MaxICA with application to brain EEG data

Chunming Zhang

Department of Statistics
University of Wisconsin-Madison

Based on joint work with:

Ruosi Guo, and Zhengjun Zhang

Outline

Outline

Background and motivation

- In many scientific fields (neuroscience, genetics, finance, analytical chemistry), non-linear temporal signals: recorded and require advanced analysis tools to extract hidden sources or components, each of which will be associated to a different physical process.

- A commonly encountered but highly under-determined problem: "**blind source separation**", aiming to separate

  $$\text{“\textbf{hidden component signals}”} \quad S_1(t), \dots, S_N(t)$$

  from

  $$\text{“\textbf{observed mixed signals}”} \quad X_1(t), \dots, X_p(t),$$

  with little information about either N, the components or the mixing mechanism.

Examples

- (neuroscience)
  - Remove artifacts, such as eye blinks, from **EEG** data.
  - Isolate both eye movement and eye blinking artifacts, as well as cardiac, myographic, and other artifacts from **MEG** signals.
- (genetics) Analysis of changes in gene expression over time in single cell RNA-sequencing experiments.
- (finance) Finding hidden factors in financial time series data.

Example 1: separation of artifacts in MEG data

MEG signals: recorded, 122 channels (at 61 locations).

$$\{X_j(t_i) : i = 1, \ldots, m\}, \quad j = 1, \ldots, 122. \tag{1}$$

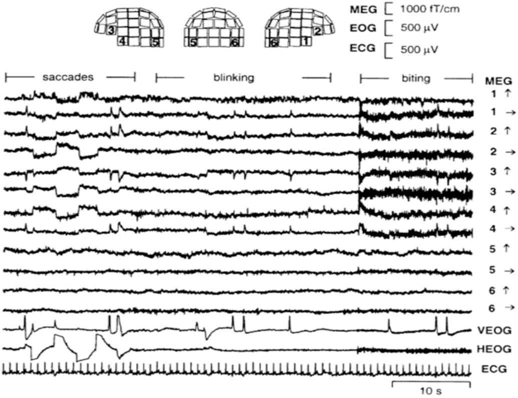


Fig. 11. (From Vigário et al., 1998a.) Samples of MEG signals, showing artifacts produced by blinking, saccades, biting and cardiac cycle. For each of the six positions shown, the two orthogonal directions of the sensors are plotted. Reprinted with permission from the MIT Press.

Plot 9 IC**s**: found from the recorded data,

- IC1, IC2: due to the muscular activity originated from the biting,
- IC3: horizontal eye movements,
- IC5: eye blinks,
- IC4: cardiac artifact,
- IC8: artifact originated at the digital watch clearly, located to the right side of the magnetometer,
- IC9: related to a sensor presenting higher RMS (root mean squared) noise than the others.

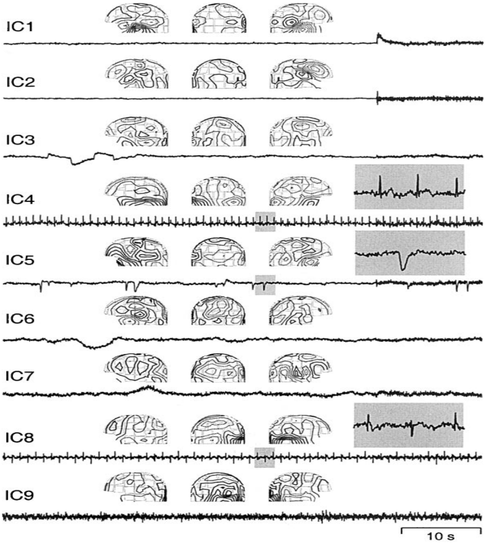


Fig. 12. (From Vigário et al., 1998a.) Nine independent components found from the MEG data. For each component the left, back and right views of the field patterns generated by these components are shown—full line stands for magnetic flux coming out from the head, and dotted line the flux inwards. Reprinted by permission from the MIT Press.

How to recover original (source) signals?

Independent Component Analysis (ICA) (Comon 1994):

- a popular approach with many applications in data analysis, source separation, and feature extraction.
- **FastICA algorithm** (Hyvärinen & Oja 2000): computationally highly efficient for **ICA** estimation.

Some details of ICA

ICA model for an observed signal $\boldsymbol{X}(t)$ at a time point t

$$\boldsymbol{X}(t) = \mathrm{A}\,\boldsymbol{S}(t) \quad \text{i.e.,} \quad \begin{array}{ccc} X_1(t) & = & a_{1,1}\,S_1(t) + \cdots + a_{1,N}\,S_N(t), \\ \cdots & \cdots & \cdots \\ X_j(t) & = & a_{j,1}\,S_1(t) + \cdots + a_{j,N}\,S_N(t), \\ \cdots & \cdots & \cdots \\ X_p(t) & = & a_{p,1}\,S_1(t) + \cdots + a_{p,N}\,S_N(t), \end{array} \tag{2}$$

where components $S_1(t), \ldots, S_N(t)$ and coefficients $\{a_{j,k}\}$ are unknown.

ICA model for observed signals $\boldsymbol{X}(t_1), \ldots, \boldsymbol{X}(t_m)$ at time points $t_1, \ldots, t_m$:

$$\begin{aligned} (\boldsymbol{X}(t_1), \ldots, \boldsymbol{X}(t_m)) &= \mathrm{A}\,(\boldsymbol{S}(t_1), \ldots, \boldsymbol{S}(t_m)), \\ \mathbf{X} &= \mathrm{A}\,\mathbf{S}. \end{aligned} \tag{3}$$

If $p = N$: i.e., # of “data sequences” = # of “source components”,

$$\mathbf{S} = \mathrm{A}^{-1}\,\mathbf{X} = \mathrm{W}\,\mathbf{X}.$$

Limitation of ICA:

- Assume that components $S_1(t), \ldots, S_N(t)$ are **mutually independent** and **jointly non-Gaussian**.
- Assume that components $S_1(t), \ldots, S_N(t)$ are **linearly** combined. Linearity assumption is not realistic in some cases, and is not exactly true in practice (Hyvärinen *et al.* 2000).
- In general, ICA can **not uniquely identify** either correct ordering or proper scaling of components $S_1(t), \ldots, S_N(t)$.

Outline

Motivating example for **MaxICA**

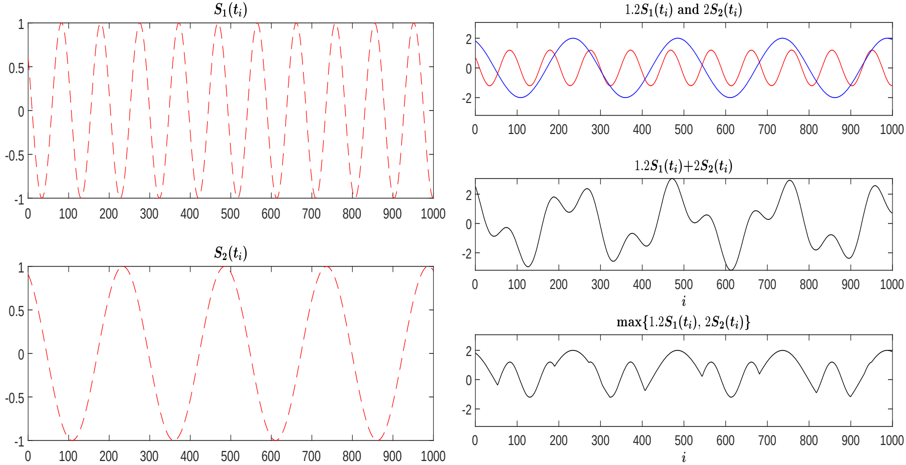


Figure 1: *Left panels: true components $S_1(t_i)$ and $S_2(t_i)$. Right panels: differences between the "sum" combination and the "maximum" combination of 2 sequences of source signals.*

How to recover components $S_1(t_i)$ and $S_2(t_i)$?

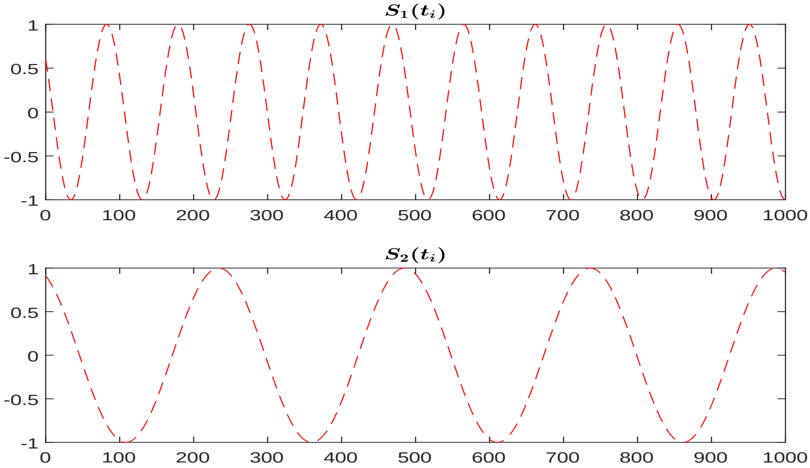


Figure 2: *True components $S_1(t_i)$ and $S_2(t_i)$ in Figure* 1.

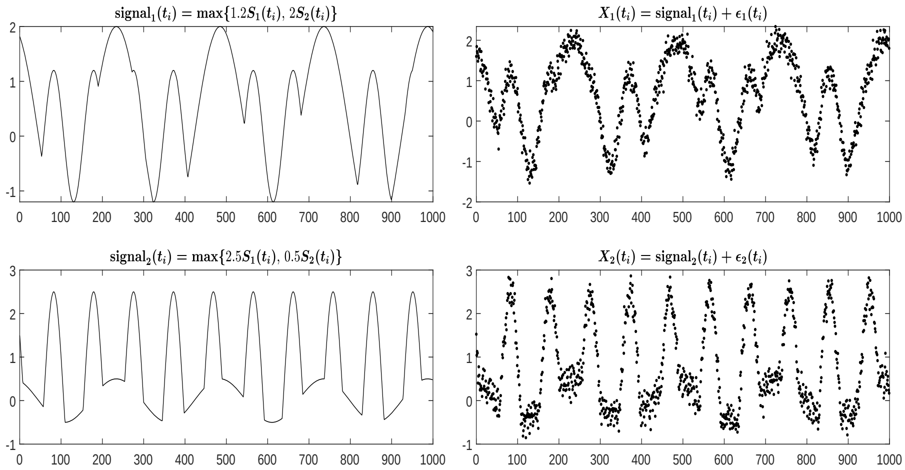


Figure 3: *Left panels: true signals* $\mathrm{signal}_j(t)$*. Right panels: observed signals* $X_j(t) = \mathrm{signal}_j(t) + \epsilon_j(t_i)$*.*

Using $X_1(t_i)$ to recover components:

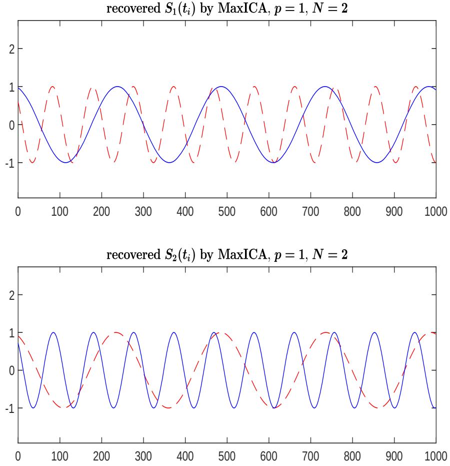


Figure 4: *Compare true components (in red dashed lines) and recovered components (in blue lines) by* **MaxICA** *from observed signals $X_1(t_i)$ in Figure* 3.

Using $X_1(t_i)$ and $X_2(t_i)$ to recover components:

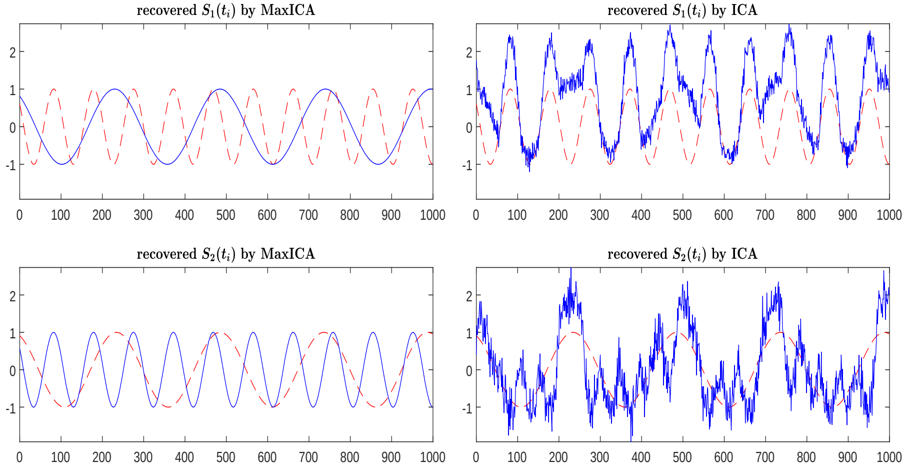


Figure 5: *Compare true components (in red dashed lines) and recovered components (in blue lines) from observed signals $X_1(t_i)$ and $X_2(t_i)$ in Figure 3. Left panels: by* **MaxICA**. *Right panels: by* **ICA**.

Using $X_1(t_i)$ and $X_2(t_i)$ to fit signals:

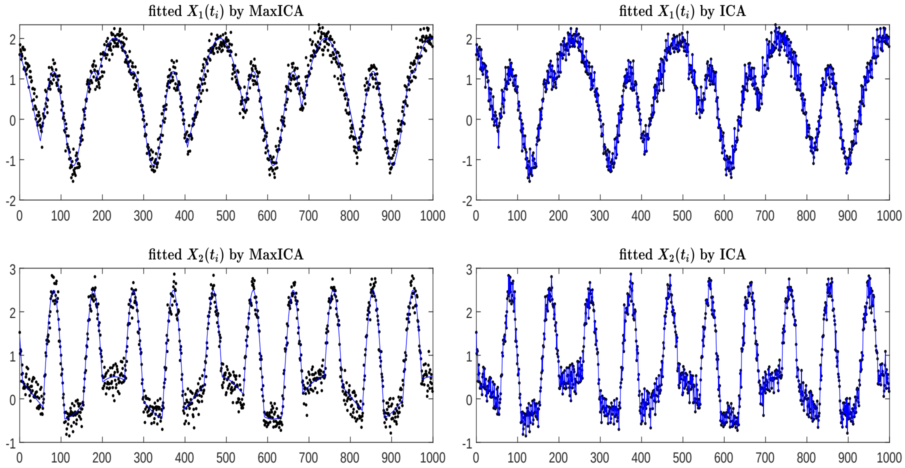


Figure 6: *Fitted signals (in blue lines) to observed signals (in dots .)* $X_1(t_i)$ *and* $X_2(t_i)$ *in Figure* 3*. Left panels: by* **MaxICA***. Right panels: by* **ICA***.*

Proposed MaxICA model for an observed signal $\boldsymbol{X}(t)$ at a time point t

$$
\begin{aligned}
X_1(t) &= \max\{S_{1,1}(t), \ldots, S_{1,N}(t)\} + \epsilon_1(t), \\
\cdots &\quad \cdots \quad \cdots \\
X_j(t) &= \max\{S_{j,1}(t), \ldots, S_{j,N}(t)\} + \epsilon_j(t), \\
\cdots &\quad \cdots \quad \cdots \\
X_p(t) &= \max\{S_{p,1}(t), \ldots, S_{p,N}(t)\} + \epsilon_p(t),
\end{aligned}
\tag{4}
$$

where

$$
S_{j,k}(t) = \sum_{\ell=1}^{n_k} b_{j,k,\ell} \sin(\alpha_{k,\ell}\, t + \beta_{k,\ell}) \tag{5}
$$

$$
\overset{b_{j,k,\ell} \equiv b_{j,k} \mathrm{a}_{k,\ell}}{=\!=\!=\!=\!=\!=} \quad b_{j,k} \sum_{\ell=1}^{n_k} \mathrm{a}_{k,\ell} \sin(\alpha_{k,\ell}\, t + \beta_{k,\ell})
$$

$$
\overset{\mathrm{a}_{k,\ell} \equiv 1}{=\!=\!=} \quad b_{j,k} \sum_{\ell=1}^{n_k} \sin(\alpha_{k,\ell}\, t + \beta_{k,\ell}) \equiv b_{j,k}\, S_k(t). \tag{6}
$$

MaxICA model (4) re-written via (6)

$$\begin{array}{rcl} X_1(t) & = & \max\{b_{1,1}\, S_1(t), \ldots, b_{1,N}\, S_N(t)\} + \epsilon_1(t), \\ \cdots & \cdots & \cdots \\ X_j(t) & = & \max\{b_{j,1}\, S_1(t), \ldots, b_{j,N}\, S_N(t)\} + \epsilon_j(t), \\ \cdots & \cdots & \cdots \\ X_p(t) & = & \max\{b_{p,1}\, S_1(t), \ldots, b_{p,N}\, S_N(t)\} + \epsilon_p(t), \end{array} \tag{7}$$

where $\mathrm{E}\{\epsilon_j(t)\} = 0$, $j = 1, \ldots, p$.

