INTERNSHIP REPORT

Multi-view learning with shared but delayed sources

Ambroise Heurtebise

Supervised by Pierre Ablin, Alexandre Gramfort and Bertrand Thirion

October 12, 2022

Abstract

Magnetoencephalography (MEG) and electroencephalography (EEG) are two ways of measuring the neuronal activity of the brain. The data obtained from these methods are composed of hundreds of signals, each signal corresponding to a specific location on the surface of the head. In this context, ICA is a widely used algorithm that models observed signals as a linear combination of sources of interest inside the brain. In order to draw general conclusions about brain functional organization, we need data from multiple subjects. However, given that two subjects don't have the same brain topography and that they can move during the experiment, their signals don't necessarily match. To summarize, we would like to use an algorithm that can extract a few signals of interest, that we call sources, from the data of the subjects. At Inria Saclay, they have recently developed such an algorithm, called MultiView Independent Component Analysis (MVICA) [Richard et al., 2020], that assumes that all the subjects share common sources. The subjects' data are modeled as a linear combination of common, shared independent sources plus noise. Compared to classical ICA, this model gives better source reconstruction. Also, this algorithm takes advantage of the fact that there are many subjects, compared to many other ICA algorithms that apply separately to individual subjects and average their results. But the assumption of MVICA that they share common sources is quite strong and biologically implausible in some scenario, so we would like to relax it. Many articles showed that a delay could appear in some subjects' data, for example due to age or autism. So, we chose to relax the assumption by considering that the common sources could vary in time for different subjects. As the likelihood of MVICA model is available in closed form, adding a delay parameter only forces us to modify this likelihood. We propose two solutions in order to take into account the subjects' delays, calling the resulting algorithm MultiView Independent Component Analysis with Delays (MVICAD). The first solution relies on the optimization of the likelihood with respect to the delays. At each iteration of MVICA, we add a step that aims to find the best delays. The second solution is simpler: since MVICA begins with an initialization step, it only consists in fitting the best delays possible during this initialization step, i.e. before the likelihood optimization. As the two solutions apply at different places in the algorithm, they can be merged. In order to evaluate our model, we generate synthetic data and try to retrieve the sources. Then, we measure the quality of the recovered decomposition. We show that the new algorithm significantly outperforms the former MVICA when temporal variability is added in the model. Actually, it even recovers the correct sources when they are shifted arbitrarily. Finally, we test our model on MEG data and discuss about the differences with MVICA.

Plagiarism Integrity Statement

I hereby confirm that the present paper is the result of my own scholarly work, and that in all cases material from the work of others (in books, articles, essays, dissertations, and on the internet) is acknowledged, and quotations and paraphrases are clearly indicated.

CONTENTS

1 Introduction

1.1 Motivations

The internship's goal is to improve an already existing algorithm called MultiView Independent Component Analysis (MVICA). This algorithm has shown extremely good results on neuroimaging group studies both based on fMRI and MEG data (these data types will be explained in section 1.2). As a reminder, ICA decomposes observed signals as a mixture of independent sources, and MVICA precisely demonstrated improved performance in identifying these independent sources.

More generally, ICA algorithms are widely used in various domains and particularly in neuroscience. They are used to identify sources but also to remove artefacts (such as eye blinks, from EEG data) [Delorme et al., 2007], model receptive fields of primary visual neurons [Bell and Sejnowski, 1997], predict decision-making using EEG [Douglas et al., 2013], sort neuronal spikes [Lewicki, 1998] or study fMRI data of a cohort of subjects [Calhoun et al., 2001a].

One direction in which MVICA can be improved is that of individual subject variability: indeed, MVICA assumes that the exact same sources are shared among subjects. Yet, neural activity of different subjects sometimes vary in time. In other words, after a common stimulus, neural activity of subject 1 can be approximately equal to neural activity of subject 2, up to a time shift. Unfortunately, MVICA doesn't take into account this time shift and its source reconstruction can suffer from that.

The subject of the internship is thus to adapt MVICA by taking these time delays into account.

1.2 Background on neuroscience time series

MVICA processes sequences of time series and could hence be employed in many different fields: neuroscience, genomics, astrophysics, finance or computer vision for instance. In our research, we focus on neuroscience. A starting point will then be to dive into the input data that one can use in this context. Many methods exist to monitor and measure brain activity non-invasively. The three main methods are electroencephalography (EEG), magnetoencephalography (MEG) and functional Magnetic Resonance Imaging (fMRI). ICA methods are routinely used to process each of these modalities but, during the internship, we mainly used MEG data.

Electroencephalography (EEG) EEG [Teplan et al., 2002] measures the electrical activity generated by the human brain. These electrical signals come from gray matter regions, more specifically from the dendrites of pyramidal neurons.

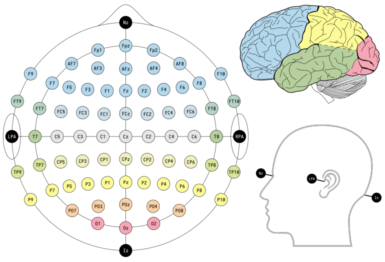


Figure 1: EEG electrodes.

Whenever a large amount of neurons are firing up in a synchronized pattern, changes in the electric potential are strong enough to be recorded with EEG electrodes on the scalp surface. These electrodes are usually placed as in figure 1. Many devices have fewer sensors than in the figure, but the idea is the same.

It is important to know that EEG is not an absolute voltage but rather a referential recording. It always represents a relative increase or decrease in electric potential at a specific location. Indeed, the voltage is recorded using a differential amplifier that takes two electrodes as input and measures the electric potential difference between them. This device also amplifies the signals, which come from very low amplitude. As we have to compare two electrodes in order to get one signal, various methods exist about the choice of the two electrodes. We call these methods montages, the principal montages being the bipolar montage (each electrode is compared to the following one in a specific order) and the common reference montage (this common reference is typically located behind the ears). The choice of the montage is to be made by a technician. When applying ICA to EEG, a common reference montage is typically used.

The changes in electric fields occur very fast, so EEG has a very high time resolution (up to 1 millisecond, depending on the sampling rate). This excellent time resolution gives insights on the precise timing of brain processing.

EEG has other advantages: it is cost-efficient (its price vary between $200 and $100.000) and mobile (so it allows to record brain processes outside of laboratory environments).

Magnetoencephalography (MEG) As neurons produce electrical potentials when firing up, they also generate magnetic fields. The role of MEG [Cohen, 1968] is to capture these magnetic fields in order to isolate regions of the brain which contain a large amount of active neurons at a specific time. Importantly, the origin of the signals of EEG and MEG is the same.

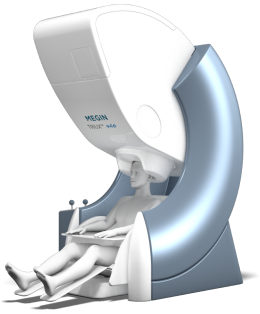


Figure 2: MEG device.

The main problem for MEG lies in the fact that magnetic fields induced by neurons are extremely low (approximately a millionth the strength of the Earth's magnetic field). In order to measure it, we then have to isolate these tiny magnetic field changes coming from the brain from ambient magnetic noise.

This is done by entering a chamber made from many magnetic shields. Then, the head is placed in a box similar to figure 2, that contains liquid helium at a very low temperature and around 300 magnetic sensors called SQUIDS. This complex device justifies the high price of MEG machines (about $1.5 to 2 million).

Also, MEG devices are completely static (similar to fMRI devices). They require the participant to lay or sit almost motionless and keep movements to a minimum.

The biggest advantage is that MEG combines the high temporal resolution similar to EEG, with a higher spatial resolution, hence allowing to locate more precisely the position of active neurons.

