{pmatrix} \in \mathbb{R}^{m\times 1},\ j = 1, \dots, p.$$
- The matrix of component signals is
$$\mathbf{S} = \begin{pmatrix} S_1(t_m) & \cdots & S_1(t_m) \\ \vdots & \dots & \vdots \\ S_N(t_m) & \cdots & S_N(t_m) \end{pmatrix} = \big(\boldsymbol{S}(t_1), \dots, \boldsymbol{S}(t_m)\big) = \begin{pmatrix} \vec{\boldsymbol{S}}_1^T \\ \vdots \\ \vec{\boldsymbol{S}}_N^T \end{pmatrix} \in \mathbb{R}^{N\times m},$$
where
$$\boldsymbol{S}(t) = \begin{pmatrix} S_1(t) \\ \vdots \\ S_N(t) \end{pmatrix} \in \mathbb{R}^{N\times 1}, \quad \vec{\boldsymbol{S}}_k = \begin{pmatrix} S_k(t_1) \\ \vdots \\ S_k(t_m) \end{pmatrix} \in \mathbb{R}^{m\times 1},\ k = 1, \dots, N.$$

Department of Statistics, University of Wisconsin, Madison, 53706, USA (e-mail: rguo25@wisc.edu). Corresponding author; Department of Statistics, University of Wisconsin, Madison, 53706, USA (e-mail: cmzhang@stat.wisc.edu). Department of Statistics, University of Wisconsin, Madison, 53706, USA (e-mail: zjz@stat.wisc.edu).

- $\vec{\boldsymbol{f}}_j(\boldsymbol{\theta}) = (f_j(\boldsymbol{\theta}, t_1), \ldots, f_j(\boldsymbol{\theta}, t_m))^T$, $j = 1, \ldots, p$, in Section 2.4: with $f_j(\boldsymbol{\theta}, t) = \max\{b_{j,1}\, S_1(t), b_{j,2}\, S_2(t), \ldots, b_{j,N}\, S_N(t)\} + \gamma$ and $\boldsymbol{\theta} = \{\{(\alpha_{k,\ell}, \beta_{k,\ell})\}_{\ell=1,\ldots,n_k;\, k=1,\ldots,N};\ \{b_{j,k}\}_{k=1,\ldots,N;\, j=1,\ldots,p};\ \gamma\}$.
- $L(\boldsymbol{\theta}) = \sum_{j=1}^{p} \|\vec{\boldsymbol{X}}_j - \vec{\boldsymbol{f}}_j(\boldsymbol{\theta})\|_2$ in (2.12): loss function of $\boldsymbol{\theta}$.
- $L_s = L(\widehat{\boldsymbol{\theta}}^{(s)})$ in (3.1): loss function calculated from the sth solution $\widehat{\boldsymbol{\theta}}^{(s)}$, $s = 1, \ldots, n$.
- In Algorithm 2, $S = \sum_{s=1}^{n} \mathfrak{f}_s$, with fitness values $\mathfrak{f}_1, \ldots, \mathfrak{f}_n$ of n chromosomes in the population.

APPENDIX B: FIGURES IN THE PAPER

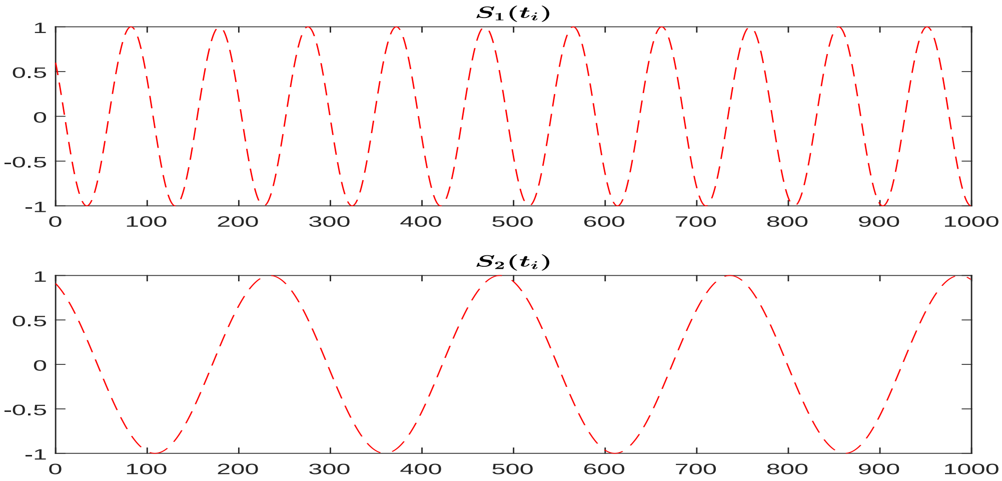


FIG 1. (***Example in Section*** 1.1) *True component signals* $S_1(t_i)$ *and* $S_2(t_i)$ *defined in* (1.1).

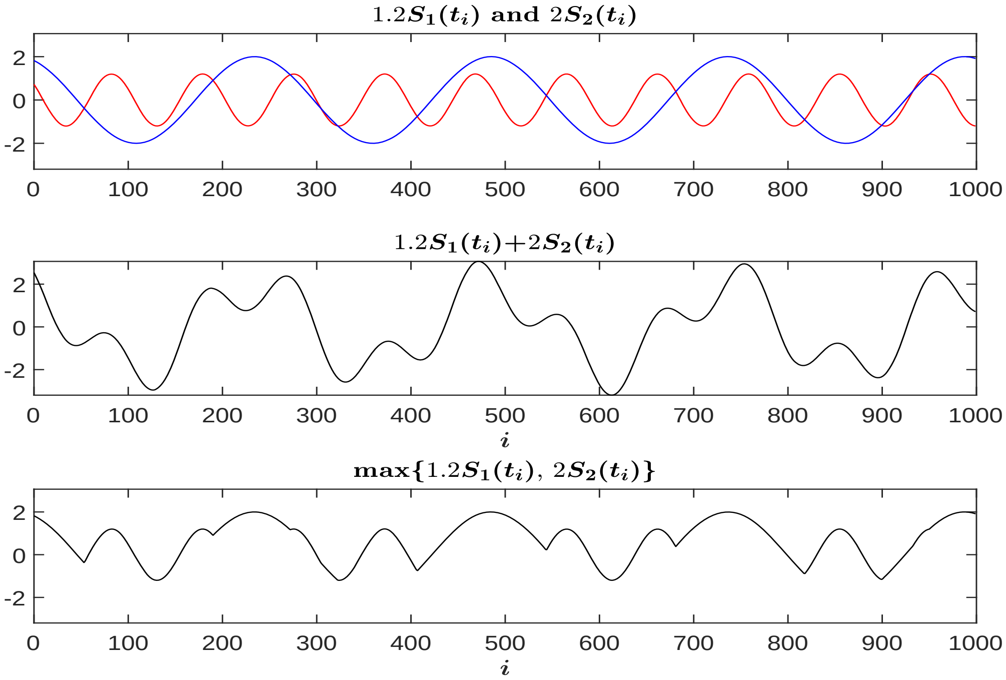


FIG 2. (***Example in Section*** 1.1) *An example illustrating differences between the "linear" combination and the "max-linear" combination. Top panel: 2 sequences of source signals with components* $S_1(t_i)$ *and* $S_2(t_i)$ *in Figure* 1*; middle panel: the "sum" of source signals; bottom panel: the "maxima" of source signals.*

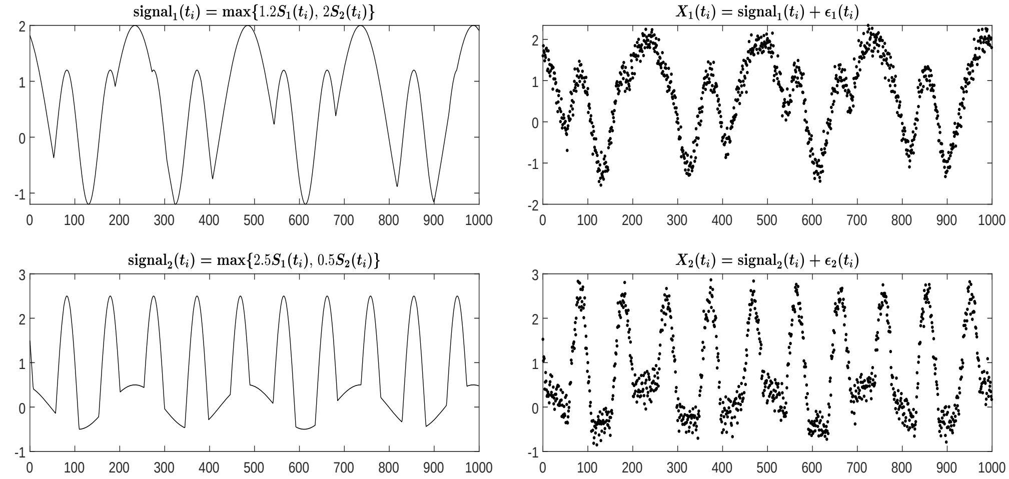


FIG 3. (***Example in Section*** 1.1) *Left panels: true signals* $\text{signal}_j(t_i)$ *in* (1.3), *with components* $S_1(t_i)$ *and* $S_2(t_i)$ *in Figure* 1. *Right panels: observed signals* $X_j(t_i)$ *equal to true signals* $\text{signal}_j(t_i)$ *plus random noises* $\epsilon_j(t_i)$.