- $b_{j,k,\ell}$ in (5): the ℓth sine waveform, within the kth source component, in the jth data sequence, for $j \in \{1, \ldots, p\}$, $k \in \{1, \ldots, N\}$, $\ell \in \{1, \ldots, n_k\}$;
- N, $\{n_k\}$, $\{b_{j,k,\ell}\}$, $\{\alpha_{k,\ell}\}$, $\{\beta_{k,\ell}\}$: unknown.
- Number of coefficients:
  - using (5): $p(n_1 + \cdots + n_N) + 2(n_1 + \cdots + n_N)$;
  - using (6): $p\, N + 2(n_1 + \cdots + n_N)$.

Loss function for parameter estimation

The jth sequence of observed signals: $\vec{\boldsymbol{X}}_j = (X_j(t_1), \ldots, X_j(t_m))^T$
The jth sequence of modeled signals:

$$\vec{\boldsymbol{f}}_j(\boldsymbol{\theta}) \quad = \quad (f_j(\boldsymbol{\theta}, t_1), \ldots, f_j(\boldsymbol{\theta}, t_m))^T, \tag{8}$$

where

$$f_j(\boldsymbol{\theta}, t) \quad \overset{(7)}{=} \quad \max\{b_{j,1}\, S_1(t), \ldots, b_{j,N}\, S_N(t)\} + \gamma,$$

- γ: location parameter (to adjust the horizontal location of the data),
- $\boldsymbol{\theta} \overset{(6)}{=} \Big\{\{(\alpha_{k,\ell}, \beta_{k,\ell}) : \ell = 1, \ldots, n_k\}_{k=1}^{N}; \{b_{j,k} : k = 1, \ldots, N\}_{j=1}^{p}; \gamma\Big\}$.

Loss function:

$$L(\boldsymbol{\theta}) \quad = \quad \sum_{j=1}^{p} \|\vec{\boldsymbol{X}}_j - \vec{\boldsymbol{f}}_j(\boldsymbol{\theta})\|_2, \tag{9}$$

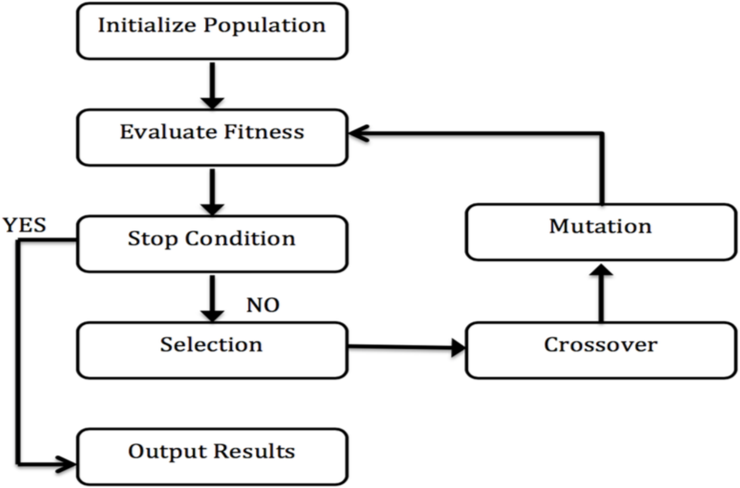
Estimation of θ: minimize $L(\theta)$ w.r.t. θ. **How to solve?**

Outline

Outline

- "Genetic Algorithms" (**GA**s): stochastic search methods for optimization, based on principles of Darwinian evolution and genetics, aiming to provide the optimal solution of a problem.
- Books:
  - "*Adaptation in Natural and Artificial Systems*" (Holland 1975): built the first **GA**s on how to apply the principles of natural evolutions to optimization problems.
  - "*Genetic Algorithms in Search, Optimization and Machine Learning*" (Goldberg 1989): helped **GA** become increasingly powerful for solving search and optimization problems.
- **GA**s evolve a population of candidate solutions, called "chromosomes", through generations to the problem, using **selection**, **crossover**, and **mutation** operators.

Flowchart of GA: with "Selection/Reproduction", "Crossover", "Mutation" operators.



Outline

1. Introduction
2. Proposed **MaxICA** model
3. Proposed parameter learning algorithm for **MaxICA** model
   - Review of classical “Genetic Algorithm” (**GA**)
   - Proposed **ERD_GA** algorithm with 3 operators
4. Simulation studies
   - Example 1
   - Example 2
   - Example 3
5. Real **EEG** data analysis
   - Visual processing data
   - Epilepsy data
6. Discussion

I. "Elite Weighted Sum selection" (EWSS) operator for MaxICA

Procedure:

- Compute the loss function $L_s = L(\widehat{\theta}^{(s)})$ in (9) from the solution $\widehat{\theta}^{(s)}$,

$$L_1 \leq L_2 \leq \cdots \leq L_n, \tag{10}$$

and compute

$$d_j = L_{j+1} - L_j, \quad j = 1, \ldots, n-1, \tag{11}$$

$$L_1 \xrightarrow{d_1} L_2 \xrightarrow{d_2} \cdots \xrightarrow{d_{n-2}} L_{n-1} \xrightarrow{d_{n-1}} L_n.$$

- Define cumulative probabilities,

$$p_0 = 0, \quad p_1 = \frac{\sum_{j=1}^{n-1} d_j}{D}, \quad p_2 = \frac{\sum_{j=1}^{n-1} d_j + \sum_{j=2}^{n-1} d_j}{D}, \quad \ldots, \quad p_{n-1} = 1, \tag{12}$$

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

- Generate a random number $r \in (0, 1]$. If $p_{s-1} < r \leq p_s$, then select the sth solution, $s = 1, \ldots, n-1$. (Drop the nth solution with the largest L_n.)

Advantages of EWSS:

- (compared with “Roulette Wheel Selection”) Guarantee all solutions, excluding the worst one, can have some chance to be selected.
- (compared with “Rank Selection”) Chances for solutions to be selected are not identical; better solutions have higher chances to be selected.
- Solutions having similar fitness values will have similar chances to be selected.
- For optimization problems, EWSS method can be more efficient.

Example:

losses L_s:	12.5,	210,	220,	310,	375,
solutions s:	1,	2,	3,	4,	5.

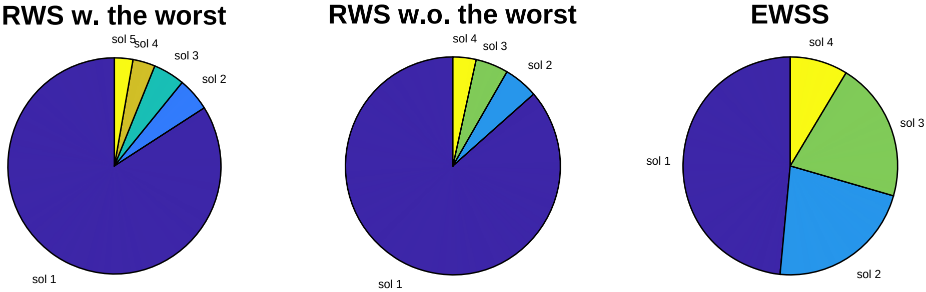
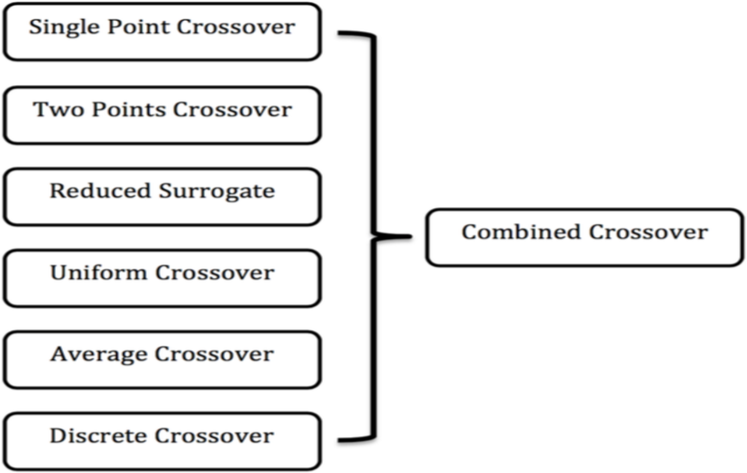


Figure 7: **(Comparison of Selection Methods)** *The area of each solution is proportional to the selection probability:* $\{1/L_s : s = 1, \ldots, 5\}$ *for the left;* $\{1/L_s : s = 1, \ldots, 4\}$ *for the middle;* $\{(L_n - L_s) : s = 1, \ldots, 4\}$ *for the right.*

II. "Random" Combined crossover operator (CCO) for MaxICA

In each iteration of the algorithm, **randomly** apply one of the crossover methods.



Comparison of our "random" CCO and Hassan (2015)'s "uniform" CCO:

- "Uniform" CCO: applies **4** crossover operators ("Heuristic crossover", "Arithmetic crossover", "Simulated binary crossover", "Linear BGA crossover") at each generation, but chooses the one with the best performance.
- Our "random" CCO: **randomly** chooses one of **6** crossover operators at each generation.

III. "Dynamic mutation" operator for MaxICA

- Mutation rate changes automatically while the algorithm gets stuck.
  - While no better solutions are found for a number of iterations, the mutation rate will decrease until it reaches the lower bound.
  - While the algorithm converges fast, the mutation rate will increase until it reaches the upper bound.

Outline

Methods for comparison

model	algorithm
MaxICA	**ERD_GA**
	classical_GA
	SA
ICA	FastICA
	Infomax **ICA**

Outline

Example 1: $\{X_j(t) : j = 1, \ldots, 5\}$ follow the $\mathbf{MaxICA}$ model (7), with

-
$$\begin{aligned} S_1(t) &= \sin(1.5\,t + 2) + \sin(0.3\,t + 1), \\ S_2(t) &= \sin(0.7\,t + 1.5) + \sin(2\,t + 2.5), \\ S_3(t) &= \sin(3\,t + 2) + \sin(t + 0.1), \\ S_4(t) &= \sin(2\,t + 1.5) + \sin(1.2\,t + 0.6), \\ S_5(t) &= \sin(t + 0.4) + \sin(3\,t + 2), \end{aligned} \tag{13}$$

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

  with $\{b_{j,k}\} \overset{\text{i.i.d.}}{\sim} \text{Uniform}[0, 5]$.
- $\{\epsilon_j(t) : j = 1, \ldots, 5\} \overset{\text{i.i.d.}}{\sim} \mathbb{N}(0, 0.5949^2)$.

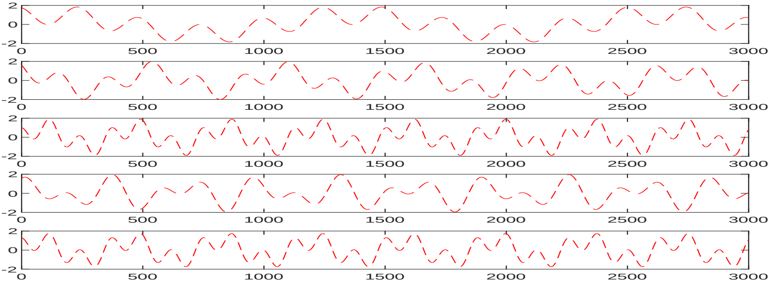


Figure 8: (**Example** 1) *True components* $S_1(t), \ldots, S_5(t)$ *in* (13).

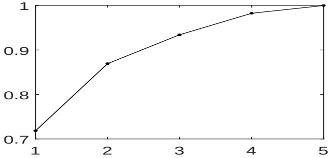


Figure 9: (**Example** 1) *Fraction of total variance retained versus the number of eigenvalues.*

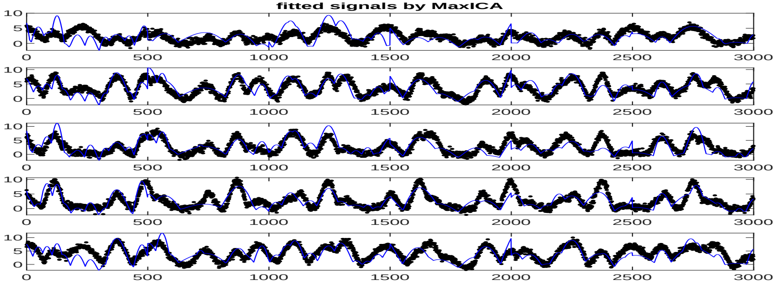


Figure 10: (**Example** 1) *Fitted signals* (*in blue lines*)*, by* **MaxICA** *assuming* 5 *hidden components in the* **MaxICA** *model, to observed signals* (*in dots* **.**)*.*

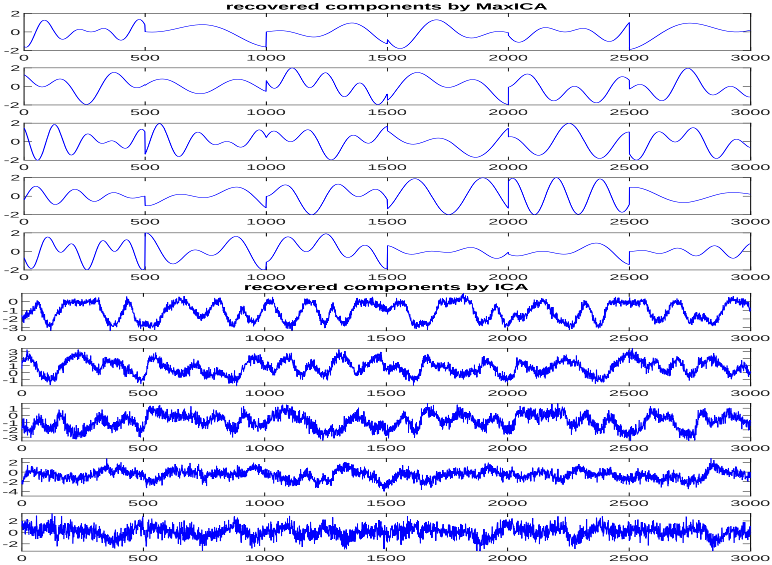


Figure 11: (**Example** 1) *Recovered components* (*in blue lines*) *in the* **MaxICA** *model. Top: by* **MaxICA**. *Bottom: by* **ICA**.