Figure 3: fMRI device.

Functional Magnetic Resonance Imaging (fMRI) fMRI [Racine et al., 2005] measures brain function by detecting changes in blood flow associated with neural activity. The idea is that neurons need more oxygen when they're active, so cerebral blood flow and neuronal activation are coupled.

The device used to detect changes in blood flow creates a strong magnetic field, emits bursts and retrieve the magnetic waves after they passed through the blood. This method allows to differentiate oxygen-rich and oxygen-poor blood, that is active and less active regions of the brain. During the experiment, patients have to lay motionless in a magnetic core while the radio frequency bursts are emitted, as in figure 3.

fMRI's strength is its excellent spatial resolution. However, it is a relatively slow neuroimaging method compared to EEG or MEG. Indeed, fMRI recovers brain images at each radio frequency bursts emission, that is approximately every second. This is far slower than EEG or MEG.

Dataset Various massive neuroimaging datasets exist, containing data from hundreds of participants. During the internship we used the Cam-CAN Stage 2 cohort study [Taylor et al., 2017]. Cambridge Centre for Ageing and Neuroscience (Cam-CAN) is a large-scale collaborative research project at the University of Cambridge. The Cam-CAN repository that we use contains data from a large (approximately $N = 700$) population-based sample. These patients are aged between 18 and 87 years old and the study goal is to characterise age-related changes in cognition and brain structure and function. For each patient, we look at their MEG data. These data are composed of 306 signals (as much as the MEG sensors) which last a few minutes, and were collected over three separate sessions. In the first session, participants had to rest with eyes closed. The second session is a sensorimotor task, meaning that patients were given audio-visual stimuli and had to manually respond to these stimuli. The third session is called audio-visual task, meaning that they were given audio-visual stimuli but didn't have to give manual response. This latter task is therefore purely passive. We used the data from this last session in our real data plots. Typically, these data look like figure 4. In this figure, we only showed a few sensors and not all of them. Moreover, the time window only represents a few seconds. Also, we can observe that there are many redundancies between channels, and that they are quite noisy. For these reasons, MEG signals can be hard to interpret for a non expert. This motivates the use of an algorithm like ICA, that extracts a simpler representation of the data.

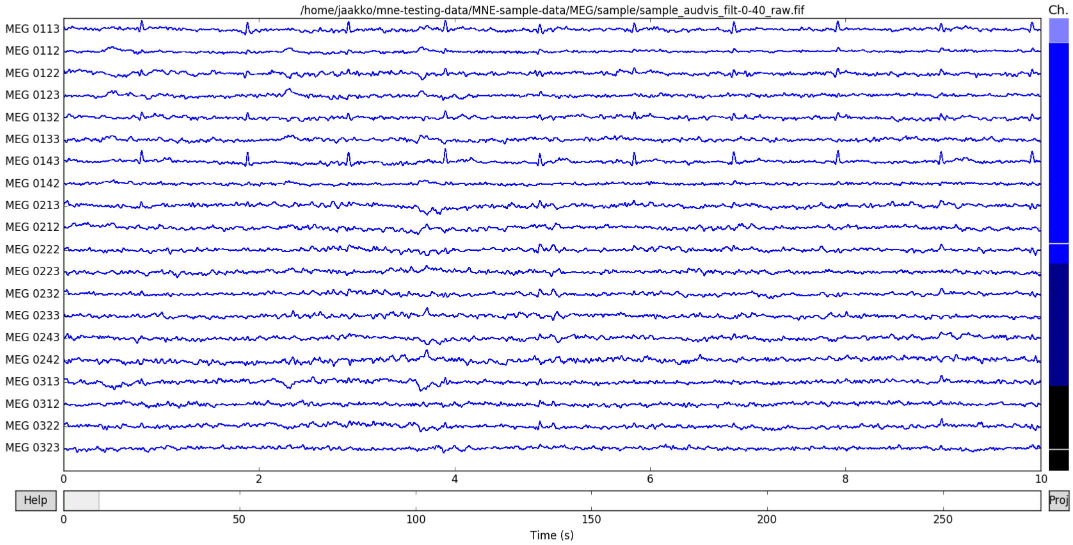


Figure 4: Typical MEG data.

1.3 From ICA to MVICA

This section is inspired by Hyvärinen and Oja [2000]. Its purpose is to give intuition about why ICA is a method of choice in neuroimaging, and how it applies to multi-subjects data.

We take the same example as in their paper. This is known as the cocktail-party problem. Imagine that you are in a room where two people are speaking simultaneously. You have installed two microphones, at different locations in the room. The microphones give you two recorded time signals, which we could denote by $x_1(t)$ and $x_2(t)$, with x_1 and x_2 the amplitudes, and t the time index. On the other hand, the speech signals emitted by the two speakers are denoted by $s_1(t)$ and $s_2(t)$. Each of the recorded signals is a weighted sum of the speech signals. Thus, we obtain a set of linear equations:

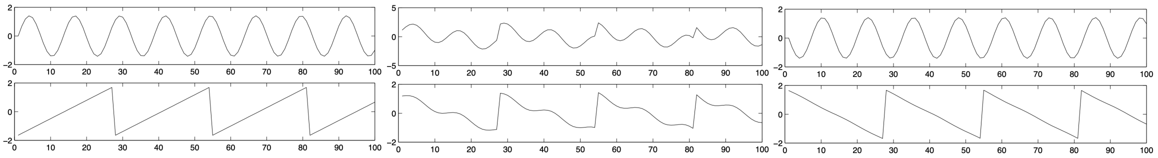
$$x_1(t) = A_{11}s_1 + A_{12}s_2 \tag{1}$$

$$x_2(t) = A_{21}s_1 + A_{22}s_2 \tag{2}$$

where A_{11}, A_{12}, A_{21}, and A_{22} depend on the distances between the microphones and the speakers. Usually, we are only interested in the speech signals, instead of a mix of speech signals. Thus, we would like to estimate the two original speech signals $s_1(t)$ and $s_2(t)$, using only the recorded signals $x_1(t)$ and $x_2(t)$.

We use images from the aforementioned paper to illustrate our example. Figure 5a represents the original signals (although not realistic) and figure 5b the observed signals. The problem is to recover the data in figure 5a using only the data in figure 5b. Of course, if we knew the parameters A_{ij}, then we could solve (1) and (2) easily. The point is, however, that if you don't know the A_{ij}, the problem is considerably more difficult.

The approach of ICA to solving this problem is thus to estimate the A_{ij} at first. In order to do that, we use some information on the statistical properties of the signals $s_i(t)$. Actually, it suffices to assume that $s_1(t)$ and $s_2(t)$, at each time instant t, are statistically independent. In many scenarios, this assumption can be realistic, and in practice it do not need to be exactly true. In this context, Independent Component Analysis (ICA) can be used to estimate the A_{ij} based on the information that the source signals are independent. When the A_{ij} are estimated, retrieving the two original source signals $s_1(t)$ and $s_2(t)$ from their mixtures $x_1(t)$ and $x_2(t)$ becomes easy. Figure 5c shows the two signals estimated by the ICA method. We observe that they are very close to the original source signals, although their signs are reversed. In practice, their order could have been changed too. However, this is not important here since order has no significance and true signs can be found visually in many situations.



(a) Original signals. (b) Observed mixture. (c) Estimates of the original signals.

Figure 5: Example of ICA with two sources.

Originally, ICA was developed to deal with this kind of problem. But since the increase of interest in this method, many other applications, such as feature extraction, have been found.