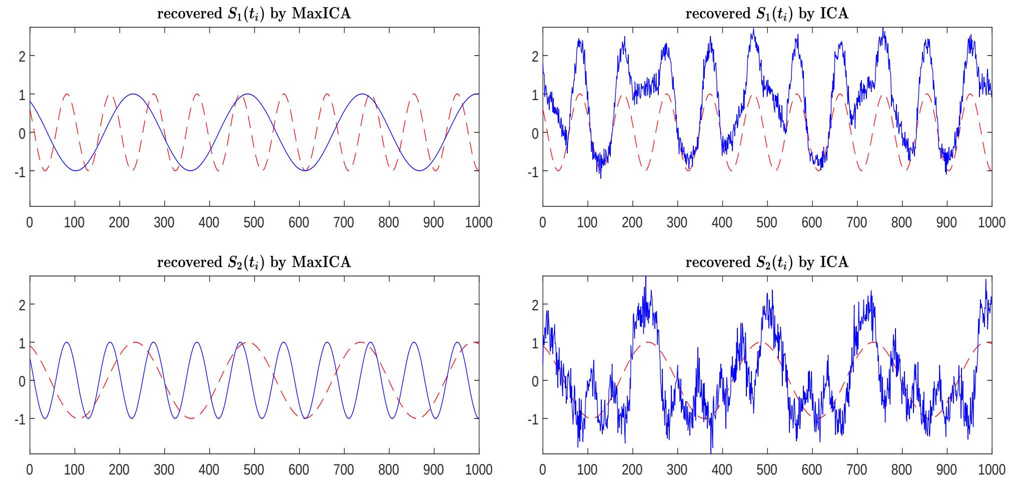


FIG 4. (***Example in Section*** 1.1) *Compare true components (in red dashed lines) and recovered components (in blue lines), by* MaxICA *(in left panels) and* ICA *(in right panels), from observed signals in Figure* 3. *See Remark* 1 *for the order of recovered components.*

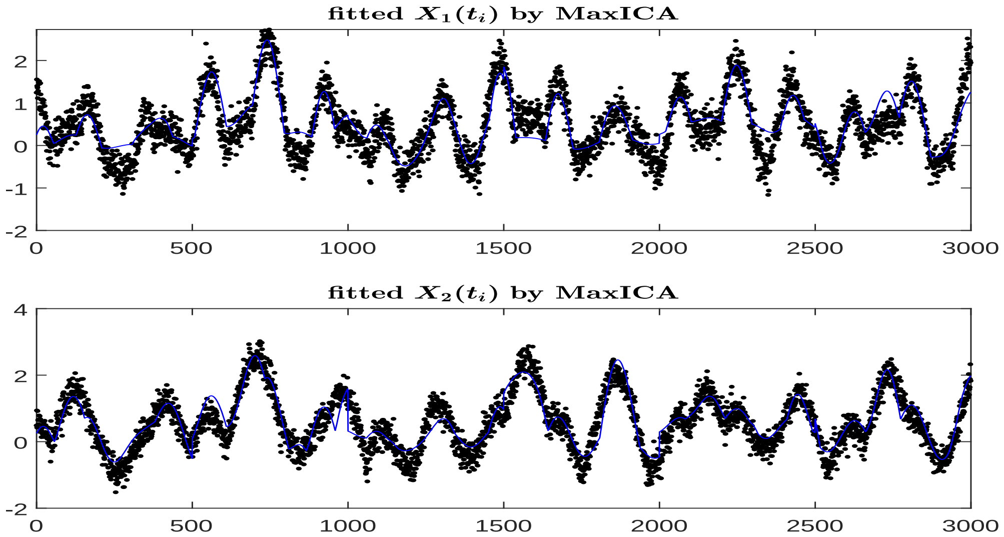


FIG 5. (***Example in Section*** 2.3) *Fitted signals (in blue lines), by the* MaxICA *model* (2.10) *with random amplitudes in Section* 2.3*, to observed signals (in black dots).*

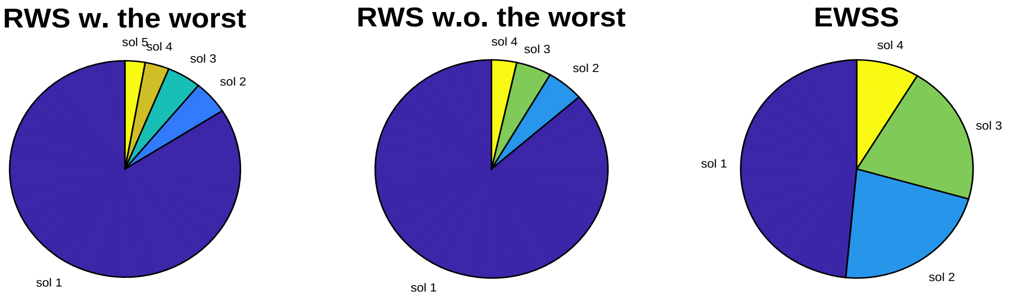


FIG 6. (***Example in Section*** 3.2.2) *Comparison of selection methods in Section* 3.2.2*. Left: areas are proportional to* $\{1/L_s : s = 1, \ldots, 5\}$*; middle: areas are proportional to* $\{1/L_s : s = 1, \ldots, 4\}$*; right: areas are proportional to* $\{(L_n - L_s) : s = 1, \ldots, 4\}$.

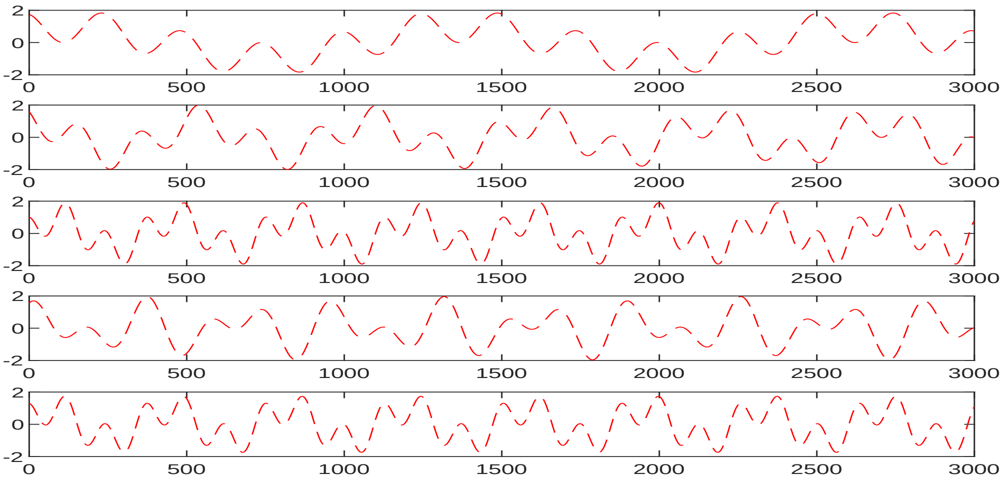


FIG 7. (***Example in Section*** 4.1) *True component signals* $S_1(t), \ldots, S_5(t)$ *in* (4.1).