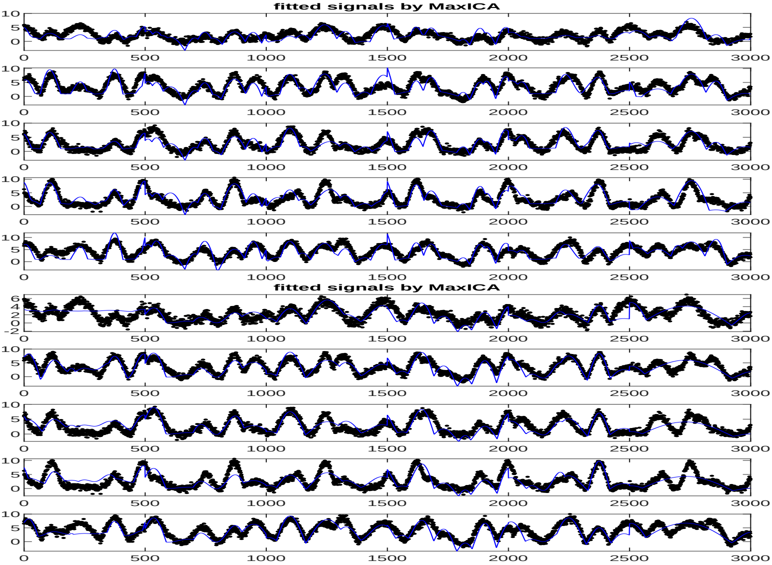


Figure 12: (**Example** 1) *Fitted signals* (*in blue lines*)*, by* **MaxICA** *assuming* 4 (*top*) *and* 2 (*bottom*) *hidden components in the* **MaxICA** *model, to observed signals* (*in dots* **.**).

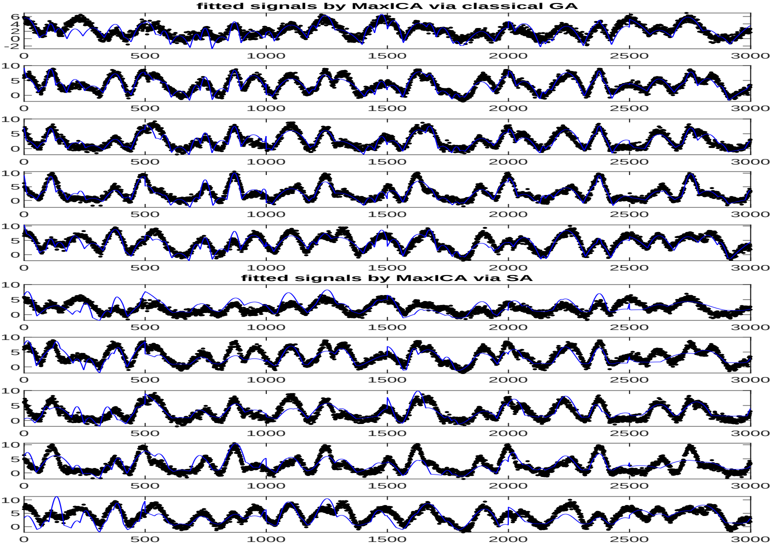


Figure 13: (**Example** 1) *Fitted signals* (*in blue lines*)*, via the* **classical_GA** (*top*) *and* **SA** (*bottom*) *algorithms, by* **MaxICA** *assuming* 5 *hidden components in the* **MaxICA** *model, to observed signals* (*in dots.*)

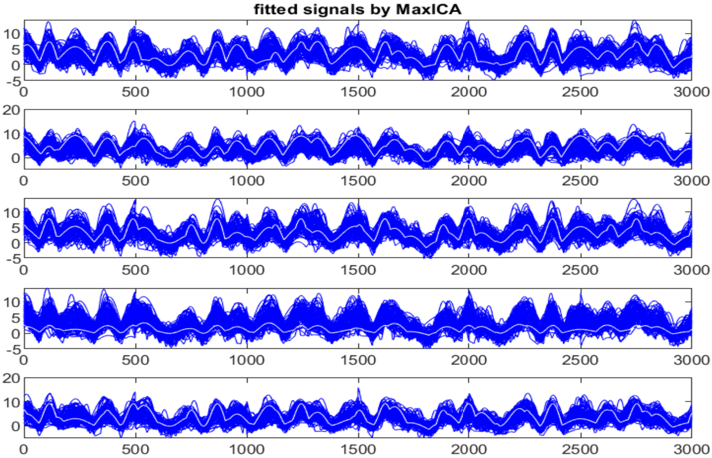


Figure 14: (**Example** 1) *Fitted signals* (*in* blue *lines*), *by* **MaxICA** *assuming* 5 *hidden components in the* **MaxICA** *model, replicated for* 100 *sets of simulated data signals, where white lines are for true signals.*

Outline

Example 2: $\{X_j(t) : j = 1, 2\}$ in data matrices $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ from noisy $\mathbf{ICA}$ models,

$$\begin{aligned}
\mathbf{X}_1 &= \mathrm{A}_1\, \mathbf{S}_1 + \epsilon_1; & (14)\\
\mathbf{X}_2 &= \mathrm{A}_1\, \mathbf{S}_1 + \epsilon_2; & (15)\\
\mathbf{X}_3 &= \mathrm{A}_2\, \mathbf{S}_2 + \epsilon_1, & (16)
\end{aligned}$$

- $$\begin{aligned}
&\text{for } \mathbf{S}_1: \quad S_1(t) = \sin(2\,t + 0.3), \quad S_2(t) = \sin(0.5\,t + 1); & (17)\\
&\text{for } \mathbf{S}_2: \quad S_1(t) = \sin^2(2\,t + 0.3), \quad S_2(t) = \sin(0.5\,t + 1),
\end{aligned}$$

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

- entries in $\epsilon_1 \overset{\text{i.i.d.}}{\sim} 0.2\,\mathbb{N}(0, 1)$;
  entries in $\epsilon_2 \overset{\text{i.i.d.}}{\sim} 0.8\,\mathbb{N}(0, 1)$.

For data matrix $\mathbf{X}_1$:

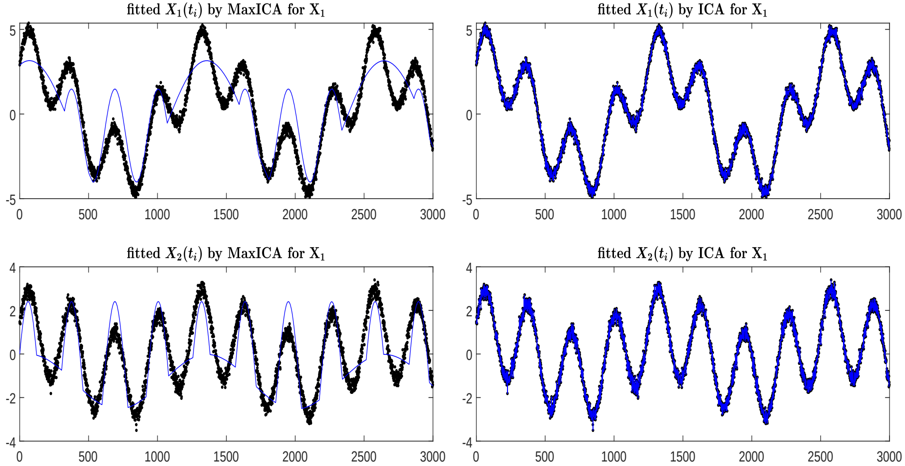


Figure 15: (**Example** 2) *Fitted signals* (*in blue lines*)*, by* **MaxICA** (*in left panels*) *and* **ICA** (*in right panels*)*, to simulated data signals* (*in dots* **.**) *of* $\mathbf{X}_1$*.*

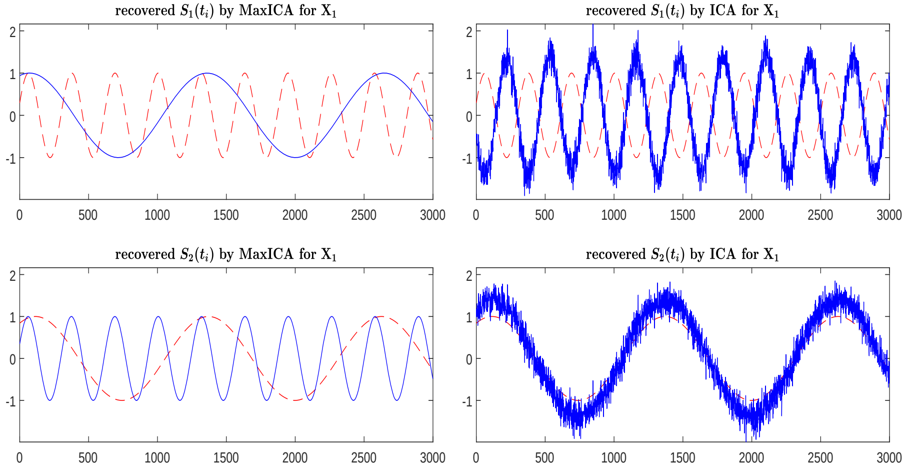


Figure 16: (**Example** 2) *Compare true components* (*in red dashed lines*) *and recovered components* (*in blue lines*), *by* **MaxICA** (*in left panels*) *and* **ICA** (*in right panels*), *from simulated data signals of* $\mathbf{X}_1$.

For data matrix $\mathbf{X}_2$:

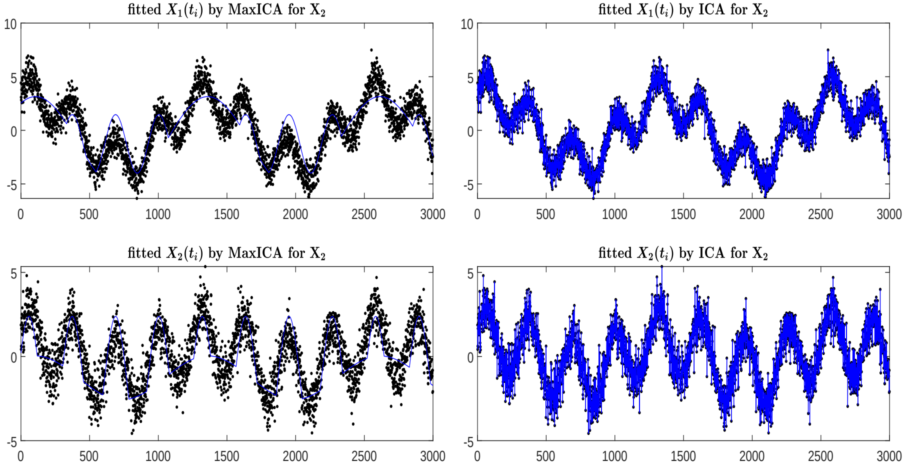


Figure 17: (**Example** 2) *Fitted signals (in blue lines), by* **MaxICA** *(in left panels) and* **ICA** *(in right panels), to simulated data signals (in dots* **.** *) of* $\mathbf{X}_2$.

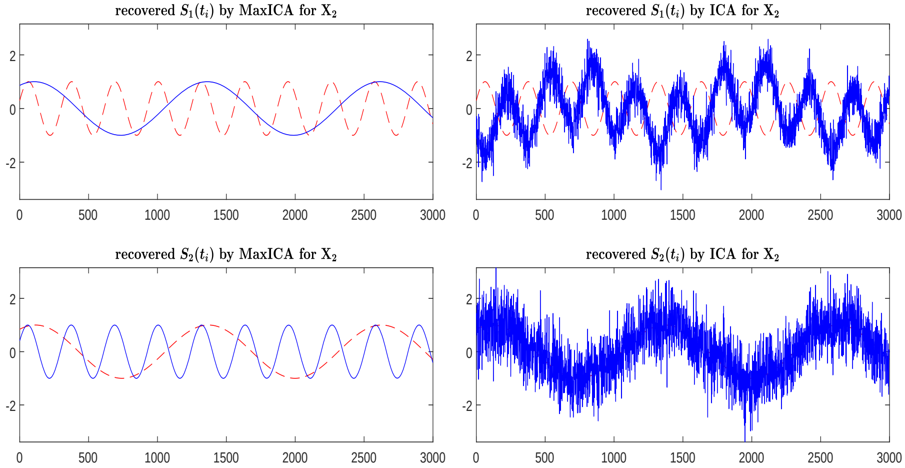


Figure 18: (**Example** 2) *Compare true components* (*in red dashed lines*) *and recovered components* (*in blue lines*), *by* **MaxICA** (*in left panels*) *and* **ICA** (*in right panels*), *from simulated data signals of* $\mathbf{X}_2$.

For data matrix $\mathbf{X}_3$:

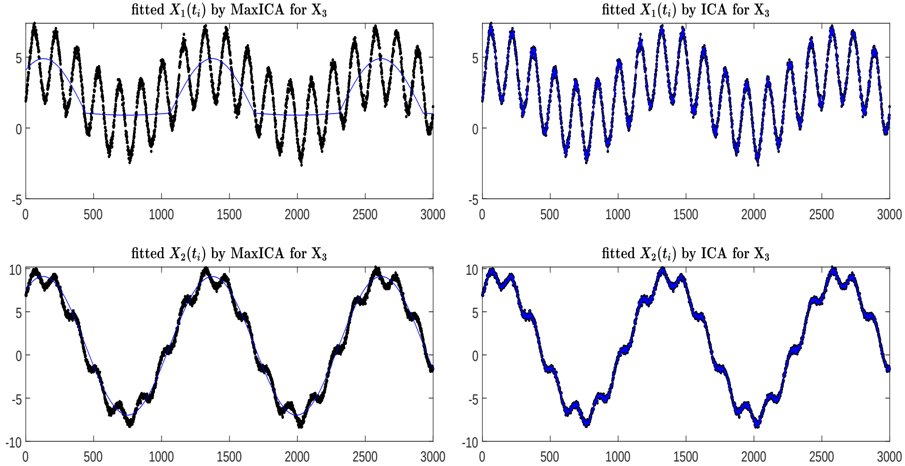


Figure 19: (**Example** 2) *Fitted signals* (*in blue lines*), *by* **MaxICA** (*in left panels*) *and* **ICA** (*in right panels*), *to simulated data signals* (*in dots* **.**) *of* $\mathbf{X}_3$.

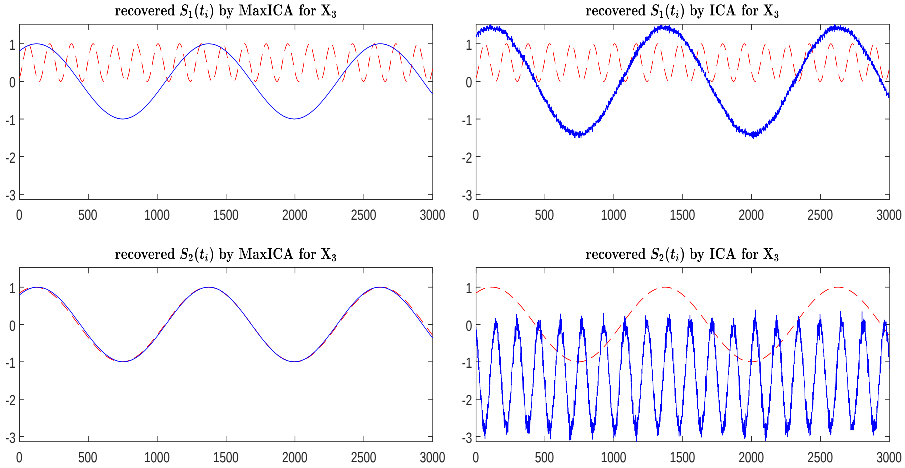


Figure 20: (**Example** 2) *Compare true components* (*in red dashed lines*) *and recovered components* (*in blue lines*), *by* **MaxICA** (*in left panels*) *and* **ICA** (*in right panels*), *from simulated data signals of* $\mathbf{X}_3$.