In our context of neuroscience time series, ICA applies perfectly. Indeed, consider for example the case of MEG. The observed data is the amplitude of magnetic fields at the different locations of the scalp surface. However, the origin of the signals do not come from the scalp level, but rather inside the brain. We can assume that the observed signals are generated by mixing some underlying components of brain activity. This situation is quite similar to the cocktail-party problem: we would like to find the original components of brain activity, but we can only observe mixtures of the components. ICA can reveal interesting information on brain activity by giving access to its independent components.

As said before, ICA can also be used to perform feature extraction. With neuroscience time series, this is very useful because it allows to remove artefacts, such as eye blinks, from brain signal recordings. Indeed, if we extract sources from observed signals and clearly recognize that one of the sources represents these eye blinks, then it suffices to remove this particular source and multiply by the a_{ij} to get cleaner signals.

So, ICA can reveal information about brain activity when applying it to a single-subject data. However, when the objective is to draw conclusions about brain functioning in general, several subjects are needed. Having more subjects boosts statistical power and allows to find similarities and differences between subjects.

However, classical methods are often poorly suited to account for multiple views of the same sample (each view corresponding here to a subject's data). Consequently, many multiview learning methods have been developed to take advantage of multiple data views and produce better results [Sun, 2013].

In the context of neuroimaging, these group-studies are difficult since data coming from multiple subjects presents a large variability in anatomy, functional topography and stimulus response.

MultiView Independent Component Analysis (MVICA) [Richard et al., 2020, 2021] is one of these group studies. It models each subjects' data as a linear combination of common, shared independent sources plus noise. This model takes advantage of the group structure of the data, but assumes that the sources are perfectly identical and aligned between subjects, hence so far ignoring differences such as the temporal variability in the neural responses of each subjects.

1.4 Adding subject-specific delays

Assume that, after a common stimulus, every subject has approximately the same neural activity, up to a time shift. This assumption makes sense in a neuroscience because time response can differ between people, due to age [Price et al., 2017] or autism [Roberts et al., 2010] for example. This phenomenon is illustrated in figure 6. Mathematically, it only means that the independent sources are the same for all subjects, up to a time shift. In this case, the MVICA model does not apply anymore, and needs to be extended.

We will propose later two solutions to this problem. The first one will be implemented during the algorithm's core optimization, and the second one during the algorithm's initialization. Both solutions can in practice be combined.

The goal of figure 6 is to demonstrate that some delay can appear in the Cam-CAN dataset. To produce this figure, we took the 618 subjects of the Cam-CAN dataset and removed those who had too noisy signals. Then, we computed the norm of their multidimensional time series at each time point. The resulting curve is usually peak-shaped, as in the figure. We split the dataset between people for whom the peak was very early, and those for whom the peak was very late. This procedure allows to separate people within two cohorts: those with early neuronal response and those with later neuronal response. Finally, we applied MVICA independently on both cohorts and kept one of the sources after a visual inspection. This source is shown in figure 6, for both cohorts. We clearly observe that the source doesn't peak at the same time for the two cohorts.

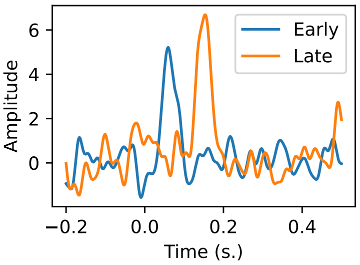


Figure 6: Same source from two different cohorts of subjects.

1.5 Inria Saclay

The internship was done at Inria Saclay Centre, which was established in 2008. It is a founding member of Paris-Saclay University and a member of the Institut Polytechnique de Paris. The centre has 35 research teams, including the MIND (Models and Inference for Neuroimaging Data) team in which I worked. This team is actually led by Alexandre Gramfort. With Pierre Ablin and Bertrand Thirion, they supervised my internship.

MIND team is supported by CEA and affiliated with NeuroSpin, the largest neuroimaging facility in France dedicated to ultra-high magnetic fields.

The team is focused on core methodological developments in: Machine learning for inverse problems, Heterogeneous data & knowledge bases, Statistics and causal inference in high dimension and Machine Learning on spatio-temporal signals. The team is also interested on more real problem and its main application domains are: Population modeling, large-scale predictive modeling, Mapping cognition & brain networks, Modeling clinical endpoints using multimodal neuroimaging techniques and Quantitative biology and physics using brain images and bio-signals.

2
RELATED WORKS

2.1 Single-view ICA principle

This section is inspired by the chapter 1.4 of Ablin [2019]. As we saw previously, the input data that we use are high-dimensional: the cohort is composed of many subjects and each subject produces 306 signals of a few minutes. Moreover, MEG detects magnetic activity at the scalp level and not at the source level inside the brain. Therefore, after calling a dimension reduction algorithm (for example PCA), we want our algorithm to isolate the sources mentioned above.

ICA [Shlens, 2014, Hyvärinen and Oja, 2000] is an unsupervised method for separating a multivariate signal into additive subcomponents. The main assumptions are that at most one subcomponent is a non-Gaussian signal and that the subcomponents are statistically independent from each other. For now, let's consider that we only have one subject. Mathematically speaking, assume that we observe samples $x \in \mathbb{R}^p$ which are a linear combination of sources $s \in \mathbb{R}^p$:

$$x = As \ ,$$

where $A \in \mathbb{R}^{p \times p}$ is called the mixing matrix, and $s \in \mathbb{R}^p$ are the independent sources. The goal is to find A and s without any further information. Given a dataset $X = \{x_1, \dots, x_n\} \in \mathbb{R}^{p \times n}$ of n samples, ICA consists in factorizing the matrix X as:

$$X = AS \ ,$$

where $S \in \mathbb{R}^{p \times n}$ is the source matrix. In our case, the matrix $X \in \mathbb{R}^{p \times n}$ represents the EEG signals after applying a dimensionality reduction. Typically, $p \approx 20$ and $n \approx 400$.

When $n \geq p$, the matrix S has more parameters than A. Thus, it is easier to search for A and then we will have $S = A^{-1}X$ (if we assume that A is invertible, as we do). Indeed, ICA model is completely identified if we estimate either A or S.

Furthermore, ICA model is identifiable. Identifiability has to be understood as the following: if you observe $X \in \mathbb{R}^{p \times n}$ where the number of samples n tends to infinity, then there is only one matrix $S \in \mathbb{R}^{p \times n}$ of independent rows that verifies $X = AS$, for a certain mixing matrix $A \in \mathrm{GL}_p$. For ICA it is almost true, as this matrix $S \in \mathbb{R}^{p \times n}$ is only unique up to a scale and permutation. Indeed, if $X = AS$ and $\Lambda \in \mathbb{R}^{p \times p}$ is a scale-permutation matrix, then we also have

$$X = \underbrace{A\Lambda}_{A'} \underbrace{\Lambda^{-1}S}_{S'} \ ,$$

with S' of independent rows.

Theorem 1 (Identifiability of ICA, thm. 11 of Comon [1994], based on Darmois [1953])**.** *Assume that s contains independent entries, of which at most one is Gaussian. Let $\Lambda \in GL_p$, such that $s' = \Lambda s$ also has independent entries. Then, Λ is a scale-permutation matrix.*

Consequently, if we find a matrix S of independent rows and such that $X = AS$, then we know that this matrix unique, up to scale and permutation. In practice, estimation is done by finding a matrix $W \in \mathrm{GL}_p$ (called unmixing matrix) such that the vector $y = Wx$ is approximately equal to s. Ideally, we want $W = A^{-1}$. As the only information we have about s is that its rows are independent and non-Gaussian, the estimation boils down to finding a matrix $W \in \mathrm{GL}_p$ such that the rows of the vector $y = Wx$ are as independent and non-Gaussian as possible. Several methods exist in order to do that: maximization of nongaussianity [Hyvärinen, 1997a], minimization of mutual information [Hyvärinen, 1997b], maximum likelihood estimation [Pham et al., 1992], etc. During the internship, we focus on the maximum likelihood estimation method.