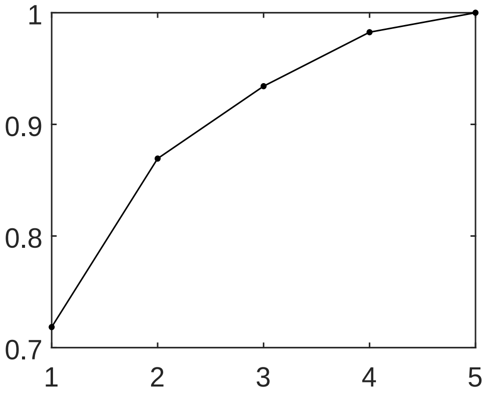


Fig 8. (***Example in Section*** 4.1) *Fraction of total variance retained versus the number of eigenvalues.*

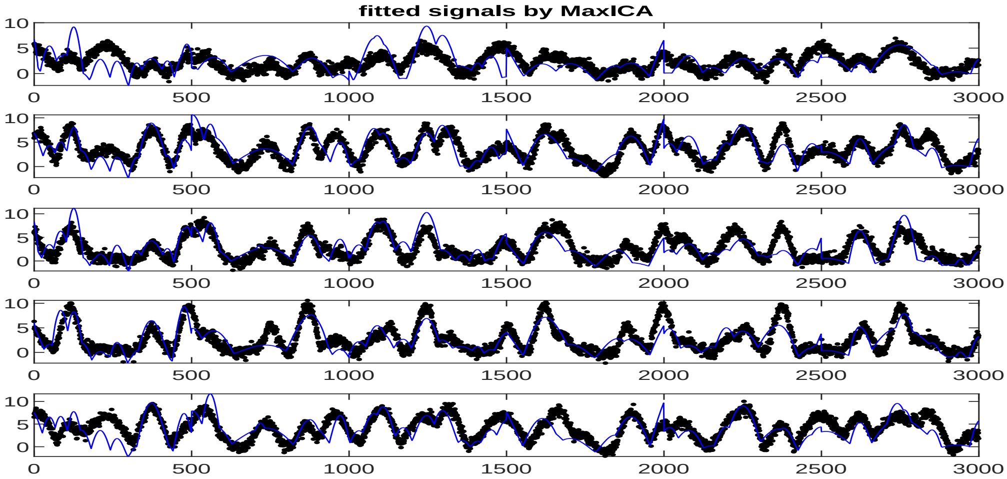


Fig 9. (***Example in Section*** 4.1) *Fitted signals (in blue lines), by* MaxICA *assuming* 5 *hidden components in the* MaxICA *model, to observed signals (in black dots).*

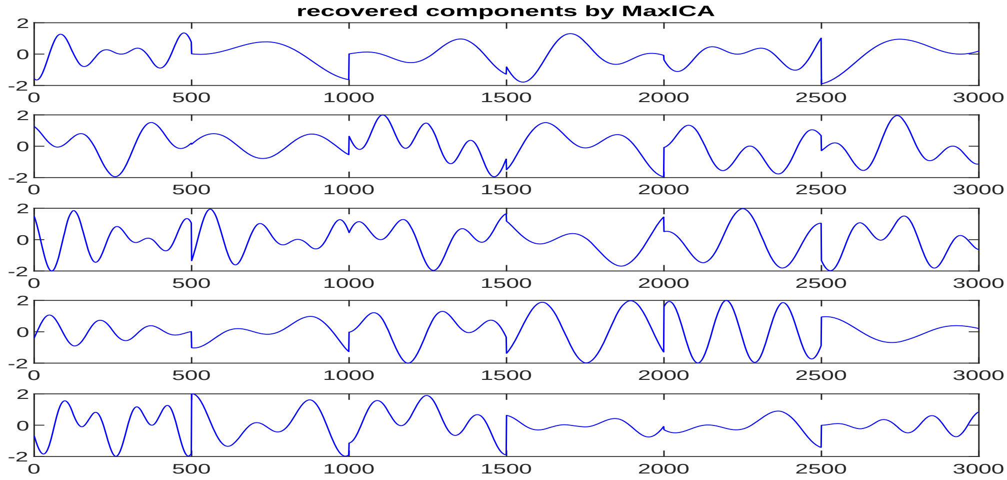


Fig 10. (***Example in Section*** 4.1) *Recovered components (in blue lines) by* MaxICA *in the* MaxICA *model.*

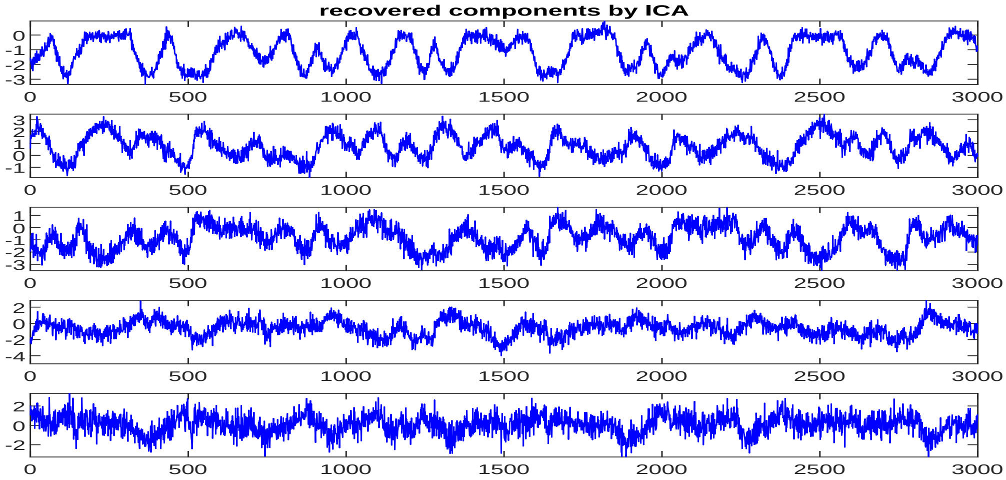


Fig 11. (***Example in Section*** 4.1) *Recovered components* (*in blue lines*) *by* ICA *in the* MaxICA *model.*

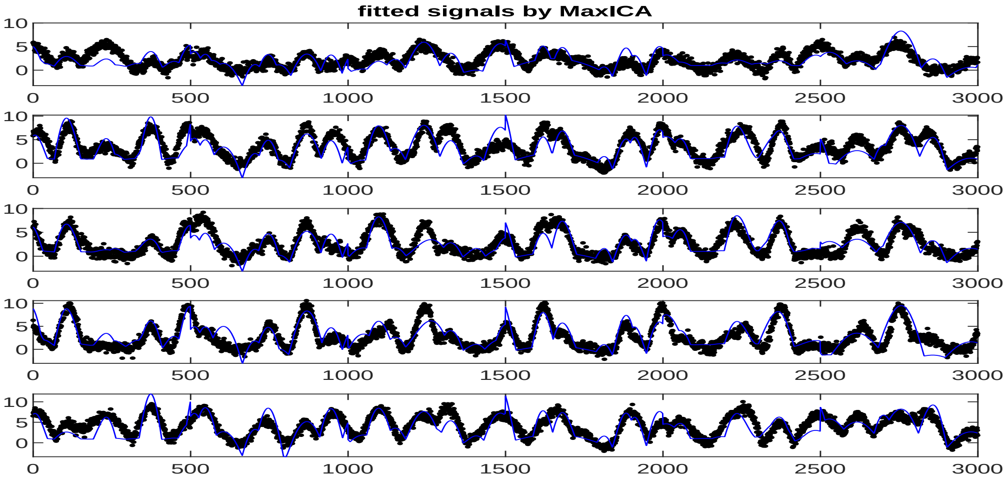


Fig 12. (***Example in Section*** 4.1) *Fitted signals* (*in blue lines*), *by* MaxICA *assuming* 4 *hidden components in the* MaxICA *model, to observed signals* (*in black dots*).

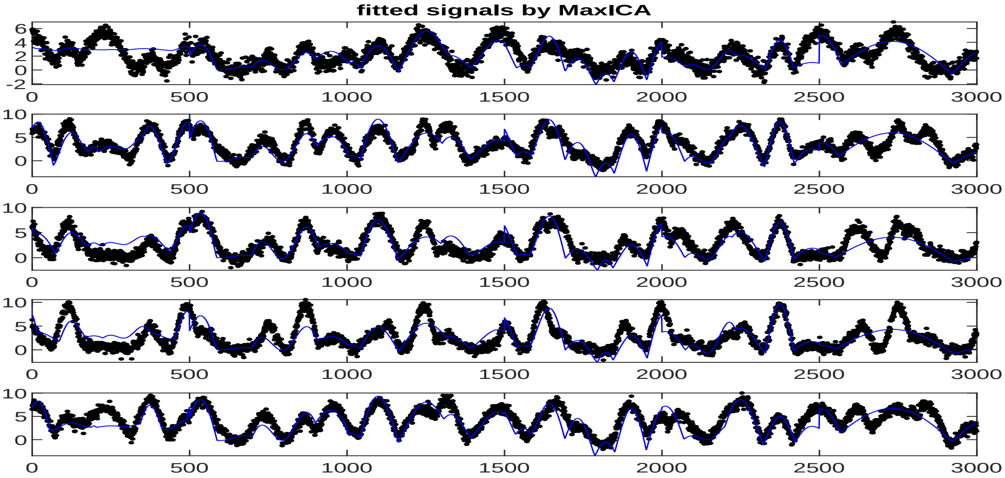


Fig 13. (***Example in Section*** 4.1) *Fitted signals* (*in blue lines*), *by* MaxICA *assuming* 2 *hidden components in the* MaxICA *model, to observed signals* (*in black dots*).