Outline

Example 3:

- Generate 5 components $S_k(t)$. Using linear and max-linear combinations to obtain 5 data signals $X_j(t)$, which are assigned to 5 specific channel locations: FP1, FP2, F3, F4, C3.

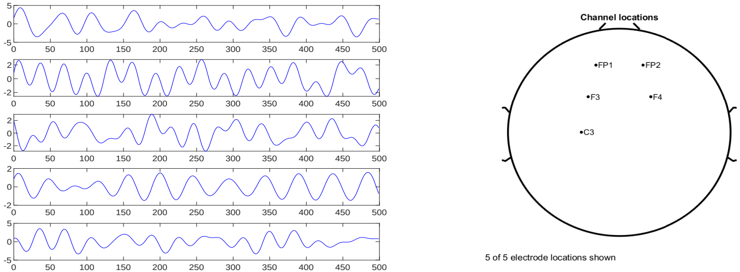


Figure 21: (**Example** 3) *Left: 5 components. Right: locations of* 5 *channels over head.*

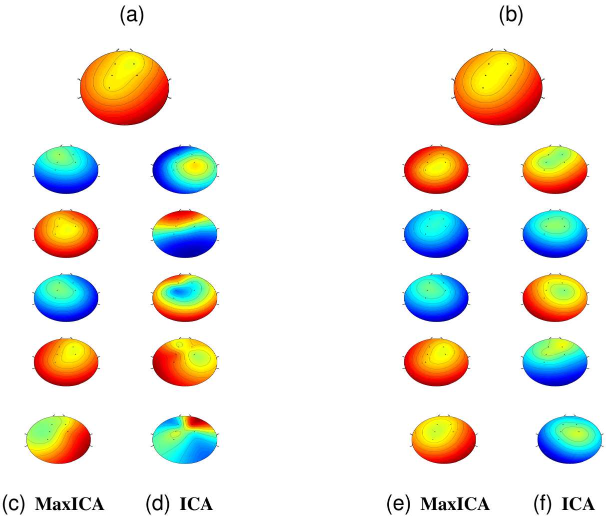
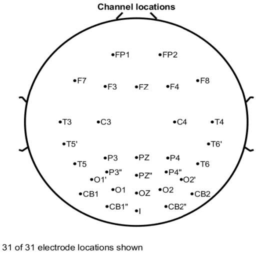


Figure 22: *Compare the decomposition of* $\mathbf{X}$ *into components* ($at\ t = 100$) *by* **MaxICA** *in* (c) *and* (e) *and by* **ICA** *in* (d) *and* (f). (a)*:* $\mathbf{X}$ *is linearly combined;* (b)*:* $\mathbf{X}$ *is max-linearly combined.*

Outline

Outline

- 32-channel EEG (Electroencephalography) data from 14 subjects (7 males, 7 females). Channel Cz is the reference channel. Electric brain potentials were recorded from 32 electrodes on an elastic cap.



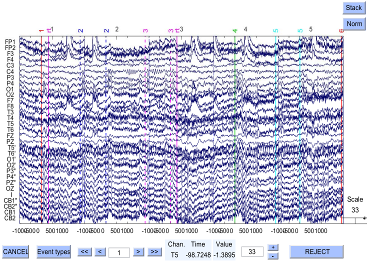


Figure 23: (**Visual processing data**) *Left: locations of* 31 *channels. Right: data of* 31 *channels.*

- Subjects participated in 2 tasks:

a go-nogo	**categorization**	task,
a go-nogo	**recognition**	task.

Tasks were operated by presenting photographs in front of subjects very briefly (Delorme *et al.* 2002).

Subject 1's **categorization** task

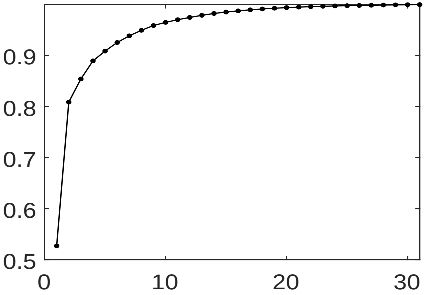


Figure 24: (**Visual processing data**) *Fraction of total variance retained versus the number of eigenvalues of subject 1's* **categorization** *task.*

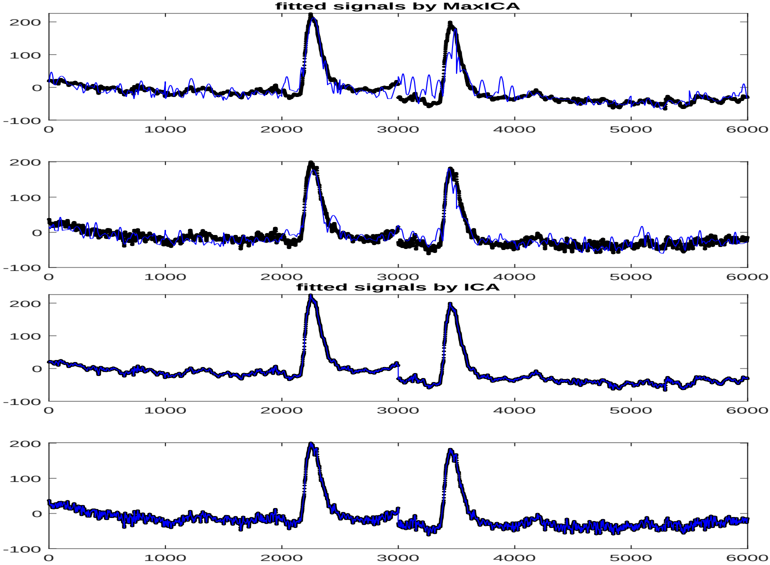


Figure 25: (**Visual processing data**) *Fitted signals* (*in* blue *lines*) *to observed signals* (*in dots* **.**) *of subject* 1*'s* **categorization** *task. Top: by* **MaxICA**. *Bottom: by* **ICA**.

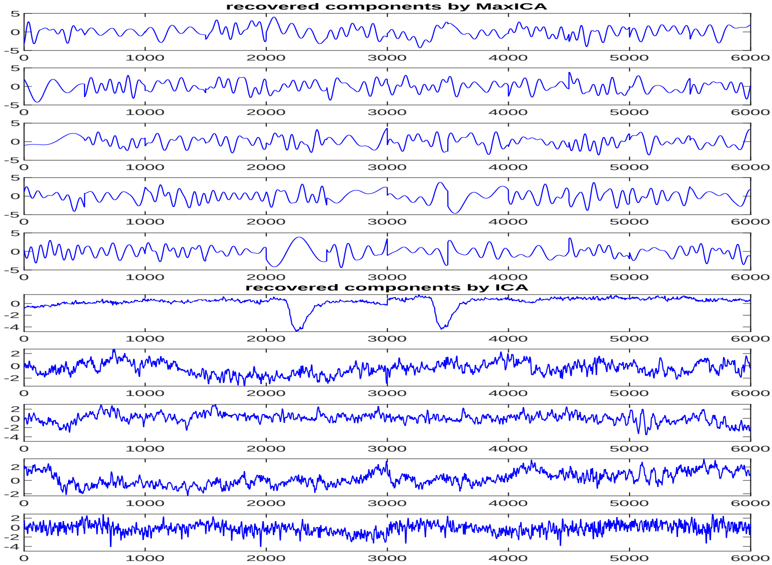


Figure 26: (**Visual processing data**) *Recovered components* (*in blue lines*) *of subject 1's* ***categorization*** *task. Top: by* **MaxICA**. *Bottom: by* **ICA**.

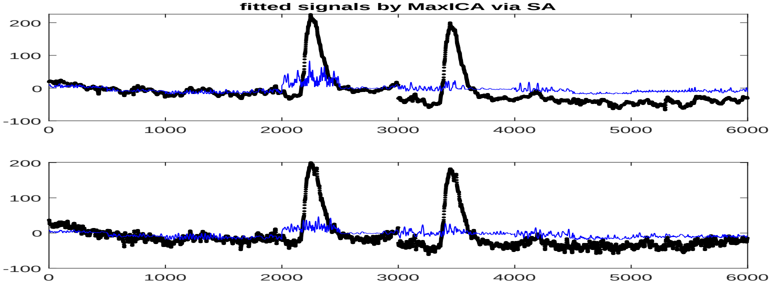


Figure 27: (**Visual processing data**) *Fitted signals* (*in blue lines*), *via the* **SA** *algorithm, by* **MaxICA**, *of subject* 1*'s* **categorization** *task, to observed signals* (*in dots* **.**).

Subject 1's **recognition** task

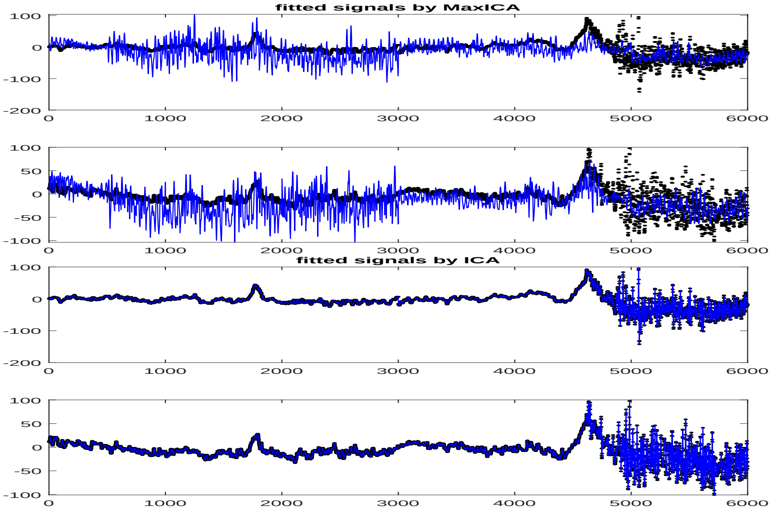


Figure 28: (**Visual processing data**) *Fitted signals* (*in blue lines*) *to observed signals* (*in dots* **.**) *of subject* 1*'s* **recognition** *task. Top: by* **MaxICA***. Bottom: by* **ICA***.*

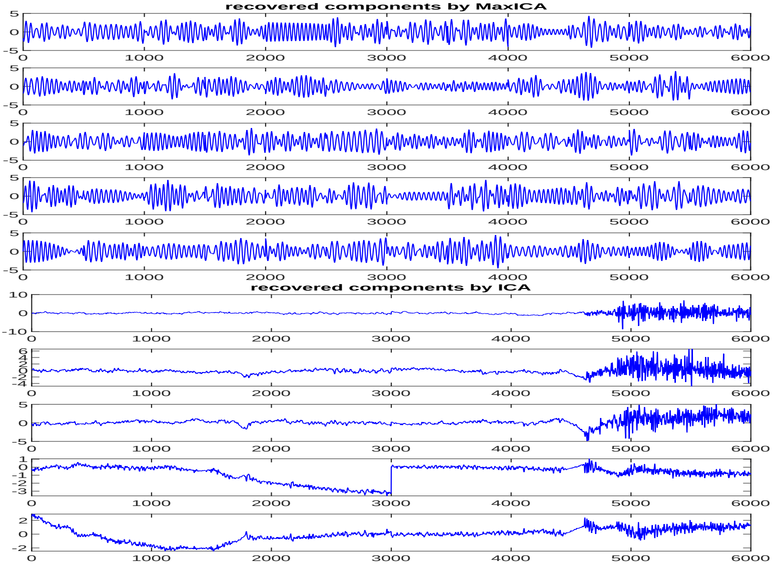


Figure 29: (**Visual processing data**) *Recovered components* (*in blue lines*) *of subject 1's* ***recognition*** *task. Top: by* **MaxICA**. *Bottom: by* **ICA**.

Subject 2's **categorization** task

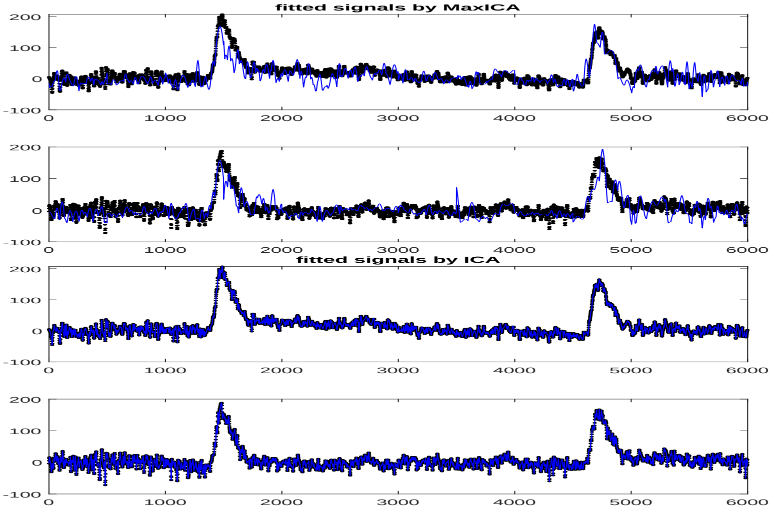


Figure 30: (**Visual processing data**) *Fitted signals (in blue lines) to observed signals (in dots .) of subject 2's* **categorization** *task. Top: by* **MaxICA**. *Bottom: by* **ICA**.

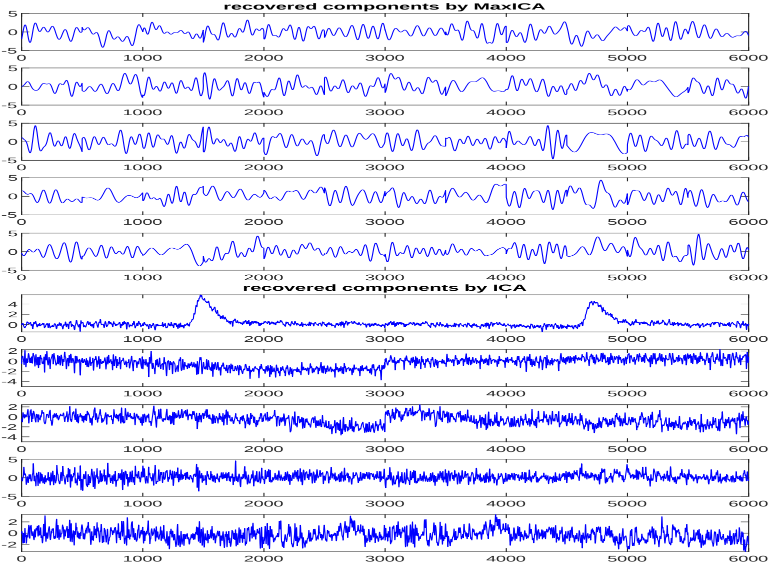


Figure 31: (**Visual processing data**) *Recovered components (in blue lines) of subject 2's **categorization** task. Top: by* **MaxICA**. *Bottom: by* **ICA**.

Outline

- **EEG** recordings from 5 epilepsy patients (Andrzejak *et al.* 2012).
- All patients underwent long term intracranial **EEG** recordings in the Department of Neurology at the University of Bern.
- All channels that detected first ictal **EEG** signal changes were classified as

  “**focal**” **EEG** channels,

  while all other channels were classified as

  “**non-focal**” **EEG** channels.
- Signals pairs were selected from “**focal**” channels and “**non-focal**” channels respectively.