In order to express this likelihood, we first have to introduce the density of the sources, d. Since the sources are assumed independent, we can write that $d(s) = \prod_{i=1}^{p} d_i(s_i)$ for some density functions d_i that are unknown.

In practice, we do not need to know the density of the sources. Indeed, as explained in Jung et al. [1997], we can only assume that d verifies:

$$-\frac{d}{dy}\log(d_i(y)) = y \pm \tanh(y) \ .$$

The choice between $+$ and $-$ in this last expression is complex. It depends on the fact that sources are super- or sub-Gaussian, and these notions are hard to formalize. Generally, the kurtosis is negative for sub-Gaussian densities (distributions flatter than Gaussian) and is positive for super-Gaussian densities (sharper than Gaussian). Usually, we assume that brain sources are super-Gaussian. Thus, the densities d_i are fixed, and we assume that, for $i = 1, \dots, p$, we have $s_i \sim d_i$.

Finally, it can be shown [Pham et al., 1992] that, for a dataset $X \in \mathbb{R}^{p\times n}$, the negative log-likelihood is:

$$\mathcal{L}(W) = -\log|W| - \frac{1}{n}\sum_{i=1}^{p}\sum_{j=1}^{n}\log(d_i([WX]_{ij})) \ , \tag{3}$$

where $|W|$ is the absolute value of the determinant of matrix W. ICA algorithms that rely on maximum likelihood estimation, such as Picard [Ablin et al., 2018], try to minimize this last expression with respect to W.

The most used ICA algorithm is FastICA [Hyvärinen and Oja, 2000, Hyvarinen, 1999].

2.2 Methods for ICA with multiple views

Let's go back to the case where we have different subjects in our MEG study. Assume that the number of subjects is m.

Let us write $\mathbf{X} = \{X^1, \dots, X^m\} \in \mathbb{R}^{m\times p\times n}$. For each subject $i = 1, \dots, m$, the goal is to find A^i and S^i such that:

$$X^i = A^i S^i \ .$$

We could apply the ICA algorithm as described previously separately on each subject, but this would give us m different mixing matrices A^i and m different source matrices S^i. As our final objective is to capture information about brain structures and functions in general, and not to a specific subject, we would like our matrices A^i or S^i to share information. Several methods have been proposed. We summarize here the characteristics of some of the most commonly used ones. This section is greatly inspired by Richard et al. [2020].

Group ICA Group ICA algorithms are often decomposed in three steps. First, we perform a dimensionality reduction on each subjects' data separately. Second, the reduced data are merged using PCA [Calhoun et al., 2001a] or multi set CCA [Varoquaux et al., 2009]. Third, an ICA algorithm for shared source extraction is applied to the merged data. The SR-ICA approach of [Zhang et al., 2016] is such a method, and looks like MVICA.

Likelihood-based models MVICA assumes that subjects share common sources and optimizes the closed-form likelihood derived from this model. If we do not make this assumption and consider that subjects have different sources [Guo and Pagnoni, 2008], then the likelihood formula becomes more complex to evaluate. Thus, it is minimized with the Expectation-Maximization (EM) algorithm, which converges slowly and unreliably [Bermond and Cardoso, 1999, Petersen et al., 2005]. Our closed-form likelihood can be minimized more efficiently than with the EM algorithm.

Structured mixing matrices Just as we imposed common sources, one can also impose some constraints on the mixing matrices. These matrices can be interpreted as topographic brain maps. Indeed, in the example of section 1.3, the parameters A_{ij} from the model $x_i = \sum_{j=1}^{p} A_{ij}s_j$ corresponded to the distance between the speakers and the microphones. In our context, these parameters are not distances but still contain information about brain topography. So, making particular assumptions about the matrix A can be plausible. MVICA only assumes that the mixing matrices are invertible. Some other approaches impose additional constraints. For instance, the Shared Response Model [Chen et al., 2015] (SRM) assumes orthogonality of the mixing matrices, although it may not be realistic. Tensorial methods [Beckmann and Smith, 2005] assume that the mixing matrices are the same up to diagonal scaling. Other methods impose a common mixing matrix [Cong et al., 2013, Grin-Yatsenko et al., 2010, Calhoun et al., 2001b, Monti and Hyvärinen, 2018].

Matching sources a posteriori Perhaps the most naive procedure would be to perform ICA independently on each subjects' data and to average their estimated sources. However, even if sources of different subjects can match, they are not necessarily in the same order and could have different signs. In order to align all the sources, PermICA [Richard et al., 2020] choose a reference subject and tries to match the sources of all other subjects to the sources of the reference subject. This matching is done using the Hungarian algorithm [Tichavsky and Koldovsky, 2004]. The process is repeated multiple times, using the average of previously aligned sources as the new reference. Importantly, PermICA is used as an initialization step, at the beginning of MVICA.

Deep Learning Deep Learning methods, such as convolutional auto-encoders (CAE) [Chen et al., 2016], can also be used but they are hard to train and not easily interpretable.

Correlated component analysis The correlated component approach of Dmochowski [Dmochowski et al., 2012] is another method that assumes that sources are shared by the subjects. However, its probabilistic version [Kamronn et al., 2015] called BCorrCA yields lower results than MVICA.

2.3 MVICA

All the technical aspects of this section are fully described in Richard et al. [2020].

Contrary to many other Group ICA methods, we impose the matrices S^i to be equal to a common matrix S. We also define a noise parameter that can be interpreted as individual variability. Thus, the model becomes:

$$X^i = A^i(S + N^i) \ , \quad i = 1, \dots, m \ , \tag{4}$$

where X^i, S, $N^i \in \mathbb{R}^{p \times n}$ and $A^i \in \mathbb{R}^{p \times p}$. We assume that samples (i.e. the columns of X^i) are observed i.i.d. For simplicity, we also assume that the sources share the same density d, so that the independence assumption is $D(s) = \prod_{j=1}^{p} d(s_j)$, where D is the density function of the vector s. Finally, we assume that the noise is Gaussian decorrelated of variance σ^2, $N^i \sim \mathcal{N}(0, \sigma^2 I_p)$, and that the noise is independent across subjects and independent from the sources. In practice, estimating the noise level is not really important, so we usually consider that it is equal to 1.

The following proposition extends the standard idenfitiability theory of ICA to MVICA, and shows that recovering the sources/mixing matrices is a well-posed problem, up to scale and permutation.

Theorem 2 (Identifiability of MVICA, prop. 1 of Richard et al. [2020])**.** *Consider $X^i, i = 1, \dots, m$, generated from (4). Assume that $X^i = A'^i(S' + N'^i)$ for some invertible matrices $A'^i \in \mathbb{R}^{p \times p}$, independent non-Gaussian sources $S' \in \mathbb{R}^p$ and Gaussian noise N'^i. Then, there exists a scale and permutation matrix $P \in \mathbb{R}^{p \times p}$ such that for all i, $A'^i = A^i P$.*

Furthermore, we have access to the likelihood of the model. We denote by $W^i = (A^i)^{-1}$ the unmixing matrices, and view the likelihood as a function of W^i rather than A^i. As explained in Richard et al. [2020], the negative log-likelihood writes:

$$\mathcal{L}(W^1, \dots, W^m) = -\sum_{i=1}^{m} \log|W^i| + \frac{1}{2\sigma^2} \sum_{i=1}^{m} \|W^i X^i - \tilde{S}\|^2 + f(\tilde{S}) \ , \tag{5}$$

where f is a smoothened version of the logarithm of the source density d, and $\tilde{S} = \frac{1}{m} \sum_{i=1}^{m} W^i X^i$ are the estimated shared sources.