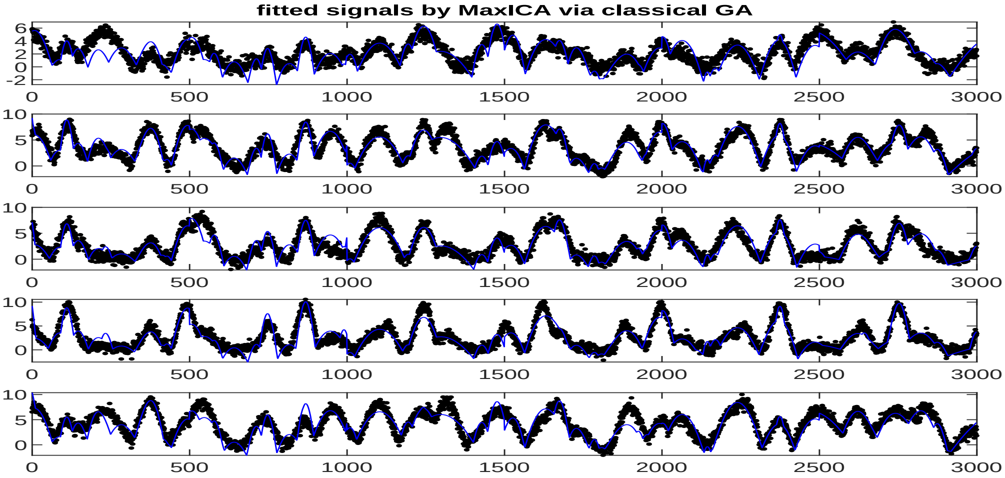


Fig 14. (***Example in Section*** 4.1) *The caption is similar to that of Figure* 9, *except that the* classical_GA *algorithm* (*using the roulette wheel selection, single-point crossover and fixed mutation rate*) *is used.*

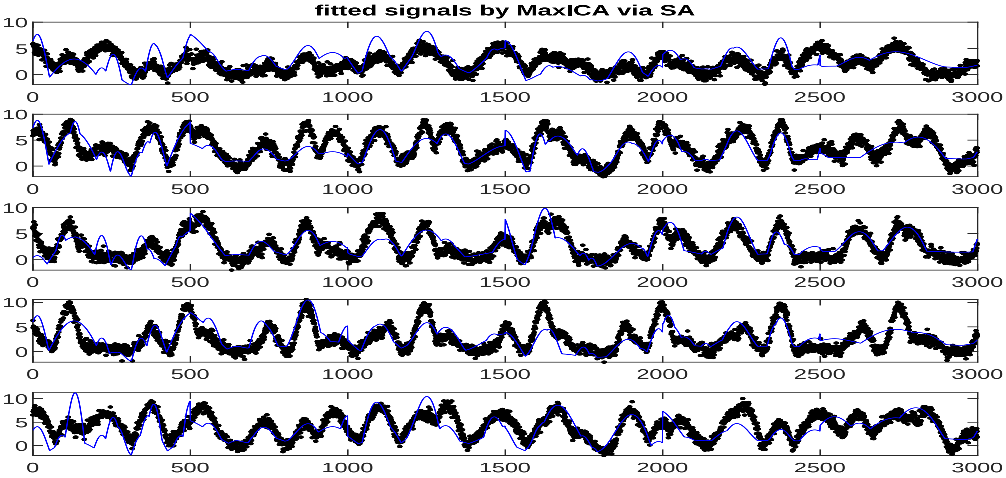


FIG 15. (***Example in Section*** 4.1) *The caption is similar to that of Figure* 9, *except that the Simulated Annealing* (SA) *algorithm is used.*

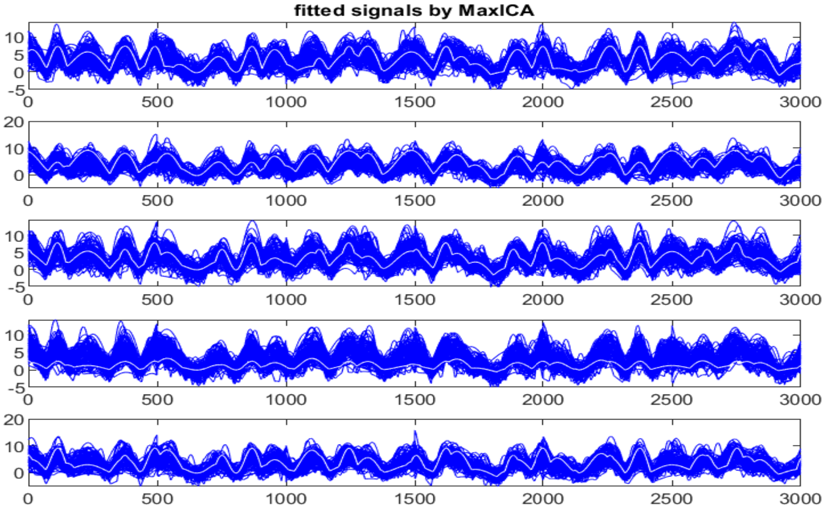


FIG 16. (***Example in Section*** 4.1) *Fitted signals* (*in blue lines*), *by* MaxICA *assuming* 5 *hidden components in the* MaxICA *model, replicated for* 100 *sets of simulated data signals, where white lines are for true signals.*

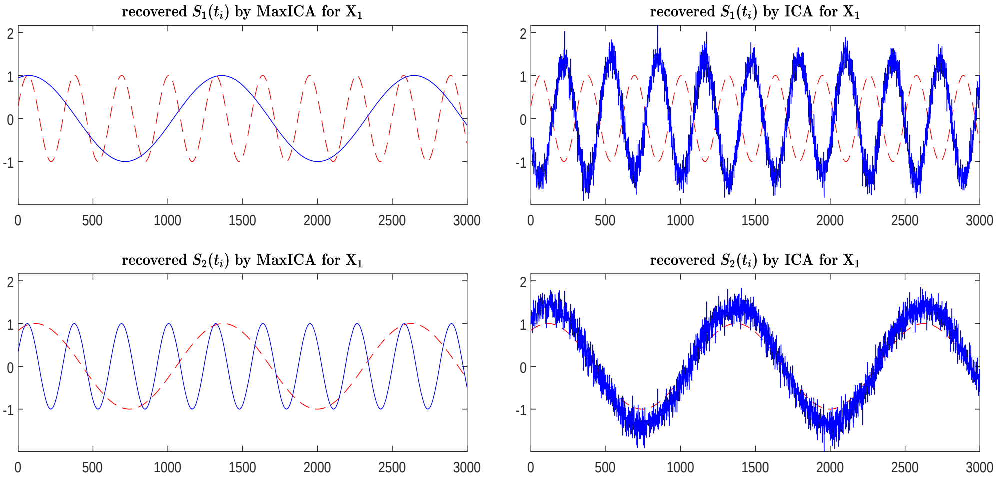


FIG 17. (***Example in Section*** 4.2) *Compare true components (in red dashed lines) and recovered components (in blue lines), by* MaxICA *(in left panels) and* ICA *(in right panels), from simulated data signals of* $\mathbf{X}_1$ *in* (4.2). *See Remark* 1 *for the order of recovered components.*

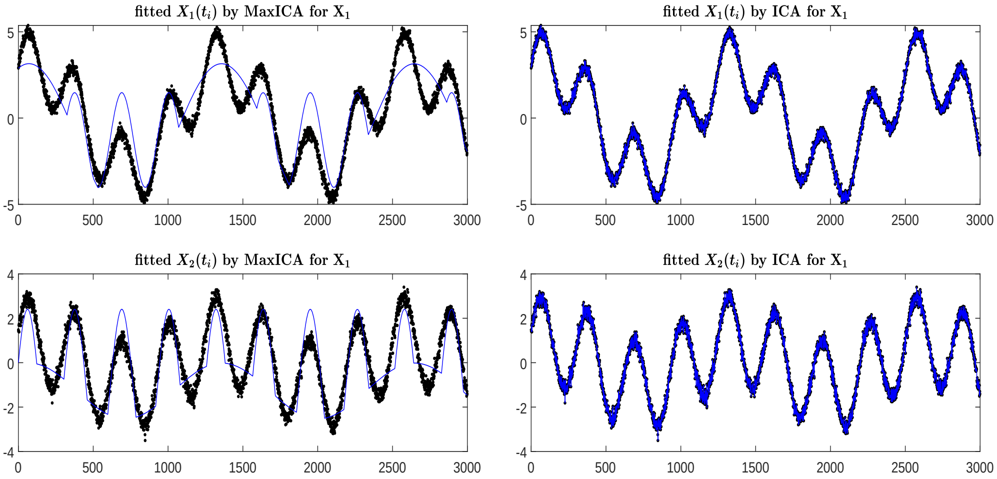


FIG 18. (***Example in Section*** 4.2) *Fitted signals (in blue lines), by* MaxICA *(in left panels) and* ICA *(in right panels), to simulated data signals (in black dots) of* $\mathbf{X}_1$ *in* (4.2).