One signal pair: focal channels

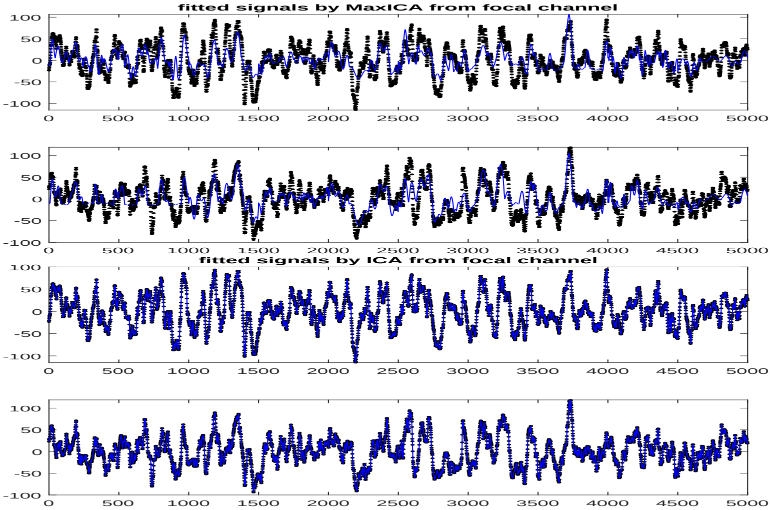


Figure 32: (**Epilepsy data, focal channels**) *Fitted signals (in* *blue* *lines) to observed signals (in dots* **.**) *for one signal pair. Top: by* **MaxICA**. *Bottom: by* **ICA**.

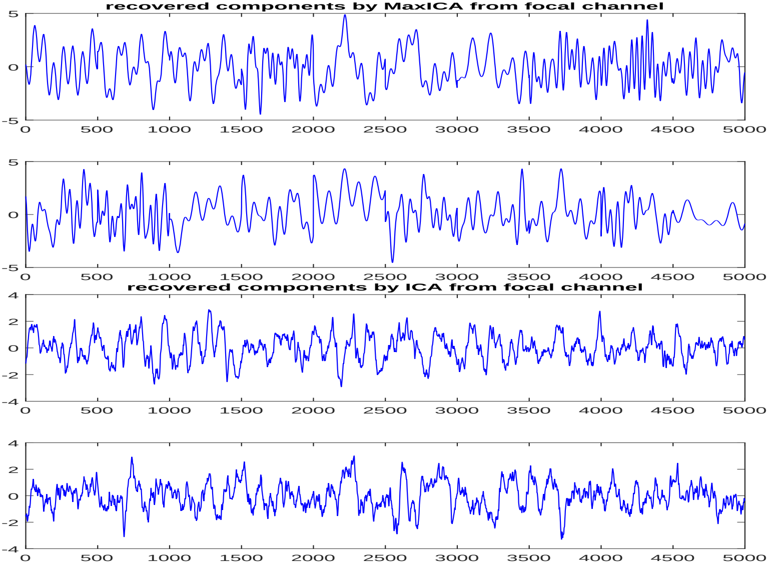


Figure 33: (**Epilepsy data, focal channels**) *Recovered components* (*in blue lines*). *Top: by* MaxICA. *Bottom: by* ICA.

non-focal channels:

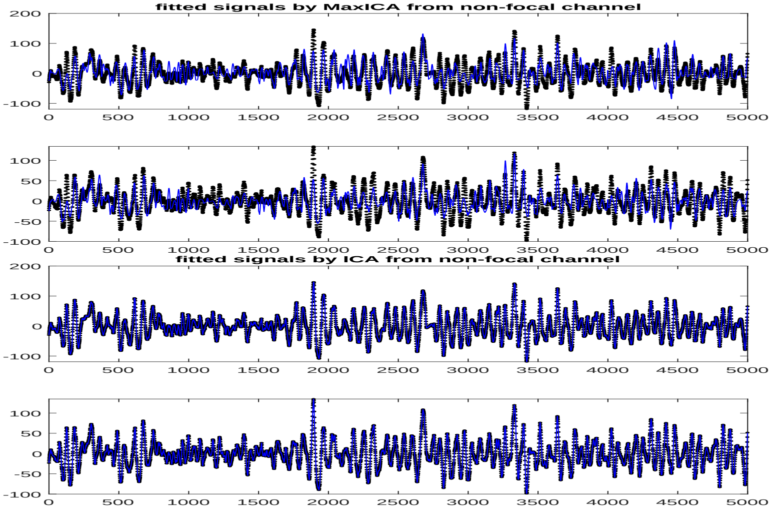


Figure 34: (**Epilepsy data, non-focal channels**) *Fitted signals (in* *blue* *lines) to observed signals (in dots* **.***) for one signal pair. Top: by* **MaxICA***. Bottom: by* **ICA***.*

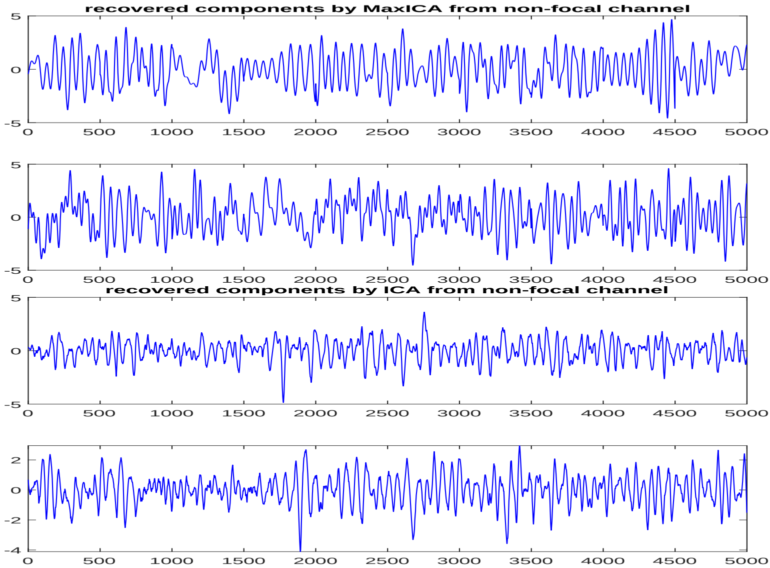


Figure 35: (**Epilepsy data, non-focal channels**) *Recovered components (in blue lines). Top: by* **MaxICA**. *Bottom: by* **ICA**.

Another signal pair: focal channels

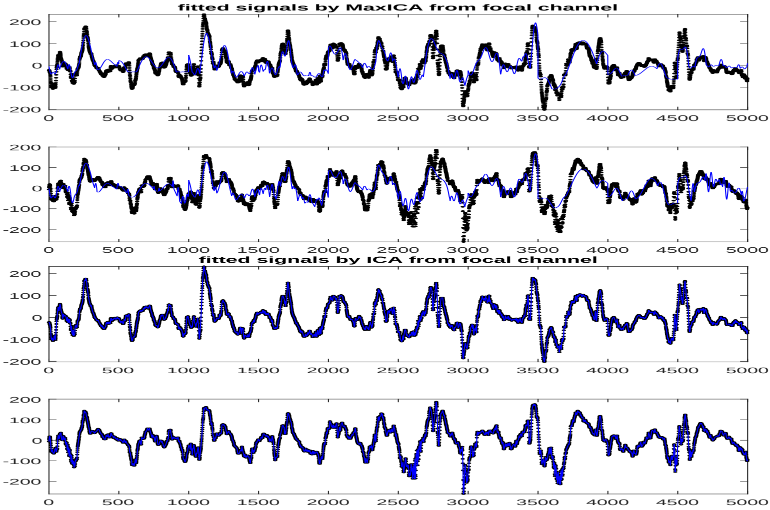


Figure 36: (**Epilepsy data, focal channels**) *Fitted signals (in blue lines) to observed signals (in dots* **.**) *for another signal pair. Top: by* **MaxICA**. *Bottom: by* **ICA**.

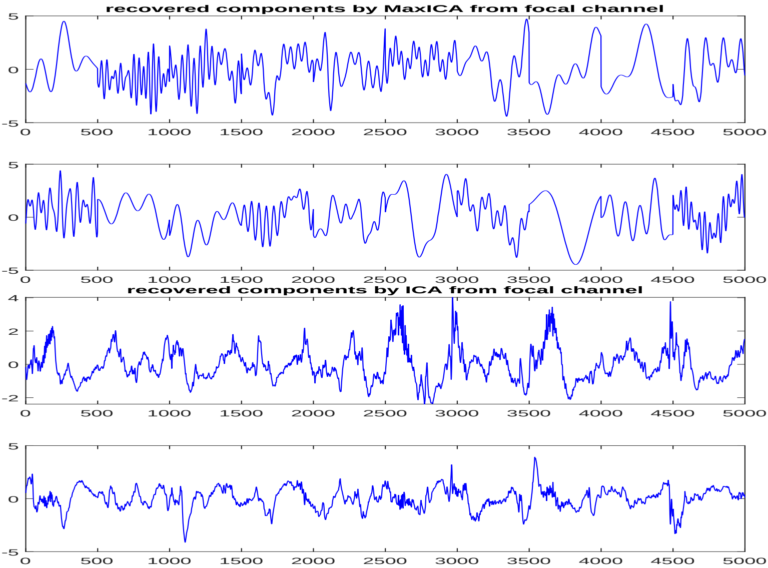


Figure 37: (**Epilepsy data, focal channels**) *Recovered components* (*in blue lines*) *for another signal pair. Top: by* MaxICA*. Bottom: by* ICA*.*

non-focal channels:

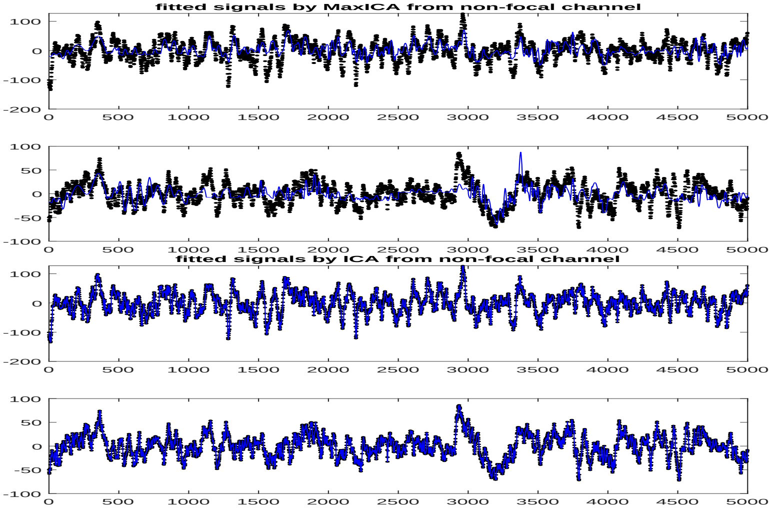


Figure 38: (**Epilepsy data, non-focal channels**) *Fitted signals (in blue lines) to observed signals (in dots .) for another signal pair. Top: by MaxICA. Bottom: by ICA.*

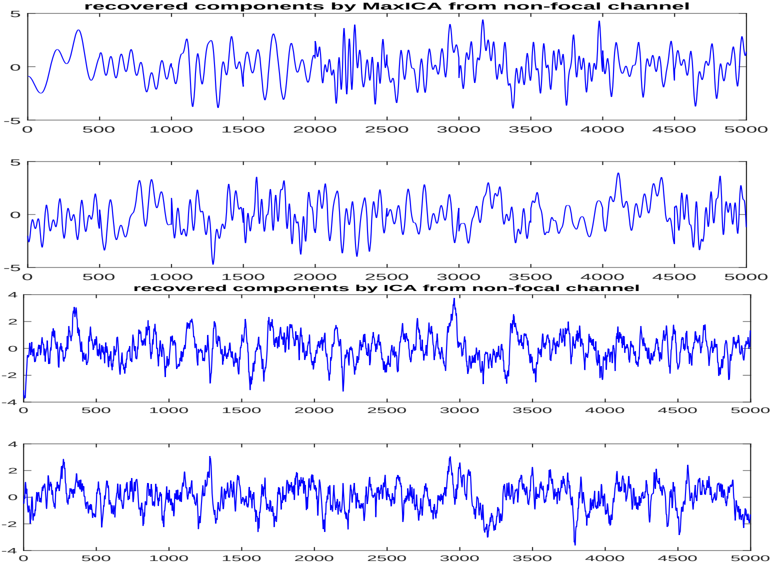


Figure 39: (**Epilepsy data, non-focal channels**) *Recovered components (in blue lines) for another signal pair. Top: by* MaxICA*. Bottom: by* ICA*.*

Outline

- A new **MaxICA** model is proposed with an "augmented genetic algorithm" for detecting blind sources.
- **MaxICA** and **ICA** aim to find hidden components, but have different application domains. It is not suggested to replace **ICA** by **MaxICA** or vice versa.
- Convergence of the **ERD_GA** algorithm:
- **MaxICA** can be extended in various other ways. Many future research topics (including theory, inference, and applications) will be under the proposed framework.

Thank you!

Summary of Latex and related files

For the paper:

- "PAPER/2018/Max_ICA_with_RSGuo_ZJZhang/MaxICA_paper/Paper/ draft_05.tex" (for the revision submitted on 08/25/2019); draft_06.tex (for the revision submitted on 10/19/2019); draft_07.tex (for the revision submitted on 10/27/2019);

For the slide:

- "PAPER/2018/Max_ICA_with_RSGuo_ZJZhang/Chunming_Paper/ slide_MaxICA_Guo_Zhang_Zhang_1.tex".

Other files:

- "PAPER/2018/Max_ICA_with_RSGuo_ZJZhang/Chunming_Paper/ slide_genetic_algorithm_Introduction.tex".
- "PAPER/2018/Max_ICA_with_RSGuo_ZJZhang/Chunming_Paper/ script_talk.docx".

Summary of codes

My own Matlab codes:

- at the directory "PAPER/2018/Max_ICA_with_RSGuo_ZJZhang/Chunming_Code/".

Joint Matlab codes:

- at the directory "PAPER/2018/Max_ICA_with_RSGuo_ZJZhang/MaxICA_paper/Codes/Version_1/": codes for the paper submitted on 01/13/2019.
- at the directory "PAPER/2018/Max_ICA_with_RSGuo_ZJZhang/MaxICA_paper/Codes/Version_2/": codes partly revised by R.S. Guo by using some functions.

- at the directory "`PAPER/2018/Max_ICA_with_RSGuo_ZJZhang/MaxICA_paper/Codes/Version_3/`": codes fully revised by C.M. Zhang by using all necessary functions. Adding "option_Guo" = 0 or 1 for using either Guo's indices i or Zhang's time pts t_i directly for $\widehat{\theta}$.
- at the directory "`PAPER/2018/Max_ICA_with_RSGuo_ZJZhang/MaxICA_paper/Codes/Version_4/`": codes for the revised paper submitted on 08/25/2016; delete "option_Guo"; delete the smooth part.
  - readme file and other issues: "`Read_me_version_4_and_other_issues.docx`".
  - Simulation studies: at "`simulation/`".
  - Real data analysis: at "`realdata/`".

- at the directory "`PAPER/2018/Max_ICA_with_RSGuo_ZJZhang/MaxICA_paper/Codes/Version_5/`":
  - Simulation studies: at "`simulation/`",
  - Real data analysis: at "`realdata/`",

  and related files
  - readme file and other issues (plotting locations and data of channels; links of 2 real data sets used in the paper): "`Readme_version_5_and_other_issues.docx`",

Flowchart of data analysis via the **MaxICA** model (7)

- Obtain the "data matrix" $\mathbf{X}$:
  - simulate via (25); get from the real data.