Minimizing (5) with respect to $\{W^1, \dots, W^m\}$ at once is complicated, so we minimize it iteratively with respect to each W^i. When minimizing (5) with respect to only one W^i, the likelihood simplifies and has the same structure as the usual single-subject cost function. Indeed, it is comparable to (3). Thus, it can be optimized using fast quasi-Newton algorithms, as with the Picard algorithm [Ablin et al., 2018].

The Newton's direction is $(\mathcal{H}^i)^{-1}G^i$ where G^i (resp. $\mathcal{H}^i$) is the gradient (resp. approximation of the Hessian) of $\mathcal{L}$ with respect to W^i.

These quantities $G^i \in \mathbb{R}^{p\times p}$ and $\mathcal{H}^i \in \mathbb{R}^{p\times p\times p\times p}$ are explained in more details in Richard et al. [2020]. They are defined as follows:

$$G^i = \frac{1}{m} f'(\tilde{S})(Y^i)^\top + \frac{1-1/m}{\sigma^2}\left(Y^i - \frac{m}{m-1}\tilde{S}^{-i}\right)(Y^i)^\top - I_p \ \text{, where } Y^i = W^i X^i \tag{6}$$

and, for $a, b, c, d = 1, \dots, p$:

$$\mathcal{H}^i_{abcd} = \delta_{ad}\delta_{bc} + \delta_{ac}\delta_{bd}\Gamma^i_{ab} \text{ with } \Gamma^i_{ab} = \left(\frac{1}{m^2} f''(\tilde{S}_a) + \frac{1-1/m}{\sigma^2}\right)\left(Y^i_b\right)^2 \ . \tag{7}$$

Algorithm 1 alternates one step of this method for each subject until convergence. Also, a line-search is used to ensure that each iteration leads to a decrease of $\mathcal{L}$. The algorithm is stopped when the gradients of $\mathcal{L}$ with respect to W^i are sufficiently low, indicating that the algorithm is close to a stationary point. However, nothing can be said about the distance between this local minimum and the global one.

```
Algorithm 1: MVICA
Input: Dataset $\mathbf{X} = (X^1, \dots, X^m)$, initial unmixing matrices $\mathbf{W} = (W^1, \dots, W^m)$, noise parameter $\sigma$,
       function $f$, tolerance $\varepsilon$
Set tol $= +\infty$, $\tilde{S} = \frac{1}{m}\sum_{i=1}^m W^i X^i$
while $tol > \varepsilon$ do
    tol $= 0$
    for $i = 1 \dots m$ do
        Compute $Y^i = W^i X^i$, $\tilde{S}^{-i} = \tilde{S} - \frac{1}{m}Y^i$, gradient $G^i$ (eq. (6)) and Hessian $\mathcal{H}^i$ (eq. (7))
        Compute the search direction $D = -\left(\mathcal{H}^i\right)^{-1} G^i$
        Find a step size $\rho$ such that $\mathcal{L}^i((I_k + \rho D)W^i) < \mathcal{L}^i(W^i)$ with line search
        Update $\tilde{S} = \tilde{S} + \frac{\rho}{m} D W^i X^i$, $W^i = (I_k + \rho D)W^i$, tol= max(tol, $\|G^i\|$)
    end
end
return Estimated unmixing matrices $\mathbf{W} = (W^1, \dots, W^m)$, estimated shared sources $\tilde{S}$
```

3
MVICA WITH DELAY

This chapter's purpose is to explain how we modified MVICA in order to take subject-specific delays into account. We propose two different solutions.

3.1 Optimize delays and unmixing matrices jointly by likelihood maximization

3.1.1 • Model and likelihood

As detailed in the previous chapter, MVICA's model writes:

$$X^i = A^i(S + N^i) \ , \quad i = 1, \dots, m \ .$$

Our assumption in this internship is that subjects share common but delayed sources. Thus, we have to modify the model in order to take these delays into account. The new model then writes:

$$X^i = A^i(S(\tau_*^i) + N^i) \ , \quad i = 1, \dots, m \ ,$$

where $\tau_*^i \in \{0, \dots, n-1\}$ represents the delay of subject i, and $S(\tau_*^i)$ are the common sources time-shifted by τ_*^i. In this model, the sources of subject i are $S(\tau_*^i)$ and not S anymore.

As a reminder, MVICA is a method that uses maximum likelihood estimation in order to find m unmixing matrices $W^i \in \mathbb{R}^{p \times p}$. Indeed, its (negative log) likelihood is available in closed form:

$$\mathcal{L}(W^1, \dots, W^m) = -\sum_{i=1}^{m} \log|W^i| + \frac{1}{2\sigma^2} \sum_{i=1}^{m} \|W^i X^i - \tilde{S}\|^2 + f(\tilde{S}) \ , \tag{8}$$

where $\tilde{S} = \frac{1}{m} \sum_{i=1}^{m} W^i X^i$ is the source estimate. It is thus possible to optimize this expression with respect to $W^1, \dots, W^m$. To adapt MVICA algorithm, we have to adapt first this likelihood.

In this expression, the role of the squared norm part is to force subjects estimate sources to look alike. Indeed, if estimate sources of subject 1, $W^1 X^1$, are far from estimate sources of subject 2, $W^2 X^2$, then the norm will explode. So, in MVICA model, the estimate sources of the subjects automatically tend towards some shared sources.

In the internship, we assume that subjects' sources only differ by a time-shift (plus some noise). In this case, averaging estimate sources to get $\tilde{S}$ doesn't really make sense, no more than computing the distance between $W^i X^i$ and $\tilde{S}$, since $W^i X^i$ may be a time-shifted version of $\tilde{S}$. Instead, we should first realign the estimate sources by removing their delay, and then compute the distance between the realigned estimated sources and the common sources.

This is precisely what we will do. Let's introduce some parameters $\tau = \{\tau^1, \dots, \tau^m\} \in \{0, \dots, n-1\}^m$, each τ^i representing the estimated delay of subject i.

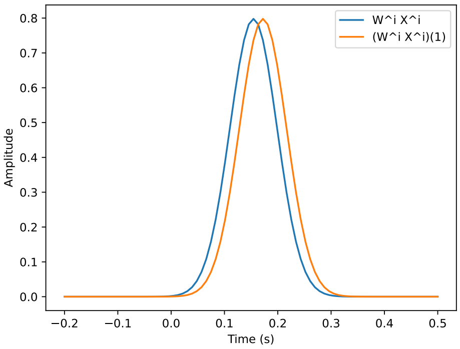


Figure 7: Example of time-shift.

Then, consider that:

$$Y^i := (W^i X^i)(\tau^i) \in \mathbb{R}^{p \times n}$$

is equal to $W^iX^i \in \mathbb{R}^{p\times n}$, but delayed in time by τ^i. Typically, if $\tau^i = 1$, then W^iX^i's last column rolls to the first column to produce Y^i, as shown in figure 7. Using rolling instead of removing the first or last points allows the number of samples to remain constant. In this figure, the blue curve represents a source and the orange curve this same source shifted in time by $\tau^i = 1$. To obtain the orange curve, the blue curve's last point rolled to the first place of the array, producing the orange curve. So, a delay $\tau^i = n$ is equivalent to no delay at all, since all the columns would roll to their initial position. That is why we will often use in the algorithms $\tau = \tau\%n$ where % is the modulo operator.