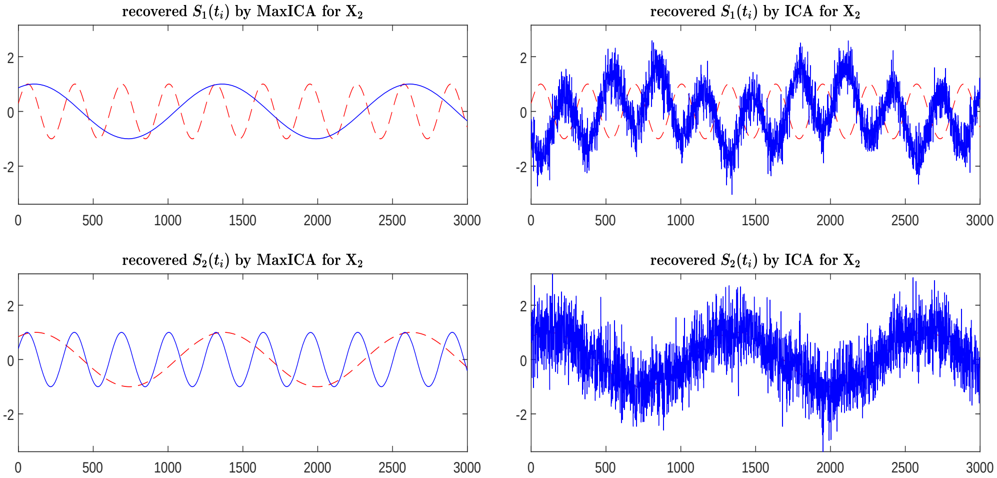


FIG 19. (***Example in Section*** 4.2) *Compare true components (in red dashed lines) and recovered components (in blue lines), by* MaxICA *(in left panels) and* ICA *(in right panels), from simulated data signals of* $\mathbf{X}_2$ *in* (4.3). *See Remark* 1 *for the order of recovered components.*

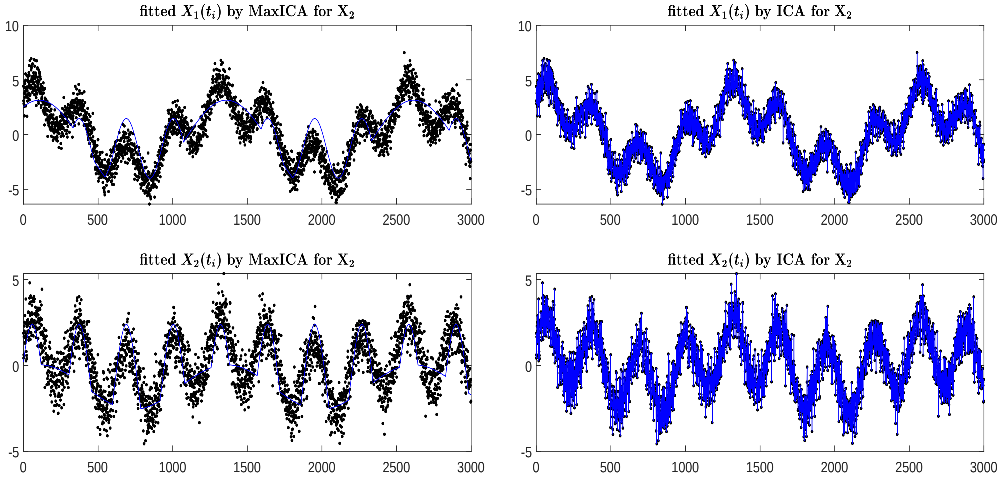


FIG 20. (***Example in Section*** 4.2) *Fitted signals (in blue lines), by* MaxICA *(in left panels) and* ICA *(in right panels), to simulated data signals (in black dots) of* $\mathbf{X}_2$ *in* (4.3).

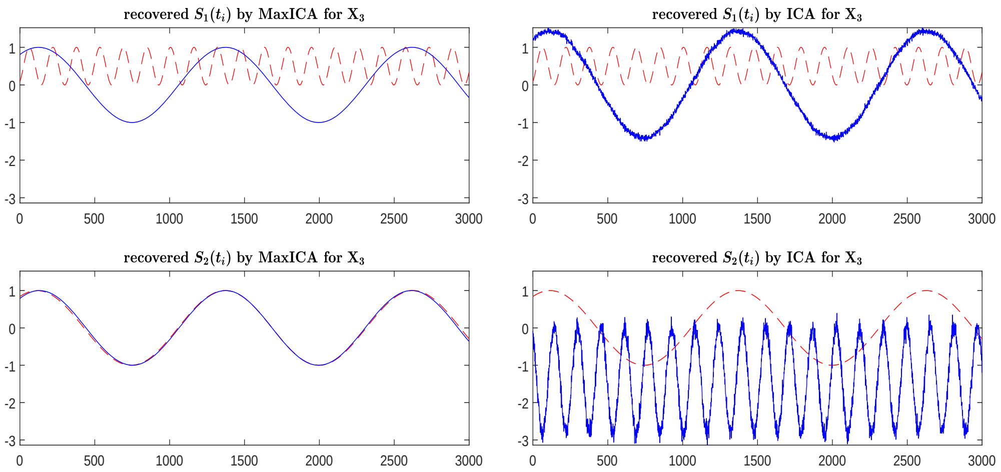


FIG 21. (***Example in Section*** 4.2) *Compare true components (in red dashed lines) and recovered components (in blue lines), by* MaxICA *(in left panels) and* ICA *(in right panels), from simulated data signals of* $\mathbf{X}_3$ *in* (4.4). *See Remark* 1 *for the order of recovered components.*

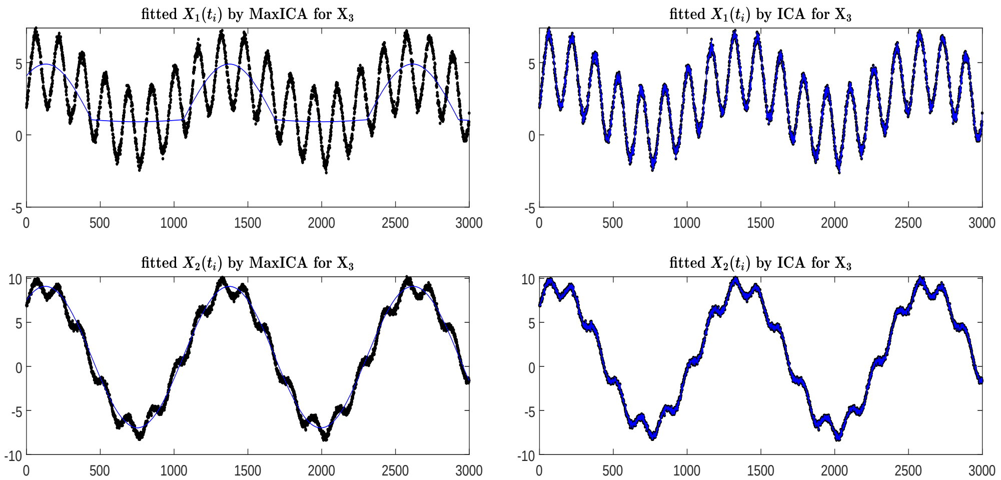
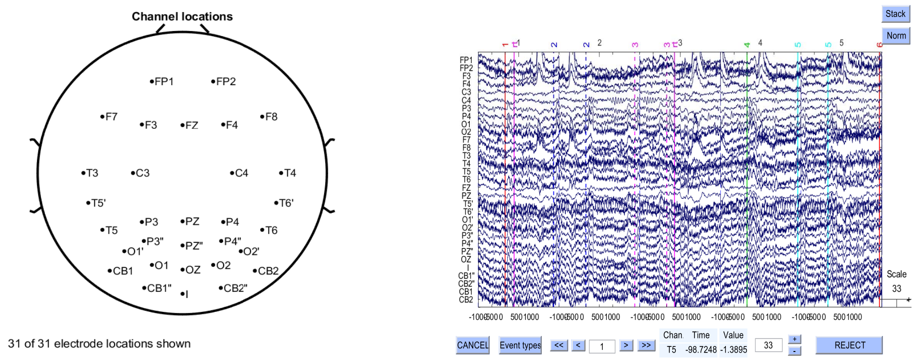


FIG 22. (***Example in Section*** 4.2) *Fitted signals (in blue lines), by* MaxICA *(in left panels) and* ICA *(in right panels), to simulated data signals (in black dots) of* $\mathbf{X}_3$ *in* (4.4).