  Initial analysis via PCA for the number N of components.
- Reform $\mathbf{X}$ with segments; set-ups for estimating $\boldsymbol{\theta}$.
- Estimate parameters by $\widehat{\boldsymbol{\theta}}$ in the **MaxICA** model: via

  either "**ERD_GA**", or "**classical_GA**", or "**SA**".

  Here,

$$\begin{array}{ccc} \mathbf{X} & \longrightarrow & \mathbf{X}^* = \begin{cases} \mathbf{X}; \\ \text{smoothed}(\mathbf{X}). \end{cases} \\ \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}; \mathbf{X}^*) & \longrightarrow & \widehat{\boldsymbol{\theta}} \end{array}$$

- Fit data sequences $\{X_j(t_i) : j = 1, \ldots, p\}$ for the **MaxICA** model: via $\widehat{\boldsymbol{\theta}}$ alone (without $\mathbf{X}$);
- Recover components $\{S_k(t_i) : k = 1, \ldots, N\}$ in the **MaxICA** model: via $\widehat{\boldsymbol{\theta}}$ alone (without $\mathbf{X}$).

Notation

- Mixing matrix in the **ICA** model (2):

$$\mathrm{A} = (a_{j,k})_{j=1,\dots,p;k=1,\dots,N} \left(\boldsymbol{a}_1, \dots, \boldsymbol{a}_N \right) \in \mathbb{R}^{p \times N}. \tag{18}$$

- Mixing matrix in the **MaxICA** model (7):

$$\mathrm{B} = (b_{j,k})_{j=1,\dots,p;k=1,\dots,N} \left(\boldsymbol{b}_1, \dots, \boldsymbol{b}_N \right) \in \mathbb{R}^{p \times N}. \tag{19}$$

- $b_{j,k,\ell}$ in the **MaxICA** model (6): the ℓth sine wave, of the kth source component, from the jth data sequence, for
  $j = 1, \dots, p$: p is the number of data sequences;
  $k = 1, \dots, N$: N is the number of hidden components;
  $\ell = 1, \dots, n_k$: n_k is the number of sine waves;
  $i = 1, \dots, m$: m is the number of time points;
- $\{t_i : i = 1, \dots, m\}$: time points;
  m: number of time points in each observed signal.

- For $X_j(t_i) \overset{(25)}{=} \text{signal}_j(t_i) + \epsilon_j(t_i)$, $j = 1, \ldots, p$: where
  - in **ICA**,

$$\text{signal}_j(t_i) = \sum_{k=1}^{N} a_{j,k}\, S_k^*(t) = a_{j,1}\, S_1^*(t) + \cdots + a_{j,N}\, S_N^*(t),$$

  - in **MaxICA**,

$$\text{signal}_j(t_i) = \max_{1\le k\le N} b_{j,k}\, S_k(t) = \max\{b_{j,1}\, S_1(t), \cdots, b_{j,N}\, S_N(t)\},$$

we use the notation:

- $X_j(t_i)$: $\boxed{\text{observed}}$ signals, measured signals, data sequences, "mixed signals", original recordings;
- $\text{signal}_j(t_i)$: $\boxed{\text{true}}$ signals,
  - ⋆ $S_k(t) \overset{(6)}{\equiv} \sum_{\ell=1}^{n_k} \sin(\alpha_{k,\ell}\, t + \beta_{k,\ell})$, $k = 1, \ldots, N$: $\boxed{\text{component}}$ signals in **MaxICA**,
  - ⋆ $S_{j,k}(t) = a_{j,k}\, S_k^*(t)$: $\boxed{\text{source}}$ signals in **ICA**.

    $S_{j,k}(t) = b_{j,k}\, S_k(t)$: $\boxed{\text{source}}$ signals in **MaxICA**.

- The frequency, phase, mixing coefficients in the vector $\boldsymbol{\theta}$ are arranged:

$$
\begin{array}{ccc}
(\alpha_{1,1}, \beta_{1,1}), \ldots, (\alpha_{1,\ell}, \beta_{1,\ell}), \ldots, (\alpha_{1,n_1}, \beta_{1,n_1}); & 2n_1 & \longrightarrow \\
\cdots & & \\
(\alpha_{k,1}, \beta_{k,1}), \ldots, (\alpha_{k,\ell}, \beta_{k,\ell}), \ldots, (\alpha_{k,n_k}, \beta_{k,n_k}); & 2(n_1 + \cdots + n_k) & \longrightarrow \\
\cdots & & \\
(\alpha_{N,1}, \beta_{N,1}), \ldots, (\alpha_{N,\ell}, \beta_{N,\ell}), \ldots, (\alpha_{N,n_N}, \beta_{N,n_N}); & 2(n_1 + \cdots + n_N) & \longrightarrow \\
(b_{1,1}, \ldots, b_{1,N}), \ldots, (b_{j,1}, \ldots, b_{j,N}), \ldots, (b_{p,1}, \ldots, b_{p,N}). & 2(n_1 + \cdots + n_N) + N \times p &
\end{array}
$$

- $\{\widehat{\theta}^{(s)} : s = 1, \ldots, n\}$: solutions of (9) in a population;
  n in (10): number of <u>solution</u>s in a population.
- $L_s = L(\widehat{\theta}^{(s)})$, $s = 1, \ldots, n$: loss function of the sth solution $\widehat{\theta}^{(s)}$. A smaller loss L_s corresponds to a better solution, and a higher fitness value $\mathfrak{f}_s = \mathfrak{f}(\widehat{\theta}^{(s)})$.
- $\mathfrak{f}_1, \mathfrak{f}_2, \ldots, \mathfrak{f}_n$: fitness values of n chromosomes in the population.
- $d_j = L_{j+1} - L_j$ in (11), $j = 1, \ldots, n-1$:
- $p_s = \sum_{k=1}^{s} \sum_{j=k}^{n-1} d_j / D$ in (12), $s = 1, \ldots, n-1$,

- For

$$\boldsymbol{X}(t) = \begin{pmatrix} X_1(t) \\ \vdots \\ X_j(t) \\ \vdots \\ X_p(t) \end{pmatrix} \in \mathbb{R}^{p\times 1}, \qquad \vec{\boldsymbol{X}}_j = \begin{pmatrix} X_j(t_1) \\ \vdots \\ X_j(t_m) \end{pmatrix} \in \mathbb{R}^{m\times 1},\ j = 1, \dots, p,$$

define the "$\boxed{\text{data matrix}}$",

$$\mathbf{X} = \begin{pmatrix} X_1(t_1) & \cdots & X_1(t_m) \\ \cdots & \cdots & \cdots \\ X_j(t_1) & \cdots & X_j(t_m) \\ \cdots & \cdots & \cdots \\ X_p(t_1) & \cdots & X_p(t_m) \end{pmatrix} = \Big(\boldsymbol{X}(t_1), \dots, \boldsymbol{X}(t_m)\Big) = \begin{pmatrix} \vec{\boldsymbol{X}}_1^T \\ \cdots \\ \vec{\boldsymbol{X}}_j^T \\ \cdots \\ \vec{\boldsymbol{X}}_p^T \end{pmatrix} \in \mathbb{R}^{p\times m}. \tag{20}$$

- For **ICA**,

$$\boldsymbol{S}(t) = \begin{pmatrix} S_1(t) \\ \vdots \\ S_k(t) \\ \vdots \\ S_N(t) \end{pmatrix} \in \mathbb{R}^{N\times 1}, \qquad \vec{\boldsymbol{S}}_k = \begin{pmatrix} S_k(t_1) \\ \vdots \\ S_k(t_m) \end{pmatrix} \in \mathbb{R}^{m\times 1},\ k = 1, \ldots, N,$$

define the " $\boxed{\text{component matrix}}$ " for **ICA**,

$$\mathbf{S} = \begin{pmatrix} S_1(t_1) & \cdots & S_1(t_m) \\ \cdots & \cdots & \cdots \\ S_k(t_1) & \cdots & S_k(t_m) \\ \cdots & \cdots & \cdots \\ S_N(t_1) & \cdots & S_N(t_m) \end{pmatrix} = \Big(\boldsymbol{S}(t_1), \ldots, \boldsymbol{S}(t_m)\Big) = \begin{pmatrix} \vec{\boldsymbol{S}}_1^T \\ \cdots \\ \vec{\boldsymbol{S}}_k^T \\ \cdots \\ \vec{\boldsymbol{S}}_N^T \end{pmatrix} \in \mathbb{R}^{N\times m}. \quad (21)$$

For observed signals in the $\mathbf{ICA}$ model (2),

$$\begin{aligned} X_j(t) &\overset{(2)}{=} a_{j,1}\, S_1(t) + \cdots + a_{j,N}\, S_N(t) = (a_{j,1}, \ldots, a_{j,N}) \begin{pmatrix} S_1(t) \\ \cdots \\ S_N(t) \end{pmatrix} \\ &\overset{(18)}{=} \sum_{k=1}^{N} a_{j,k}\, S_k(t) = \boldsymbol{e}_j^T \mathrm{A}\, \boldsymbol{S}(t), \quad j = 1, \ldots, p, \end{aligned}$$

thus the jth row vector, the ith column vector, the data matrix are

$$\begin{aligned} \vec{\boldsymbol{X}}_j^T &= (X_j(t_1), \ldots, X_j(t_m)) \\ &= \boldsymbol{e}_j^T \mathrm{A}\, (\boldsymbol{S}(t_1), \ldots, \boldsymbol{S}(t_m)) \overset{(21)}{=} \boldsymbol{e}_j^T \mathrm{A}\, \mathbf{S} \\ &= \sum_{k=1}^{N} a_{j,k}\, \vec{\boldsymbol{S}}_k^T, \quad j = 1, \ldots, p, \\ \boldsymbol{X}(t_i) &= \begin{pmatrix} X_1(t_i) \\ \vdots \\ X_p(t_i) \end{pmatrix} = \mathrm{A}\, \boldsymbol{S}(t_i) = \boxed{\sum_{k=1}^{N} \boldsymbol{a}_k\, S_k(t_i)}, \quad i = 1, \ldots, m, \end{aligned}$$

$$\mathbf{X} \overset{(3)}{=} \mathrm{A}\,\mathbf{S} \overset{(18),(21)}{=\!=\!=} (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k, \ldots, \boldsymbol{a}_N) \begin{pmatrix} \vec{\boldsymbol{s}}_1^T \\ \cdots \\ \vec{\boldsymbol{s}}_k^T \\ \cdots \\ \vec{\boldsymbol{s}}_N^T \end{pmatrix} = \boxed{\sum_{k=1}^{N} \boldsymbol{a}_k\, \vec{\boldsymbol{S}}_k^T} \tag{22}$$

$$\overset{(29)}{=} \begin{pmatrix} \mathbf{1}_N^T M_1 \\ \vdots \\ \mathbf{1}_N^T M_p \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^N a_{1,k} \vec{\boldsymbol{S}}_k^T \\ \vdots \\ \sum_{k=1}^N a_{p,k} \vec{\boldsymbol{S}}_k^T \end{pmatrix} = \sum_{k=1}^{N} \begin{pmatrix} a_{1,k} \vec{\boldsymbol{S}}_k^T \\ \vdots \\ a_{p,k} \vec{\boldsymbol{S}}_k^T \end{pmatrix},$$

where

$$M_j \overset{(28)}{=} \mathrm{diag}(\boldsymbol{e}_j^T \mathrm{A})\,\mathbf{S} = \begin{pmatrix} a_{j,1} \vec{\boldsymbol{S}}_1^T \\ \vdots \\ a_{j,N} \vec{\boldsymbol{S}}_N^T \end{pmatrix}.$$

- For **MaxICA**,

$$\boldsymbol{S}(t) = \begin{pmatrix} S_1(t) \\ \vdots \\ S_k(t) \\ \vdots \\ S_N(t) \end{pmatrix} \in \mathbb{R}^{N\times 1}, \qquad \vec{\boldsymbol{S}}_k = \begin{pmatrix} S_k(t_1) \\ \vdots \\ S_k(t_m) \end{pmatrix} \in \mathbb{R}^{m\times 1},\ k = 1, \ldots, N,$$

define the "$\boxed{\text{component matrix}}$" for **MaxICA**,

$$\mathbf{S} = \begin{pmatrix} S_1(t_1) & \cdots & S_1(t_m) \\ \cdots & \cdots & \cdots \\ S_k(t_1) & \cdots & S_k(t_m) \\ \cdots & \cdots & \cdots \\ S_N(t_1) & \cdots & S_N(t_m) \end{pmatrix} = \Big(\boldsymbol{S}(t_1), \ldots, \boldsymbol{S}(t_m) \Big) = \begin{pmatrix} \vec{\boldsymbol{S}}_1^T \\ \cdots \\ \vec{\boldsymbol{S}}_k^T \\ \cdots \\ \vec{\boldsymbol{S}}_N^T \end{pmatrix} \in \mathbb{R}^{N\times m}. \qquad (23)$$

For observed signals in the **MaxICA** model (7),

$$\begin{aligned} X_j(t) &\overset{(7)}{=} \max_{1\le k\le N} b_{j,k}\, S_k(t) \\ &= \max\{b_{j,1}\, S_1(t), \ldots, b_{j,N}\, S_N(t)\}, \quad j = 1, \ldots, p, \end{aligned}$$

thus the jth row vector, the ith column vector, the data matrix are

$$\begin{aligned} \vec{\boldsymbol{X}}_j^T &= (X_j(t_1), \ldots, X_j(t_m)) \\ &= \max\{b_{j,1}\, \vec{\boldsymbol{S}}_1^T, \ldots, b_{j,N}\, \vec{\boldsymbol{S}}_N^T\}, \quad j = 1, \ldots, p, \\ \boldsymbol{X}(t_i) &= \max \begin{pmatrix} b_{1,1}\, S_1(t_i), & \ldots & , b_{1,N}\, S_N(t_i) \\ & \vdots & \\ b_{p,1}\, S_1(t_i), & \ldots & , b_{p,N}\, S_N(t_i) \end{pmatrix} \\ &= \boxed{\max\{\boldsymbol{b}_1\, S_1(t_i), \ldots, \boldsymbol{b}_N\, S_N(t_i)\}}, \quad i = 1, \ldots, m, \end{aligned}$$

$$\mathbf{X} \overset{(31)}{=} \begin{pmatrix} \max(M_1) \\ \vdots \\ \max(M_p) \end{pmatrix} = \begin{pmatrix} \max\{b_{1,1}\,\vec{\boldsymbol{s}}_1^T, & \ldots, & b_{1,N}\,\vec{\boldsymbol{s}}_N^T\} \\ \vdots & & \\ \max\{b_{p,1}\,\vec{\boldsymbol{s}}_1^T, & \ldots, & b_{p,N}\,\vec{\boldsymbol{s}}_N^T\} \end{pmatrix}$$

$$= \max \begin{pmatrix} b_{1,1}\,\vec{\boldsymbol{s}}_1^T, & \ldots, & b_{1,N}\,\vec{\boldsymbol{s}}_N^T \\ & \vdots & \\ b_{p,1}\,\vec{\boldsymbol{s}}_1^T, & \ldots, & b_{p,N}\,\vec{\boldsymbol{s}}_N^T \end{pmatrix} = \boxed{\max\{\boldsymbol{b}_1\vec{\boldsymbol{s}}_1^T, \ldots, \boldsymbol{b}_N\vec{\boldsymbol{s}}_N^T\}}, \quad (24)$$

where

$$M_j \overset{(37)}{=} \mathrm{diag}(\boldsymbol{e}_j^T \mathrm{B})\,\mathbf{S} = \begin{pmatrix} b_{j,1}\,\vec{\boldsymbol{s}}_1^T \\ \vdots \\ b_{j,N}\,\vec{\boldsymbol{s}}_N^T \end{pmatrix}.$$