Therefore, instead of optimizing (8) with respect to the unmixing matrices only, we now allow the estimate sources to vary in time, and thus should optimize (8) with respect to these time parameters too. Thus, the new negative log-likelihood of the model becomes:

$$\mathcal{L}(W^1,\dots,W^m,\tau^1,\dots,\tau^m) = -\sum_{i=1}^m \log|W^i| + \frac{1}{2\sigma^2}\sum_{i=1}^m \|Y^i - \tilde{S}\|^2 + f(\tilde{S}) \ , \tag{9}$$

where $\tilde{S} = \frac{1}{m}\sum_{i=1}^m Y^i$ are the new common sources. The only difference with expression (8) is that we took the delays into account in the terms with the norm and with function f.

The question is now: how do we optimize (9) with respect to $\{W^1,\dots,W^m,\tau^1,\dots,\tau^m\}$? We would like to optimize a complex expression with respect to m parameters and m matrices. Since it is complicated to do it at once, we choose to split the optimization into two parts. First, we minimize (9) with respect to $\{\tau^1,\dots,\tau^m\}$, then we minimize it with respect to $\{W^1,\dots,W^m\}$, then we come back to $\{\tau^1,\dots,\tau^m\}$, etc.

Since the optimization of (9) with respect to $\{W^1,\dots,W^m\}$ is already described in MVICA's paper, we only focus on the minimization with respect to $\{\tau^1,\dots,\tau^m\}$.

3.1.2 • From optimizing delays alone...

Assume that $\{W^1,\dots,W^m\}$ are fixed. We want to minimize (9) with respect to $\{\tau^1,\dots,\tau^m\}$. The first term $-\sum_{i=1}^m \log|W^i|$ is independent from $\{W^1,\dots,W^m\}$, so it can be discarded. In addition, the third term $f(\tilde{S})$ is quite complex, so we chose not to consider it either. Ignoring f was an important choice, but including it in the optimization would have been really complicated. Thus, the new expression $\mathcal{L}'$ that we need to minimize is:

$$\mathcal{L}'(\tau^1,\dots,\tau^m) = \sum_{i=1}^m \|Y^i - \tilde{S}\|^2 \ ,$$

where $Y^i = (W^iX^i)(\tau^i)$ and $\tilde{S} = \frac{1}{m}\sum_{i=1}^m Y^i$.

In order to minimize it, we will use a small trick. Observe that:

$$\mathcal{L}'(\tau^1,\dots,\tau^m) = \sum_{i=1}^m \left\| Y^i - \frac{1}{m}\sum_{i=1}^m Y^j \right\|^2 = \min_{Y\in\mathbb{R}^{p\times n}} \sum_{i=1}^m \left\|Y^i - Y\right\|^2 \ .$$

So, if we take the minimum over $\{\tau^1,\dots,\tau^m\}$ on both sides:

$$\min_{\tau^1,\dots,\tau^m} \mathcal{L}'(\tau^1,\dots,\tau^m) = \min_{\tau^1,\dots,\tau^m,Y} \mathcal{L}''(\tau^1,\dots,\tau^m,Y) \ ,$$

where

$$\mathcal{L}''(\tau^1,\dots,\tau^m,Y) = \sum_{i=1}^m \left\|Y^i - Y\right\|^2 \ . \tag{10}$$

We will try to minimize $\mathcal{L}''$ with respect to $\{\tau^1,\dots,\tau^m,Y\}$ instead of minimizing $\mathcal{L}'$ with respect to $\{\tau^1,\dots,\tau^m\}$. Since $\mathcal{L}''$ is a surrogate [Mairal, 2017] of $\mathcal{L}'$, both functions have the same minimum but they aren't equal.

Minimizing $\mathcal{L}''$ with respect to $\{\tau^1,\dots,\tau^m,Y\}$ at once is complicated, so we chose to optimize it iteratively with respect to each τ^i and then to Y.

This procedure's advantage is that it allows to choose the value of Y at the first iteration. For example, selecting $Y = Y^1$ would be a wise choice. It would force the subjects' sources to align on the sources of the first subject. If we use instead a naive formulation, such as $Y = \frac{1}{m}\sum_{i=1}^m Y^i$, then we would have a flatter reference Y that would not look like a real neural response.

Let's dive into the minimization of (10) with respect to $\{\tau^1, \ldots, \tau^m, Y\}$. First, let us look at the minimization with respect to τ^1. As Y^1 is the only variable that depends on τ^1, we just need to minimize $\left\|Y^1 - Y\right\|^2$ with respect to τ^1. Furthermore:

$$\left\|Y^1 - Y\right\|^2 = \left\|Y^1\right\|^2 + \left\|Y\right\|^2 - 2\langle Y^1, Y\rangle \ .$$

Since $\left\|Y^1\right\|^2$ and $\left\|Y\right\|^2$ don't depend on τ^1, we just have to maximise $\langle Y^1, Y\rangle$ with respect to τ^1. We have:

$$\langle Y^1, Y\rangle = \mathrm{Tr}(Y^1 Y^\top) = \sum_{i=1}^p (Y_i^1)^\top Y_i \ ,$$

where $Y_i^1 \in \mathbb{R}^n$ is the i-th row of Y^1 and $Y_i \in \mathbb{R}^n$ is the i-th row of Y. So, minimizing (10) with respect to τ^1 is easy. It suffices to keep the delay that minimizes the correlation between Y^1 and Y. The computational complexity of this operation is only of $n\log(n)$.

It remains to know how to minimize (10) with respect to Y. The solution is trivially given by:

$$Y = \frac{1}{m}\sum_{i=1}^m Y^i.$$

Now that we know a way of maximizing our likelihood with respect to $\{\tau^1, \ldots, \tau^m\}$, we can write the algorithm that optimizes the delays, when the unmixing matrices are fixed.

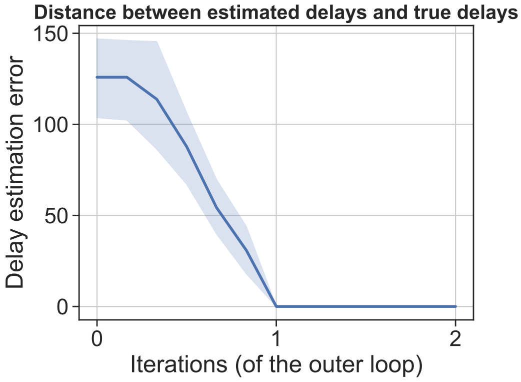


Figure 8: Delays optimization with fixed unmixing matrices.

Algorithm 2: Delay optimization algorithm

```
Input: Estimated sources $\mathbf{S} = \{S^1, \ldots, S^m\}$, number of iterations $n\_iter\_delay$
Set $\mathbf{Y} = \mathbf{S}$, $Y = Y^1$, $\tau = (0, \ldots, 0)$
for $i = 1 \ldots n\_iter\_delay$ do
    Set $\tau' = (0, \ldots, 0)$
    for $j = 1 \ldots m$ do
        Compute the argmax of the sum of correlations between the rows of $Y^j$ and $Y$ and put it in $(\tau')^j$
        Delay $Y^j$ by $(\tau')^j$
    end
    Set $Y = \frac{1}{m}\sum_{i=1}^m Y^i$
    $\tau = (\tau + \tau')\%n$
end
return Estimated delays $\tau$
```

Actually, from the second loop over the subjects, we can optimize with respect to Y after each optimization step with respect to Y^i and, thus, update Y in the loop.

Figure 8 shows the algorithm's performance. Since the sources used as inputs are particularly smooth, the algorithm converges after one loop over $\tau = \{\tau^1, \ldots, \tau^m\}$ in this example. But in practice, it sometimes needs two or three loops to converge. Thus, we set by default $n\_iter\_delay = 3$.