Chan. Time Value
T5 -98.7248 -1.3895

FIG 23. (***Visual processing data***) *Left: locations of* 31 *channels. Right: data of* 31 *channels.*

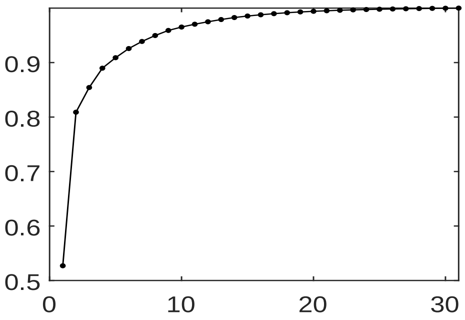


FIG 24. (***Visual processing data***) *Fraction of total variance retained versus the number of eigenvalues of subject* 1*'s categorization task.*

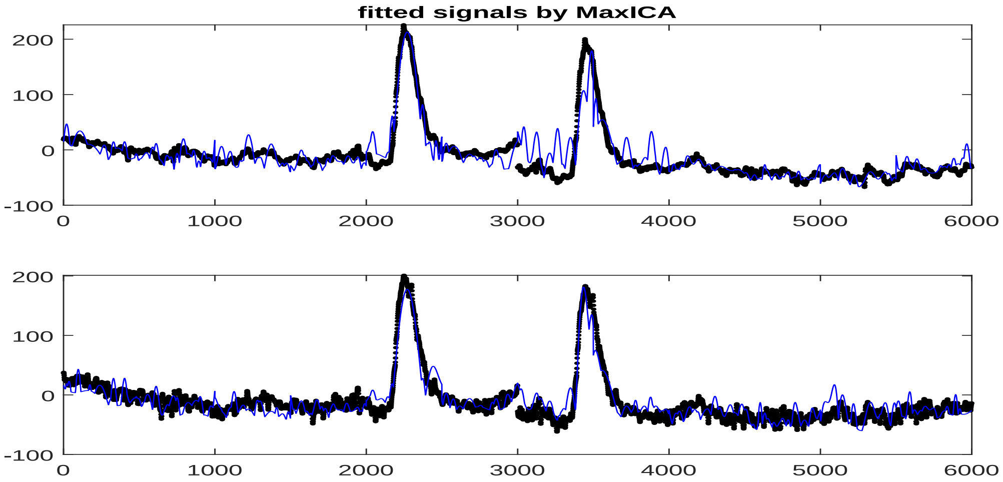


FIG 25. (***Visual processing data***) *Fitted signals* (*in blue lines*), *by* MaxICA, *to observed signals* (*in black dots*) *of subject* 1*'s* **categorization** *task.*

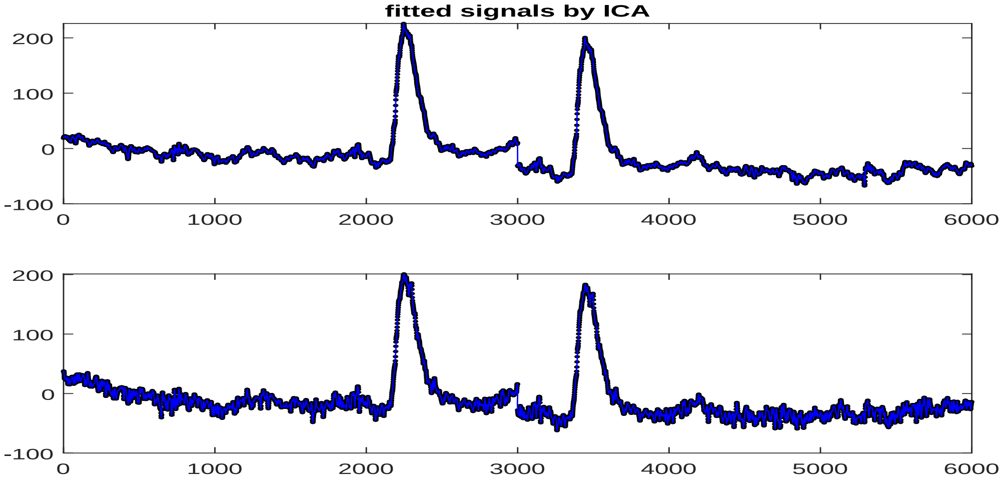


FIG 26. (***Visual processing data***) *Fitted signals* (*in blue lines*), *by* ICA, *to observed signals* (*in black dots*) *of subject* 1*'s* **categorization** *task.*

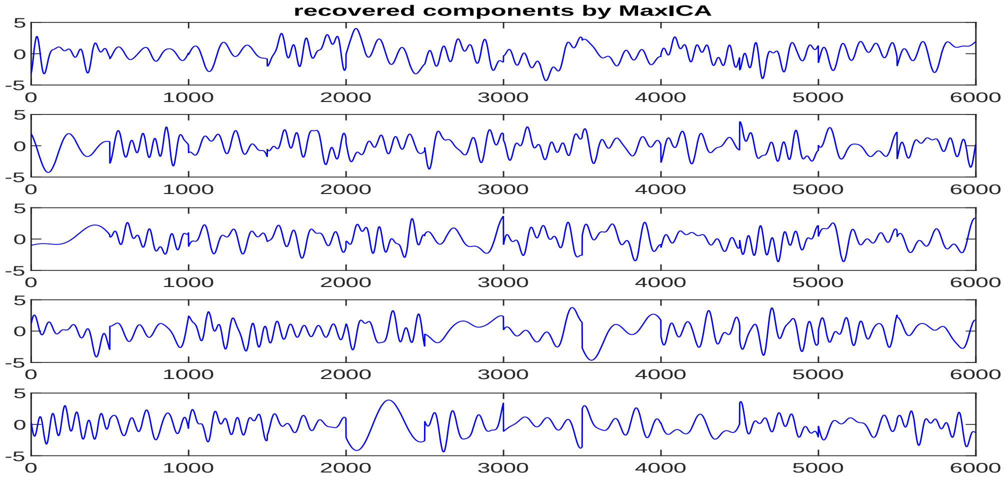


FIG 27. (***Visual processing data***) *Recovered components (in blue lines) by* MaxICA *of subject* 1*'s* ***categorization*** *task.*

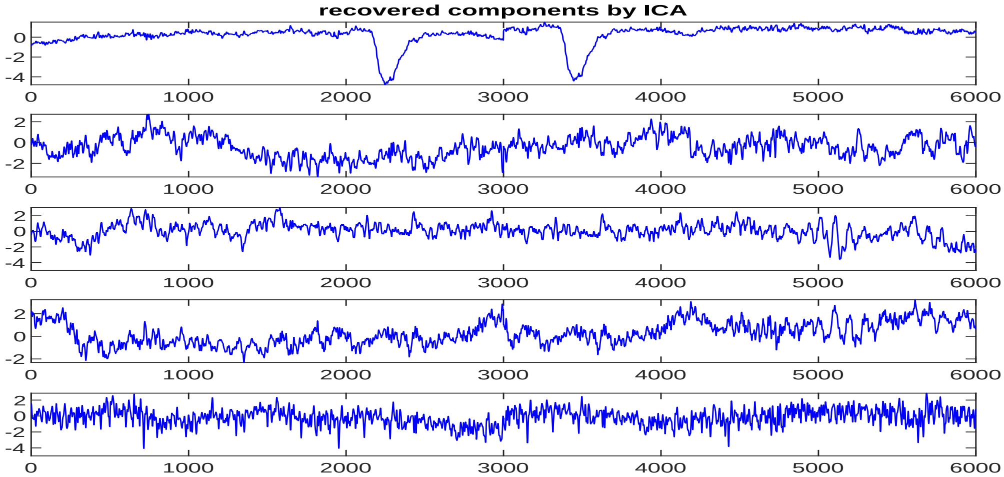


FIG 28. (***Visual processing data***) *Recovered components (in blue lines) by* ICA *of subject* 1*'s* ***categorization*** *task.*

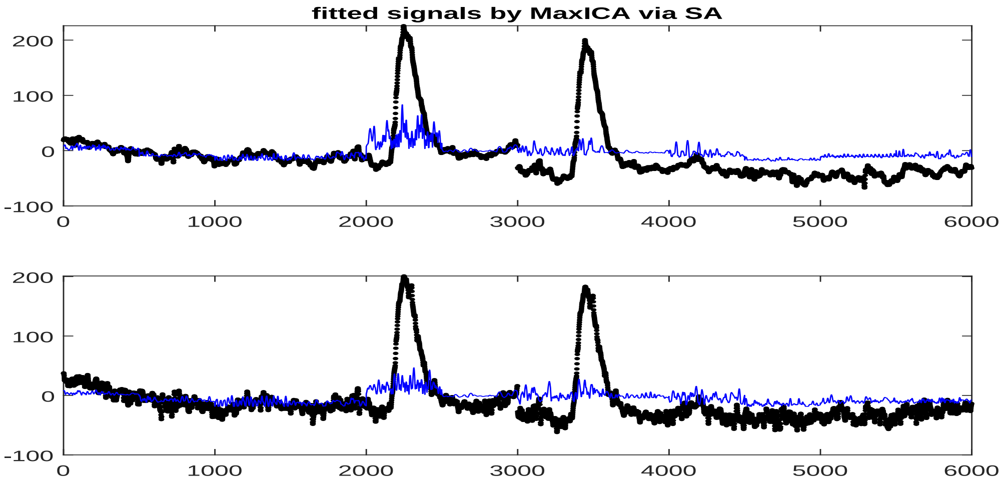


Fig 29. (***Visual processing data***) *The caption is similar to that of Figure* 25*, except that the Simulated Annealing* (SA) *algorithm is used.*