- Model:

$$
\begin{aligned}
\boxed{\text{data\_matrix}} &= \boxed{\text{signal\_matrix}} + \text{noise\_matrix}, \\
\mathbf{X}_{p\times m} &= \mathcal{S}_{p\times m} + \mathbf{E}_{p\times m}, \\
\begin{pmatrix} X\text{\_1\_vector} \\ \cdots \\ \boxed{X\text{\_}j\text{\_vector}} \\ \cdots \\ X\text{\_}p\text{\_vector} \end{pmatrix} &= \begin{pmatrix} \text{signal\_1\_vector} \\ \cdots \\ \boxed{\text{signal\_}j\text{\_vector}} \\ \cdots \\ \text{signal\_}p\text{\_vector} \end{pmatrix} + \begin{pmatrix} \epsilon\text{\_1\_vector} \\ \cdots \\ \epsilon\text{\_}j\text{\_vector} \\ \cdots \\ \epsilon\text{\_}p\text{\_vector} \end{pmatrix},
\end{aligned}
\tag{25}
$$

where

$$
\boxed{X\text{\_}j\text{\_vector}} = (X_j(t_1), \cdots, X_j(t_m)). \tag{26}
$$

- In **ICA** (2),

$$\begin{aligned}
\text{signal_}j\text{_vector} &= (a_{j,1},\ldots,a_{j,N})\begin{pmatrix} S_1(t_1) & \cdots & S_1(t_m) \\ \vdots & & \vdots \\ S_N(t_1) & \cdots & S_N(t_m)\end{pmatrix} \\
&= (\boldsymbol{e}_j^T \mathrm{A})\,\mathbf{S}, \quad j = 1,\ldots,p, && (27) \\
&= \mathbf{1}_N^T \operatorname{diag}(\boldsymbol{e}_j^T \mathrm{A})\,\mathbf{S} \equiv \boxed{\mathbf{1}_N^T M_j}, && (28) \\
\text{signal_matrix} &\overset{(27)}{=} \mathrm{A}\,\mathbf{S} \overset{(28)}{=} \boxed{\begin{pmatrix} \mathbf{1}_N^T & & \\ & \ddots & \\ & & \mathbf{1}_N^T \end{pmatrix}\begin{pmatrix} M_1 \\ \vdots \\ M_p \end{pmatrix}}. && (29)
\end{aligned}$$

- In **MaxICA** (7),

$$\begin{aligned}
\text{signal_}j\text{_vector} &\overset{(33),(37)}{=\!=\!=} \max\{\operatorname{diag}(\boldsymbol{e}_j^T \mathrm{B})\,\mathbf{S}\} \overset{(33)}{=} \boxed{\max(M_j)}, \quad j = 1,\ldots,p, && (30) \\
\text{signal_matrix} &\overset{(30)}{=} \boxed{\begin{pmatrix} \max(M_1) \\ \vdots \\ \max(M_p) \end{pmatrix}}. && (31)
\end{aligned}$$

Procedure to generate the data matrix **X** in (25) for **MaxICA** model (7)

From model (4),

$$\begin{pmatrix} X_1(t) \\ \cdots \\ X_j(t) \\ \cdots \\ X_p(t) \end{pmatrix} = \begin{pmatrix} \max \begin{pmatrix} S_{1,1}(t) \\ \vdots \\ S_{1,N}(t) \end{pmatrix} \\ \vdots \\ \max \begin{pmatrix} S_{j,1}(t) \\ \vdots \\ S_{j,N}(t) \end{pmatrix} \\ \vdots \\ \max \begin{pmatrix} S_{p,1}(t) \\ \vdots \\ S_{p,N}(t) \end{pmatrix} \end{pmatrix} + \begin{pmatrix} \epsilon_1(t) \\ \vdots \\ \epsilon_j(t) \\ \vdots \\ \epsilon_p(t) \end{pmatrix}.$$

Thus,

$$\begin{pmatrix} X_1(t_1) & \cdots & X_1(t_m) \\ \cdots & \cdots & \cdots \\ X_j(t_1) & \cdots & X_j(t_m) \\ \cdots & \cdots & \cdots \\ X_p(t_1) & \cdots & X_p(t_m) \end{pmatrix} = \begin{pmatrix} \max \begin{pmatrix} S_{1,1}(t_1) & \cdots & S_{1,1}(t_m) \\ \cdots & \cdots & \cdots \\ S_{1,N}(t_1) & \cdots & S_{1,N}(t_m) \end{pmatrix} \\ \vdots \\ \max \begin{pmatrix} S_{j,1}(t_1) & \cdots & S_{j,1}(t_m) \\ \cdots & \cdots & \cdots \\ S_{j,N}(t_1) & \cdots & S_{j,N}(t_m) \end{pmatrix} \\ \vdots \\ \max \begin{pmatrix} S_{p,1}(t_1) & \cdots & S_{p,1}(t_m) \\ \cdots & \cdots & \cdots \\ S_{p,N}(t_1) & \cdots & S_{p,N}(t_m) \end{pmatrix} \end{pmatrix} + \begin{pmatrix} \epsilon_1(t_1) & \cdots & \epsilon_1(t_m) \\ \cdots & \cdots & \cdots \\ \epsilon_j(t_1) & \cdots & \epsilon_j(t_m) \\ \cdots & \cdots & \cdots \\ \epsilon_p(t_1) & \cdots & \epsilon_p(t_m) \end{pmatrix}$$

i.e.,

$$X\_j\_\text{vector} = \text{signal}\_j\_\text{vector} + \epsilon\_j\_\text{vector}, \tag{32}$$

where

$$\text{signal}\_j\_\text{vector} = \max(M_j), \quad j = 1, \ldots, p, \tag{33}$$

where

$$M_j = \begin{pmatrix} S_{j,1}(t_1) & \cdots & S_{j,1}(t_m) \\ \cdots & \cdots & \cdots \\ S_{j,N}(t_1) & \cdots & S_{j,N}(t_m) \end{pmatrix}_{N\times m}, \quad j = 1, \ldots, p. \tag{34}$$

So

$$\mathbf{X} \overset{(20)}{=} \begin{pmatrix} X\_1\_\text{vector} \\ \vdots \\ X\_j\_\text{vector} \\ \vdots \\ X\_p\_\text{vector} \end{pmatrix} = \begin{pmatrix} \max(M_1) \\ \vdots \\ \max(M_j) \\ \vdots \\ \max(M_p) \end{pmatrix}_{p\times m} + \begin{pmatrix} \epsilon\_1\_\text{vector} \\ \vdots \\ \epsilon\_j\_\text{vector} \\ \vdots \\ \epsilon\_p\_\text{vector} \end{pmatrix}. \tag{35}$$

For $S_{j,k}(t) \overset{(6)}{=} b_{j,k}\, S_k(t)$,

$$
\begin{aligned}
\boxed{M_j} &\overset{(34)}{=} \begin{pmatrix} b_{j,1}\, S_1(t_1) & \cdots & b_{j,1}\, S_1(t_m) \\ \cdots & \cdots & \cdots \\ b_{j,N}\, S_N(t_1) & \cdots & b_{j,N}\, S_N(t_m) \end{pmatrix} \\
&= \begin{pmatrix} b_{j,1} & & \\ & \ddots & \\ & & b_{j,N} \end{pmatrix} \begin{pmatrix} S_1(t_1) & \cdots & S_1(t_m) \\ \cdots & \cdots & \cdots \\ S_N(t_1) & \cdots & S_N(t_m) \end{pmatrix} \\
&= \begin{pmatrix} b_{j,1} & & \\ & \ddots & \\ & & b_{j,N} \end{pmatrix} \begin{pmatrix} S\_1\_\text{vector} \\ \vdots \\ S\_N\_\text{vector} \end{pmatrix} \in \mathbb{R}^{N \times m} && (36) \\
&\overset{(23)}{=} \operatorname{diag}(b_{j,1}, \ldots, b_{j,N})\, \mathbf{S} = \operatorname{diag}(\boldsymbol{e}_j^T \mathrm{B})\, \mathbf{S}, && (37)
\end{aligned}
$$

where

$$
\boxed{S\_k\_\text{vector}} = (S_k(t_1), \ldots, S_k(t_m)), \quad k = 1, \ldots, N. \qquad (38)
$$

$$\mathbf{S} \overset{(23)}{=} \begin{pmatrix} S\_1\_vector \\ \vdots \\ S\_k\_vector \\ \vdots \\ S\_N\_vector \end{pmatrix}_{N \times m}, \tag{39}$$

and B is the coefficient matrix,

$$\mathrm{B} = (b_{j,k}) = \begin{pmatrix} b_{1,1} & \cdots & b_{1,N} \\ \cdots & \cdots & \cdots \\ b_{j,1} & \cdots & b_{j,N} \\ \cdots & \cdots & \cdots \\ b_{p,1} & \cdots & b_{p,N} \end{pmatrix}_{p \times N}. \tag{40}$$

In summary

- Inputs:
  - $\boxed{\mathrm{T}}$ for the time interval $[0, \mathrm{T}]$;
    $\boxed{m}$ for time points $t_1, \ldots, t_m$; # of segments for **MaxICA**;
  - matrix $\mathrm{B} = (b_{j,k}) \in \mathbb{R}^{p \times N}$ (mixing coefficients) in (40), $j = 1, \ldots, \boxed{p}$ (sequences) and $k = 1, \ldots, \boxed{N}$ (components),
  - array $\{\mathrm{a}_{k,\ell}\}$ (amplitudes), array $\{\alpha_{k,\ell}\}$ (frequencies), array $\{\beta_{k,\ell}\}$ (phases), $k = 1, \ldots, \boxed{N}$ and $\ell = 1, \ldots, \boxed{n_k}$,
    scalar γ (location)

- Compute
  - the kth component function: $S_k(t) \overset{(6)}{=} \sum_{\ell=1}^{n_k} \boxed{\mathrm{a}_{k,\ell}} \sin(\boxed{\alpha_{k,\ell}}\, t + \boxed{\beta_{k,\ell}})$, $k = 1, \ldots, N$.
  - the kth component vector: $S\_k\_\text{vector} \overset{(38)}{=} (S_k(t_1), \ldots, S_k(t_m))$, $k = 1, \ldots, N$.

  - the jth mean matrix: $M_j \in \mathbb{R}^{N \times m}$, $j = 1, \ldots, p$, via B and
    - either computing (36),
    - or first computing the "component matrix" $\mathbf{S} \in \mathbb{R}^{N \times m}$ in (39) and then computing (37).
  - the jth signal vector: $\text{signal\_}j\text{\_vector} \overset{(33)}{=} \max(M_j)$, $j = 1, \ldots, p$,
  - the jth data vector: X_j_vector in (32). The "data matrix" $\mathbf{X} \in \mathbb{R}^{p \times m}$ in (35).
- Note: R.S. Guo's version for estimating α used $\sin(\alpha_i\, i + \beta)$, rather than $\sin(\alpha_t\, t_i + \beta)$. The conversion I adopted is

$$\alpha_i \frac{m}{\#\mathrm{S}} = \alpha_t \frac{\mathrm{T}}{\#\mathrm{S}} \Longleftrightarrow \alpha_i\, m = \alpha_t\, \mathrm{T} \Longleftrightarrow \boxed{\alpha_t = \alpha_i \frac{m}{\mathrm{T}}}.$$

An interval partitioned with segments

For an interval $[a, b]$ equally partitioned with m points t_i, starting at a and ending at b: $t_i = a + i\frac{b-a}{m-1}$, $i = 0, 1, \dots, (m-1)$.

- If $a = 0$, then

$$t_i = i\frac{b}{m-1} \iff i = \frac{t_i}{\frac{b}{m-1}} = t_i\frac{m-1}{b}.$$

- For $m = m_s\,\#\text{S}$, where #S is the # of segments, each segment has the length $\frac{b}{\#\text{S}}$.
  Then indices i of t_i located in segments are

$$\begin{aligned}
\text{segment 1}: &\quad i = 0, 1, \dots, (m_s - 1),\\
\text{segment 2}: &\quad i = m_s, m_s + 1, \dots, (2 * m_s - 1),\\
\dots &\quad \dots\\
\text{segment seg}: &\quad i = (\text{seg} - 1) * m_s, (\text{seg} - 1) * m_s + 1, \dots, (\text{seg} * m_s - 1)\\
\dots &\quad \dots\\
\text{segment \#S}: &\quad i = (\#\text{S} - 1) * m_s, (\#\text{S} - 1) * m_s + 1, \dots, (\#\text{S} * m_s - 1).
\end{aligned}$$

Some details

- The **ICA** model (2) is

$$X_j(t) \overset{(2)}{=} \sum_{k=1}^{N} a_{j,k}\, S_k(t), \quad j = 1, \ldots, p.$$

The **MaxICA** model (4) is

$$\begin{aligned} X_j(t) &\overset{(4)}{=} \max\{S_{j,k}(t) : k = 1, \ldots, N\}, \quad j = 1, \ldots, p \\ &\overset{(6)}{=} \max\left\{b_{j,k}\, S_k(t) : k = 1, \ldots, N\right\}, \quad j = 1, \ldots, p \\ &= \max_{k=1,\ldots,N} b_{j,k}\, S_k(t), \quad j = 1, \ldots, p. \end{aligned}$$

So, the **ICA** model (2) can be regarded as using “sum” to replace “maximum” in the **MaxICA** model (4)–(7), i.e.,

$$S_{j,k}(t) \overset{(6)}{=} \boxed{b_{j,k}\, S_k(t)} \text{ in (7)} \quad \leftrightsquigarrow \quad \boxed{a_{j,k}\, S_k(t)} \text{ in (2)}.$$

- (7) is a special case of (4): “$b_{j,k,\ell}$ in (4), $\ell = 1, \ldots, n_k$, with $n_k = 2$” is “$b_{j,k}$ in (7)”.

- For $S_{j,k}(t) \overset{(5)}{=} \sum_{\ell=1}^{n_k} b_{j,k,\ell} \sin(\alpha_{k,\ell}\, t + \beta_{k,\ell})$, $j = 1, \ldots, p$, $k = 1, \ldots, N$, the parameters are
  - $\{b_{j,k,\ell} : j = 1, \ldots, p;\ k = 1, \ldots, N;\ \ell = 1, \ldots, n_k\}$: $p\,(n_1 + \cdots + n_N)$,
  - $\{(\alpha_{k,\ell}, \beta_{k,\ell}) : k = 1, \ldots, N;\ \ell = 1, \ldots, n_k\}$: $(n_1 + \cdots + n_N) \times 2$.

  The phase of an oscillation or signal refers to a sinusoidal function,

$$y(t) \quad = \quad A \cdot \sin(2\pi \mathrm{f}\, t + \varphi),$$

  where A, f and φ are constant parameters called the "amplitude", "frequency", and "phase" of the sinusoid.
- For $S_{j,k}(t) \overset{(6)}{=} b_{j,k} \sum_{\ell=1}^{n_k} \sin(\alpha_{k,\ell}\, t + \beta_{k,\ell})$, $j = 1, \ldots, p$, $k = 1, \ldots, N$, the parameters are
  - $\{b_{j,k} : j = 1, \ldots, p;\ k = 1, \ldots, N\}$: $p\,N$,
  - $\{(\alpha_{k,\ell}, \beta_{k,\ell}) : k = 1, \ldots, N;\ \ell = 1, \ldots, n_k\}$: $(n_1 + \cdots + n_N) \times 2$.
- Challenges of our optimization problem (9): the loss function is non-convex, and non-differentiable.