Although this algorithm works well in practice, it does not guarantee that $\mathcal{L}'$ decreases at each iteration. Indeed, it optimizes $\mathcal{L}''$ instead of $\mathcal{L}'$. However, since the only difference between these objective functions lies in the source reference at first iteration, optimizing $\mathcal{L}''$ or $\mathcal{L}'$ does not change a lot the performance of the algorithm.

Yet, we describe in Appendix another algorithm that guarantees the decrease of $\mathcal{L}'$.

3.1.3 • … to optimizing delays and unmixing matrices alternatively

Now that we described the function that minimizes the likelihood with respect to the delays, when unmixing matrices are fixed, we can inject this function into MVICA. Thus, MVICAD proceeds by jointly optimizing subject-specific delays and unmixing matrices as follows: first, optimize delays with A algorithm 2, then optimize unmixing matrices as in MVICA, then optimize delays again, etc.

Since this procedure relies on MVICA, its pseudo-code looks like the one in Richard et al. [2020].

Algorithm 3 summarizes the whole MVICAD procedure.

Algorithm 3: MVICAD

```
Input: Dataset $\mathbf{X} = (X^1, \dots, X^m)$, initial unmixing matrices $\mathbf{W} = (W^1, \dots, W^m)$, noise parameter $\sigma$,
       function $f$, tolerance $\varepsilon$, number of iterations for the delay optimization $n\_iter\_delay$
Set tol $= +\infty$, $\mathbf{S} = (W^1 X^1, \dots, W^m X^m)$
while $tol > \varepsilon$ do
    Only using $\mathbf{S}$ and algorithm 2, compute the delays $\tau = \{\tau^1, \dots, \tau^m\}$ that align as much as possible
      the sources of the subjects
    Set $\mathbf{Y} = \mathbf{S}$ delayed by $\tau$, $\tilde{S} = \frac{1}{m}\sum_{i=1}^m Y^i$, tol $= 0$
    for $i = 1 \dots m$ do
        Set $\bar{X}^i = X^i$ delayed by $\tau^i$, $\tilde{S}^{-i} = \tilde{S} - \frac{1}{m} Y^i$, gradient $G^i$ (eq. (6)) and Hessian $\mathcal{H}^i$ (eq. (7))
        Compute the search direction $D = -\left(\mathcal{H}^i\right)^{-1} G^i$
        Find a step size $\rho$ such that $\mathcal{L}^i((I_k + \rho D) W^i) < \mathcal{L}^i(W^i)$ with line search
        Update $W^i = (I_k + \rho D) W^i$, $Y^i = W^i \bar{X}^i$, $S^i = Y^i$ delayed by $-\tau^i$, $\tilde{S} = \frac{1}{m}\sum_{i=1}^m Y^i$,
          tol$=$ max(tol, $\|G^i\|$)
    end
end
return Estimated unmixing matrices $\mathbf{W} = (W^1, \dots, W^m)$, estimated shared sources $\tilde{S}$, estimated
  delays $\tau$
```

3.2 Fit delays during initialization

Using algorithm 3 alone can give poor results. Indeed, delays also need to be fitted during the initialization of the unmixing matrices, at the very beginning of the algorithm.

Thus, we added a second necessary optimization step that takes place during the unmixing matrices initialization. See Appendix for a better understanding of the problem and why this new optimization step is necessary.

This section aims to explain how this preliminary optimization step is done.

MVICA starts by calling the Picard algorithm [Ablin et al., 2018] on each subjects' data, independently to other subjects. In other words, it starts by calling Picard on first subject's data, X^1, to get W^1. Then, it does the same on X^2 to get W^2, and so on until the last subject. Once the unmixing matrices are initialized, they are used to compute estimated sources as follows:

$$\mathbf{S} = \{W^1 X^1, \dots, W^m X^m\}.$$

But these sources have different delays, since it is the case for $\{X^1, \dots, X^m\}$. In addition, Picard don't always find sources in the same order, with the same signs or even the same scales. For example, first subject's sources may be in another order than second subject's sources.

Consequently, MVICA tries to match the sources of the subjects. In other words, MVICA wants all subjects to have the same sources' order and signs (scale is not important since the estimated sources are normalized). This matching is a particular case of the optimal assignment problem and is solved using the Hungarian algorithm [Tichavsky and Koldovsky, 2004]. Although this matching function works fine when there is no delay, it often fails when some delay is introduced, as shown in Appendix. Intuitively, imagine that there are 2 subjects and 2 sources, a sine and a cosine. If the sources of subject 2 are delayed compared to those of subject 1, then it becomes impossible to differentiate the sine and the cosine. Therefore, we are not able to tell if the sources of the 2 subjects are in the same order and have the same signs.

Thus, MVICA's sources often are in the wrong order after the unmixing matrices initialization. As a consequence, algorithm 3 starts with a bad initialization, and its delays optimization comes too late.

To solve this problem, we propose the following solution. We compare each subject's sources to the sources of subject one. For example, let's consider subjects one and two. If we assume that the rows of S^1 and S^2 are normalized (what is done in the algorithm), and since these rows are independent by hypothesis, we have:

$$S^1(S^1)^\top = S^2(S^2)^\top = I_p \ ,$$

where $I_p \in \mathbb{R}^{p\times p}$ is the identity matrix of size p. Since S^1 and S^2 should be close, up to order, signs and delay, we should also have:

$$S^1(S^2(\tau^2))^\top \approx \Lambda \ ,$$

where $\Lambda \in \mathbb{R}^{p\times p}$ is a scale and permutation matrix, and for a certain $\tau^2 \in \{0, \dots, n-1\}$. We thus keep the delay τ^2 that makes $S^1(S^2(\tau^2))^\top$ look the most like a scale and permutation matrix. This is done using the Amari distance that will be introduced in section 4.1.1. Then, the entries of Λ allow us the change the order and signs of S^2. We repeat this with $S^3, \dots, S^m$. Finally, we have found delays that align all subjects on first subject. This is algorithm 4.

In practice, this method solves the problem mentioned above, as shown in Appendix. Since this method takes place during the initialization, it can be coupled with algorithm 3.

Thus, we have two solutions to deal with delays. Either we only fit delays during initialization, or we fit delays during initialization but also after, using algorithm 3.

Algorithm 4: Delay optimization algorithm with not necessarily aligned sources

```
Input: Estimated sources $\mathbf{s} = \{S^1, \dots, S^m\}$ eventually in different orders and with different signs
Set $\tau = (0, \dots, 0)$
for $i = 2 \dots m$ do
    Set $obj = (0, \dots, 0)$
    for $\tau^i = 0, \dots, n-1$ do
        Set $S'^i = S^i$ delayed by $\tau^i$
        Compute $M = S^1(S'^i)^T$
        Call the Hungarian algorithm [Tichavsky and Koldovsky, 2004] that measures how much $M$ is
          close to a scale-permutation matrix and put this cost in $obj[\tau^i]$
    end
    Set $\tau[i-1] = \mathrm{argmax}(obj)$
end
return Estimated delays $\tau$
```

Computationally, this method computes $(m-1)n$ times a matrix multiplication, and calls the Hungarian algorithm as often. However, it seems hard to do better.

4 NUMERICAL RESULTS

All the code is written in Python. The code of MVICAD and files used to reproduce the figures are available at: https://github.com/AmbroiseHeurtebise/internship_m2.

4.1 Synthetic experiments

4.1.1 • Tools description

Amari distance This chapter's role is to evaluate our new version of MVICA and to compare it to the former version. But first, we need to understand how to measure the quality of an ICA algorithm. As ICA is an unsupervised algorithm, this task is not trivial. Indeed, since we do not have access to the true sources, computing the distance between the true and estimated sources is impossible. Usually, the quality estimation is done using what we call the Amari distance.