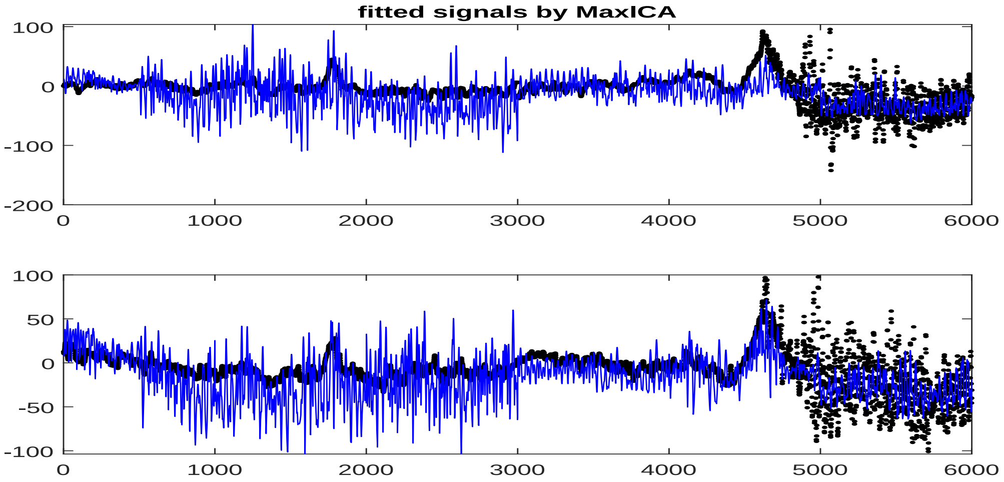


Fig 30. (***Visual processing data***) *Fitted signals* (*in blue lines*), *by* MaxICA, *to observed signals* (*in black dots*) *of subject* 1*'s* ***recognition*** *task.*

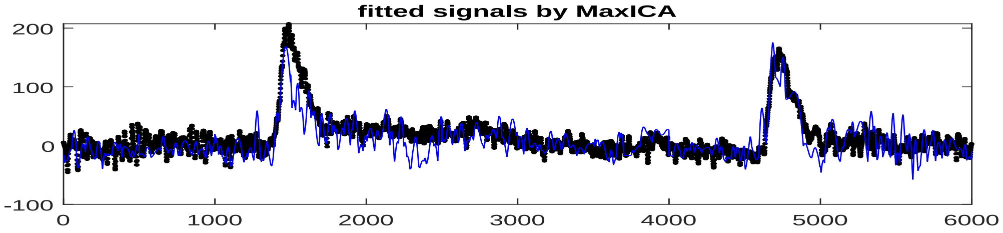


FIG 31. (***Visual processing data***) *Fitted signals* (*in blue lines*), *by* MaxICA, *to observed signals* (*in black dots*) *of subject* 2*'s* ***categorization*** *task.*

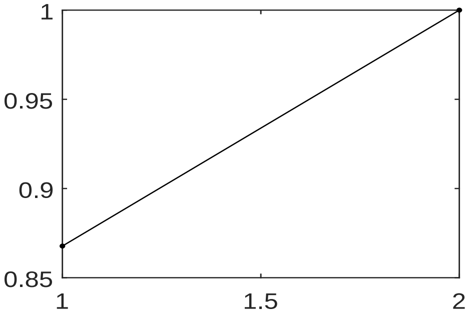


FIG 32. (***Epilepsy data, focal channels***) *Fraction of total variance retained versus the number of eigenvalues.*

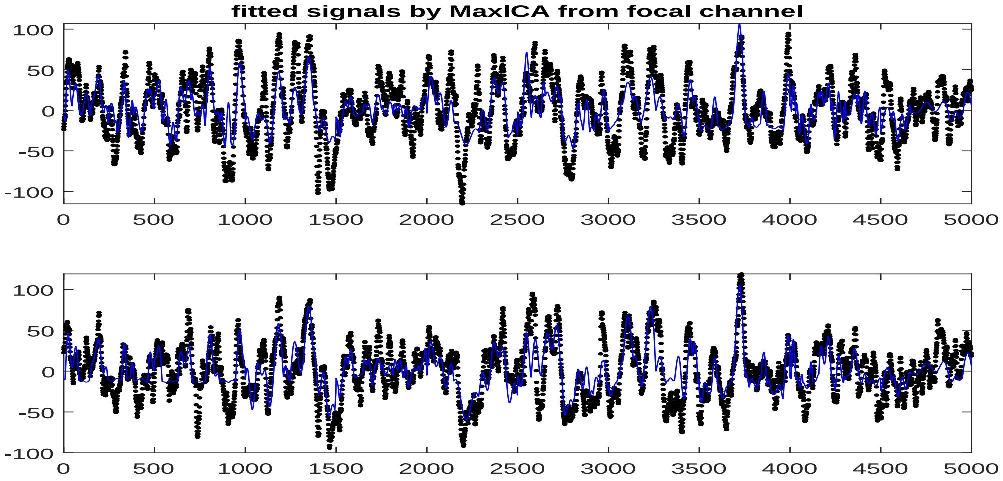


FIG 33. (***Epilepsy data, focal channels***) *Fitted signals* (*in blue lines*), *by* MaxICA, *to observed signals* (*in black dots*) *for one* EEG *signal pair.*

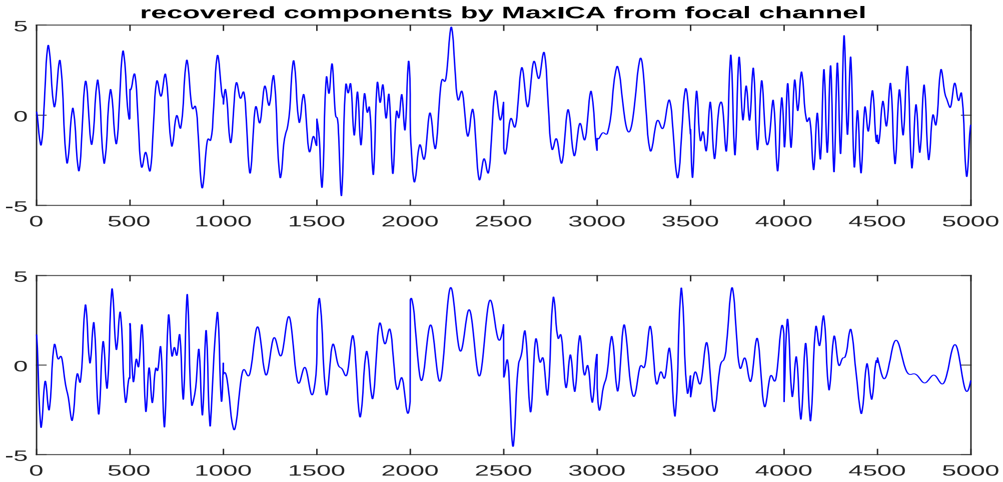


Fig 34. (***Epilepsy data, focal channels***) *Recovered components (in blue lines) by* MaxICA.

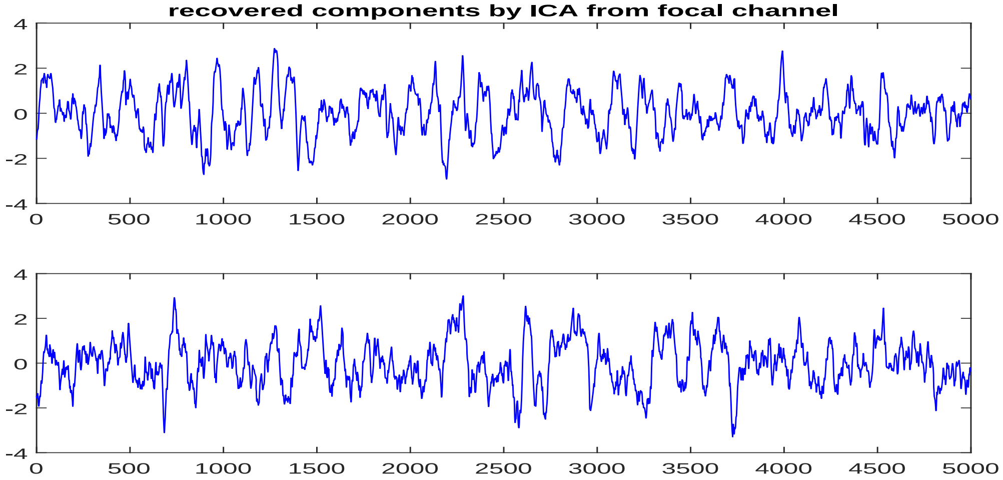


Fig 35. (***Epilepsy data, focal channels***) *Recovered components (in blue lines) by* ICA.