- In figures,
  - "**white** lines" for **true** $\mathrm{signal}_j(t)$ when overlapping with 100 sets of fitted signals $\{X_j(t_i) : j = 1, \ldots, p\}$ corresponding to 100 Monte Carlo runs;
  - "**black .**" for "**observed**" signals $X_j(t)$;
    "**blue** lines" for "**fitted**" signals $X_j(t)$.
  - "**red –**" for "**true**" components $S_k(t)$;
    "**blue** lines" for "**recovered**" components $S_k(t)$.
-
  - Linear combination contains all information from the 2 original signals. All details will be reflected in the combined signals. Even trivial fluctuation will have effects on a combined signal.
  - Under the maximum combination, we can see that the curve consists of fragments exactly came from one of the 2 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.
- The **ERD_GA** algorithm is for "**global**" optimization.

- From a population $\{\theta^{(1)}, \dots, \theta^{(n)}\} \longrightarrow$ a new population:
  - "select" parents;
  - "crossover" parents $\longrightarrow$ a new offspring;
  - "mutate" the new offspring, to be placed in a new population.
- For (12) and Figure 7, $p_s = \sum_{k=1}^{s} \sum_{j=k}^{n-1} d_j/D$, $s = 1, \dots, n-1$,

$$\begin{aligned} p_s &= \sum_{k=1}^{s} \sum_{j=k}^{n-1} d_j/D \\ &= \sum_{k=1}^{s-1} \sum_{j=k}^{n-1} d_j/D + \sum_{k=s}^{n-1} d_j/D = p_{s-1} + \sum_{k=s}^{n-1} d_j/D, \\ p_s - p_{s-1} &= \sum_{k=s}^{n-1} d_j/D = \sum_{k=s}^{n-1} (L_{j+1} - L_j)/D = (L_n - L_s)/D, \end{aligned} \tag{41}$$

so the smaller loss L_s, the better fitness value, the larger selection probability $(p_s - p_{s-1})$.

- For the use of PCA: Figure 9 plots $\{\frac{\sum_{j=0}^{k-1}\lambda_{(p-j)}}{\sum_{j=0}^{p-1}\lambda_{(p-j)}} : k=1,\dots,p\}$, where $\lambda_{(1)} \leq \cdots \leq \lambda_{(p)}$ are eigen-values of $\widehat{\Sigma} \in \mathbb{R}^{p\times p}$, and $\widehat{\Sigma}$ is the sample covariance-matrix of the data matrix $\mathbf{X} \in \mathbb{R}^{p\times m}$. Here

$$
\begin{aligned}
\frac{\sum_{j=0}^{k-1}\lambda_{(p-j)}}{\sum_{j=0}^{p-1}\lambda_{(p-j)}} &= \frac{\sum_{j=0}^{k-1}\lambda_{(p-j)}}{\lambda_{(p)}+\cdots+\lambda_{(1)}} \\
&= \begin{cases} \frac{\lambda_{(p)}}{\lambda_{(p)}+\cdots+\lambda_{(1)}}, & \text{if } k=1, \\ \frac{\lambda_{(p)}+\lambda_{(p-1)}}{\lambda_{(p)}+\cdots+\lambda_{(1)}}, & \text{if } k=2, \\ \cdots & \cdots \\ 1, & \text{if } k=p, \end{cases} \\
&= \frac{\text{cum\_sum}([\lambda_{(p)},\dots,\lambda_{(1)}])}{\text{sum}([\lambda_{(p)},\dots,\lambda_{(1)}])}.
\end{aligned}
$$

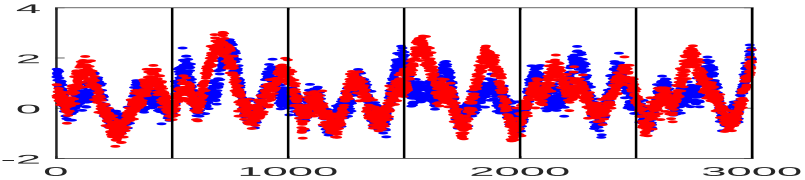
Note that here the sample covariance matrix $\widehat{\Sigma} \in \mathbb{R}^{p\times p}$ for the data matrix $\mathbf{X} \in \mathbb{R}^{p\times m}$ typically performs OK if $p \ll m$, which holds for the EEG data (with $p \approx 30$ channels and $m \approx 1000$ times points).

- Summary of the algorithm for **MaxICA**:

Table 1: *Summary of the 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**[1] to each piece respectively.

[1] **ERD_GA** algorithm with **E**lite weighted sum selection, **R**andom combined crossover, **D**ynamic mutation. The stopping criterion is: no better solution can be found for 10,000 iterations.



- For the real “Visual processing data”: The file names of 3 datasets have the format,

a	ff	b
− − − − − − − −		− − − − − − − −
cba1: subject 1		01: **categorization**
ega1: subject 2		03: **recognition**

- For the real “Epilepsy data”: Andrzejak *et al.* (2012) mentioned “We only included signals that where measured from neighboring contacts. EEG signals measured from neighboring contacts are frequently but not necessarily strongly correlated”.

Flowchart of GA

0. Generate an initial population $\{\widehat{\boldsymbol{\theta}}^{(1)}, \ldots, \widehat{\boldsymbol{\theta}}^{(n)}\}$ of “individuals”, each representing a possible (coded) solution; compute the fitness value $\mathfrak{f}(\widehat{\boldsymbol{\theta}}^{(1)}), \ldots, \mathfrak{f}(\widehat{\boldsymbol{\theta}}^{(n)})$ of each individual.
1. If the stop condition is not met, produce a new population. Cycle through each of $[n/2]$ pairs,
   - “**Select**” 2 individuals randomly from the old generation (via a scheme favoring the more fit individuals) for mating.
   - “**Crossover**” the 2 selected individuals to give 2 offspring. Compute the fitness values of the 2 offspring.
   - “**Mutate**” the 2 offspring individually, and insert in the new generation.

   If the population has converged (the “average” fitness will approach that of the “best” individual, i.e., $\frac{1}{n}\sum_{s=1}^{n} \mathfrak{f}(\widehat{\boldsymbol{\theta}}^{(s)}) = \max_{1 \le s \le n} \mathfrak{f}(\widehat{\boldsymbol{\theta}}^{(s)})$), finish; otherwise, repeat Step 1.

Terminologies from GAs

- David E. Goldberg's *Genetic Algorithms in Search, Optimization and Machine Learning* is by far the bestselling introduction to genetic algorithms. Goldberg is one of the preeminent researchers in the field–he has published over 100 research articles on genetic algorithms and is a student of John Holland, the father of genetic algorithms–and his deep understanding of the material shines through. The book contains a complete listing of a simple genetic algorithm in Pascal, which C programmers can easily understand. The book covers all of the important topics in the field, including crossover, mutation, classifier systems, and fitness scaling, giving a novice with a computer science background enough information to implement a genetic algorithm and describe genetic algorithms to a friend.

- CCO: "combined crossover operator"
- a population of solutions, called chromosomes,
- crossover operator:
  - combines 2 chromosomes (parents) to produce a new chromosome (child)
  - operates on 2 parents and has no self adaptation properties
  - works with 2 parent solutions and creates the offspring.
- the single-point crossover operator on binary strings.
- mutation rate equal to 0.01; crossover rate varies for each run
- the ability of **GA** for not to be trapped in the local optima
- the best fitness value; the worst fitness value;

- Operators in **GA**:
  - "crossover": produces offspring by representing the information from two parents.
    combines 2 chromosomes (parents) to produce a new chromosome (child).
    Hassan (1995) paper has some explicit formulas to compute a "child" from "2 parents".
  - "mutation": prevents convergence of the population by flipping a small number of randomly selected bits to continuously introduce variation.

List of questions

- How to interpret results/colors of brain images in Figures in the original Example 4?
  Answer: Yes, on the left hand side, 5 plots generated at 5 time points, where each of these 5 plots gives the brain image of 5 components. It's same on the right hand side, 5 plots for SUM (MAX) generated at the same time points.
- Figures 25 and 27: what are the roles of the top and bottom panels?
  Answer: for 2 observed signals.
- Is the "genetic algorithm" related to the "bootstrap" method?
- Parameters in (9) exclude $\{N, n_k\}$, since
  - N will be selected by PCA. For simulation studies 1 and 2, $N = 5$; For real data 1, $N = 5$, and for real data 2, $N = 2$.
  - For n_k, the paper suggests $n_1 = \cdots = n_N = 5$ in the numerical work.

- The paper mentioned the “location offset parameter” γ to be included for estimation. Where (i.e., name of which code, and which part) could we see the estimation of γ in the code?
  R.S. Guo’s Answer: For example, in “`maxica_simu.m`”, line 81, there is a “+1”, line 108, the lower bound and upper bound “lb3”, “ub3”, and in function “`frealga.m`”, line 33, there’s ”`-x(npar)`”.
- For the “identifiability” issue, does **MaxICA** preserve the orders and scalings of the components?
  Answer: Zhengjun said “No”, since in (7),
  $\max\{b_{j,1}\, S_1(t), \dots, b_{j,N}\, S_N(t)\} = \max\{b_{j,k_1}\, S_{k_1}(t), \dots, b_{j,k_N}\, S_{k_N}(t)\}$ for any order $k_1, \dots, k_N$, which permutes $1, \dots, N$.
- In the section on the convergence of **ERD_GA**, the paper uses the inverse choice $\mathfrak{f}_s = 1/L_s$, which is actually for the “Roulette Wheel Selection” (RWS), not for EWSS. My question: whether the negative choice $\mathfrak{f}_s \propto (L_n - L_s)$ as in (41) will be valid for EWSS?

- Our loss function $L(\boldsymbol{\theta}) \overset{(9)}{=} \sum_{j=1}^{p} \|\vec{\boldsymbol{X}}_j - \vec{\boldsymbol{f}}_j(\boldsymbol{\theta})\|_2$ (without taking squares). Why not use $\sum_{j=1}^{p} \|\vec{\boldsymbol{X}}_j - \vec{\boldsymbol{f}}_j(\boldsymbol{\theta})\|_2^2$ (taking squares)? Does taking squares seem to be easier or harder for optimization? Note

$$\sum_{j=1}^{p} \|\vec{\boldsymbol{X}}_j - \vec{\boldsymbol{f}}_j(\boldsymbol{\theta})\|_2 = \sum_{j=1}^{p} \sqrt{\sum_{i=1}^{m} \{X_j(t_i) - f_j(\boldsymbol{\theta}, t_i)\}^2},$$

is not-separable w.r.t. time points t_i, but

$$\begin{aligned} \sum_{j=1}^{p} \|\vec{\boldsymbol{X}}_j - \vec{\boldsymbol{f}}_j(\boldsymbol{\theta})\|_2^2 &= \sum_{j=1}^{p} \sum_{i=1}^{m} \{X_j(t_i) - f_j(\boldsymbol{\theta}, t_i)\}^2 \\ &= \sum_{i=1}^{m} \sum_{j=1}^{p} \{X_j(t_i) - f_j(\boldsymbol{\theta}, t_i)\}^2 \\ &= \sum_{i=1}^{m} \|\boldsymbol{X}(t_i) - \boldsymbol{f}(\boldsymbol{\theta}, t_i)\|_2^2, \end{aligned} \tag{42}$$

is separable w.r.t. time points t_i, where $\boldsymbol{f}(\boldsymbol{\theta}, t_i) = (f_1(\boldsymbol{\theta}, t_i), \ldots, f_p(\boldsymbol{\theta}, t_i))^T \in \mathbb{R}^{p \times 1}$.

- R.S. Guo's Answer: I use the L2-norm distance, i did't try the one without square, if there's no square, the scores just will be very large, but i don't know how much would this affect the code.

To-do list for the paper

- Try new results using the loss function (42) to replace (9).
- Proof of convergence part of **ERD_GA**:
  The proof (by R.S. Guo) was based on Markov chain, which is for finite discrete-states. But, the **ERD_GA** is for continuous-states. That proof was removed from the paper to the Appendix (to be further checked).
  - Convergence of the classical **GA**: with **discrete** fitness values, proved in Bhandari, Murthy, and Pal (1996).
  - Convergence of our **ERD_GA**: in the **continuous** case, discusses (but may not be valid, and thus is excluded from the paper)
    - classification of strings,
    - partitioning of populations,
    - transitions between two populations,
    - convergence of **ERD_GA** algorithm based on the Markov chain.

- Our current set-up for numerical studies considers the case of

  $$\text{\# of data points} \quad \gg \quad \text{the \# of parameters.}$$

  For the case of

  $$\text{\# of data points} \quad \ll \quad \text{the \# of parameters,}$$

  some penalized methods may be used. So what about the penalized estimation of $\boldsymbol{\theta}$ in $\mathbf{MaxICA}$ in the high dimensional case?
- What about replacing the “maximum” in (4) for $\mathbf{MaxICA}$ by taking “quantiles” $\tau \in (0, 1)$?
- Use a non-linear “functional regression model” for the $\mathbf{MaxICA}$ model, and estimate the component functions $S_k(t)$ in (6) via non-parametric method.
- Allow dependency such as ARMA models for auto-correlated noise terms.

- Other than PCA, how to determine the number N of components?
  I asked Bowen Zhang to think about.
- Other algorithms for parameter learning: Bayesian method?
- From Yongfeng's presentation, it seems that
  "tensor model decomposition", PCA and **ICA** have some connections.
  See also connections with "factor models" (in panel data models).

Some comparisons/conclusions made so far

- Both **ICA** and **MaxICA** methods focus on separating "hidden components" $S_1(t), \ldots, S_N(t)$ from "observed signals" $X_1(t), \ldots, X_p(t)$, but in general, can not uniquely identify the original ordering of components $S_1(t), \ldots, S_N(t)$.
- The **ICA** model (2):
  - requires component variables to be (a) mutually independent, (b) jointly non-Gaussian, and (c) assumes $N \leq p$ (and typically $N = p$);
  - assumes non-structured components and excludes noise terms, oftentimes suffering from the "overfitting" problem as seen in Figures 6, 15, 17, 19 in simulations, 25, 28, 30 in visual processing data, 32, 34, 36, 38 in Epilepsy data.
  - uses the FastICA algorithm (`http://research.ics.aalto.fi/ica/fastica/`).

- The **MaxICA** model (4):
  - removes these 3 constrains, as seen in Figure 4, which allows $N > p$, with $N = 2$ and $p = 1$;
  - embeds structured components and incorporates noise terms with either known or unknown distributions, ameliorating such "overfitting" problem;
  - uses the **ERD_GA** algorithm, which outperforms **classical_GA** and **SA**.
    - For "`simu_example_1.m`", where data come from the **MaxICA** model, the "**ERD_GA**" is as fast as the "**classical_GA**" and "**SA**", but the "**ICA**" method/algorithm nearly diverges (i.e., converges very slowly).
    - For the "visual processing data", "**ERD_GA**" is much faster than "**SA**" and fits better.

- Bounds of intervals for searching parameters in θ:
  - in "**MaxICA**": needed in all 3 algorithms, "**ERD_GA**", "**classical_GA**" and "**SA**",
  - in "**ICA**": not needed.
- Numerical comparison of **MaxICA** and **ICA**:
  - In simulation examples: distinctions are clear.
  - For the visual processing data: distinctions are clear.
    For the Epilepsy data: **MaxICA** and **ICA** give similar results in recovery of hidden components and fits of observed signals.