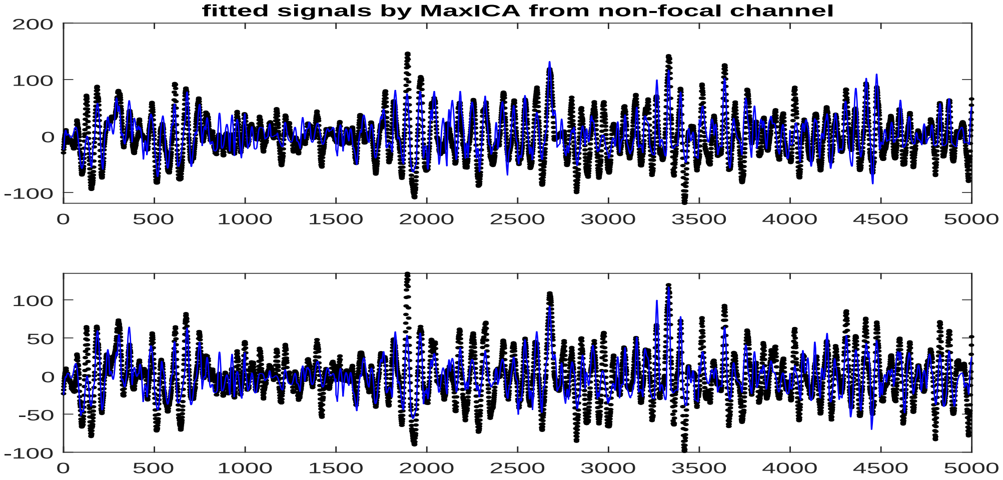


Fig 36. (***Epilepsy data, non-focal channels***) *Fitted signals (in blue lines), by* MaxICA, *to observed signals (in black dots) for one* EEG *signal pair.*

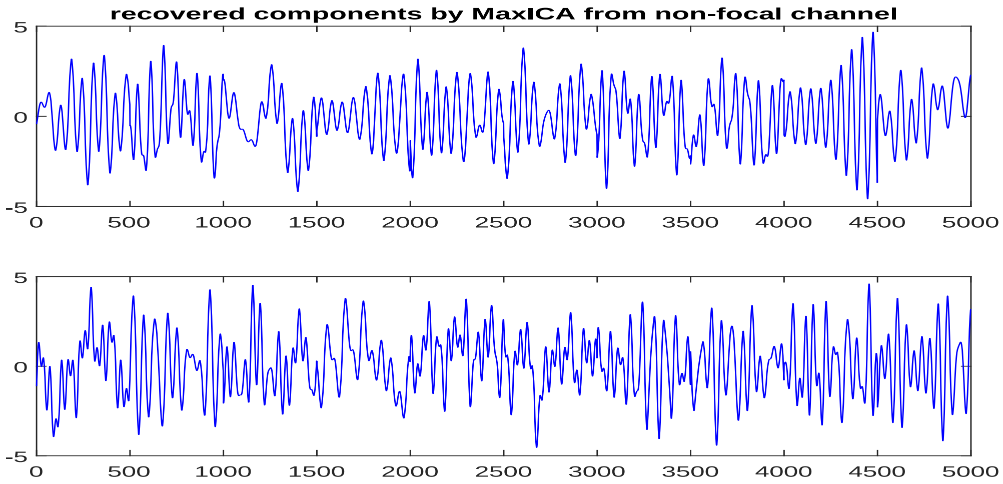


FIG 37. (***Epilepsy data, non-focal channels***) *Recovered components (in blue lines) by* MaxICA.

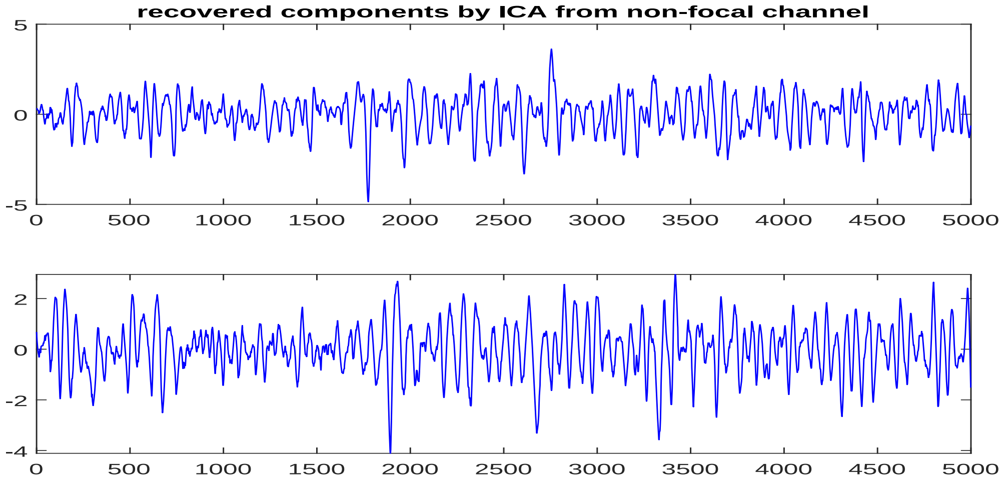


FIG 38. (***Epilepsy data, non-focal channels***) *Recovered components (in blue lines) by* ICA.

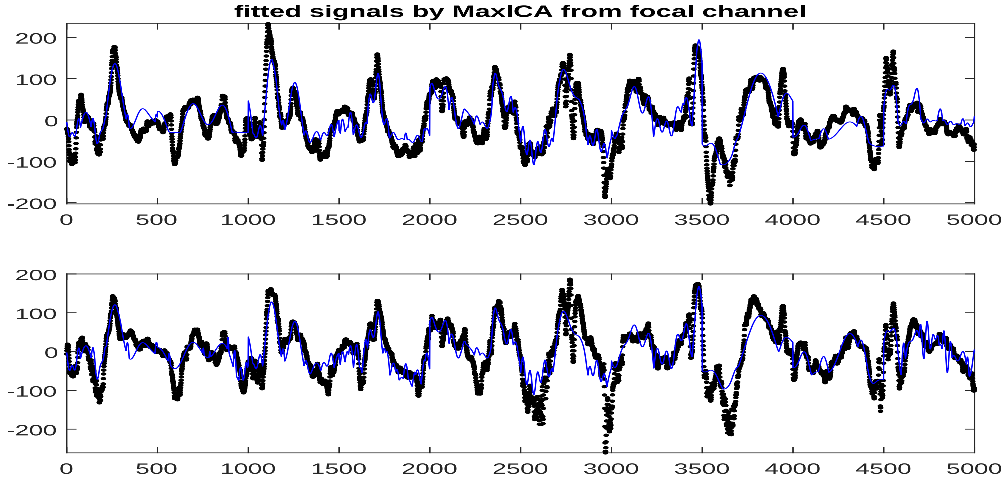


FIG 39. (***Epilepsy data, focal channels***) *Fitted signals (in blue lines), by* MaxICA, *to observed signals (in black dots) for another* EEG *signal pair.*

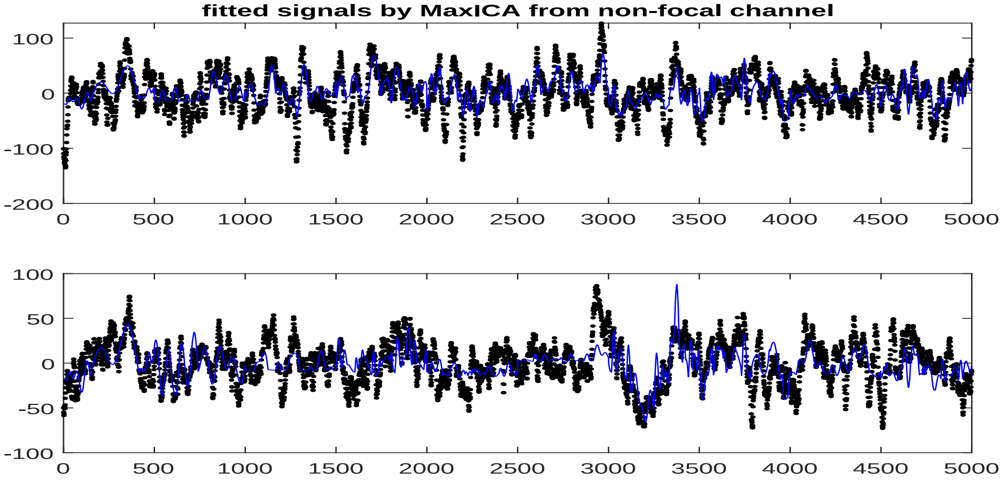


FIG 40. (***Epilepsy data, non-focal channels***) *Fitted signals* (*in blue lines*), *by* MaxICA, *to observed signals* (*in black dots*) *for another* EEG *signal pair.*