Recall that the ICA model is:

$$X = AS \ ,$$

where $X \in \mathbb{R}^{p\times n}$, $A \in \mathbb{R}^{p\times p}$ and $S \in \mathbb{R}^{p\times n}$. In practice, ICA tries to find a matrix $W \in \mathbb{R}^{p\times p}$ such that $W \approx \Lambda A^{-1}$, where $\Lambda \in \mathbb{R}^{p\times p}$ is a scale and permutation matrix. Thus:

$$WA \approx \Lambda \ .$$

Also, having $WA = \Lambda$ would say that we perfectly recovered the sources, according to the identifiability theorem 1. Taking W and A as inputs, the Amari distance precisely computes how much a matrix looks like a scale and permutation matrix.

In practice, we do not have access to the matrix A. Thus, if we want to evaluate our algorithm, we have to simulate data according to the model:

$$X^i = A^i(S(\tau_*^i) + N^i) \ , \quad i = 1, \dots, m \ , \tag{11}$$

where $S(\tau_*^i)$ are the sources of subject i, i.e. $S(\tau_*^i)$ is equal to S but delayed by τ_*^i.

MVICA (or MVICAD) is designed to be applied to multiple subjects, and not only one. When applying these algorithms to $\mathbf{X} = \{X^1, \dots, X^m\}$, we obtain a list of unmixing matrices $\mathbf{W} = \{W^1, \dots, W^m\}$. In order to evaluate our algorithm, we then have to compute the Amari distance for each subject and to average the results, as follows:

$$d_{\text{Amari}}(\mathbf{W}, \mathbf{A}) = \frac{1}{m}\sum_{i=1}^{m} d_{\text{Amari}}(W^i, A^i) \ ,$$

where $\mathbf{A} = \{A^1, \dots, A^m\}$ is the list of the mixing matrices that we generated. We now have access to a measure of the quality of our algorithm on synthetic data. However, this measure is not perfect because it does not depend on the ordering of the sources. Typically, we could swap sources between subjects and still get a 0 error.

Reconstruction error Before comparing the different algorithms, we propose here another way of evaluating our results. We recall that our final objective is to recover sources from the input signals. Hence, the quantities of interest are the sources, rather than the mixing matrices. Yet, the Amari distance focuses on the mixing matrices instead of the sources. This motivates the use of another measure, that we call reconstruction error, that uses the sources as input.

The principle is very similar to the Amari distance. Let's assume for now that there is no delay in the model and only one subject. We recall that with ICA, we search for a matrix $W \in \mathbb{R}^{p\times p}$ such that $W \approx \Lambda A^{-1}$, where $\Lambda \in \mathbb{R}^{p\times p}$ is a scale and permutation matrix. Thus:

$$\hat{S} := WX \approx \Lambda A^{-1}X = \Lambda S \ .$$

So, the estimated sources $\hat{S}$ that we obtain at the end of the ICA algorithm should look like the true sources S, up to scale and permutation. This means that:

$$\hat{S}S^\top \approx \Lambda SS^\top \ .$$

So, if we assume that the rows of S are normalized (which can be done in the algorithm), and since these rows are statistically independent by hypothesis, then $SS^\top = I_p$, where $I_p \in \mathbb{R}^{p\times p}$ is the identity matrix of size p. Thus:

$$\hat{S}S^\top \approx \Lambda \ .$$

In other words, we want the square matrix $\hat{S}S^\top$ to be as close as possible to a scale and permutation matrix (actually, since we normalized the rows of S, we want this square matrix to be as close as possible to a sign and permutation matrix). This measurement is done by computing the Amari distance between $\hat{S}S^\top$ and the identity matrix I_p. Hence, the reconstruction error's formula is:

$$\mathrm{RE}(\hat{S}, S) = d_{\mathrm{Amari}}(\hat{S}S^\top, I_p) \ .$$

As with the Amari distance, the reconstruction error should be minimized.

Now, let us take into account the fact that, in our experiments, we have multiple subjects with delayed sources. Since the true sources are not delayed in the data generation (as we will see right after), there is a time shift between the estimated sources $\hat{S}$ that we get from MVICA (or MVICAD) and the true sources S. To deal with this time shift, we compute the reconstruction error for every possible delay and only keep the lowest result. So, in the case of multiple subjects, the reconstruction error's formula is:

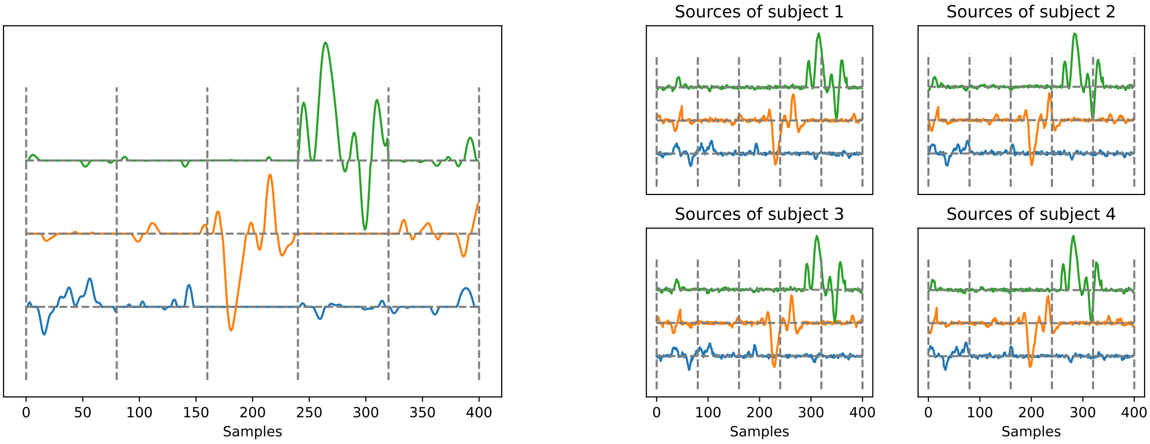
$$\mathrm{RE}(\hat{S}, S) = \min_{\tau} \left\{ d_{\mathrm{Amari}}(\hat{S}(\tau)S^\top, I_p) \right\} \ .$$

This method gives us a second way of evaluating our algorithms. Contrary to the Amari distance, it depends on the ordering of the sources.

Data generation It remains to generate data that roughly look like neuroscience time series. As a reminder, we have to generate time series as in model (11). That means that we have to generate X^i, A^i and $S(\tau_*^i)$ for $i = 1, \dots, m$. We begin by sampling the common sources S. These sources have to be super-Gaussian (see section 2.1 for an explanation), which is a complicated thing to do, and temporally smooth enough (otherwise, it would look like noise). The idea behind super-Gaussian variables is that they often are close to 0 and sometimes explode. In order to generate data with this particular shape, we chose to separate the signals into intervals that each have a random intensity before applying a soft-treshold. This treshold forces the signals of intervals with low intensity to be equal to 0. On the other intervals (those with higher intensity), we sampled random frequencies to obtain wave curves. The explanation of the sources generation is not really important, what to remember is that the sources S look like figure 9a which shows the sources of 4 subjects.

Then, we sample a random delay τ_*^i per subject, uniformly in $\{0, \dots, T\}$ where $T < n$ is the largest possible delay, and shift the sources by these delays. This gives us a new source matrix $S(\tau_*^i)$ for each subject. These sources look like figure 9b.

Finally, we sample the mixing matrices $\mathbf{A} = \{A^1, \dots, A^m\}$ from a normal distribution. The computation $X^i = A^i(S(\tau_*^i) + N^i)$ then gives us the MEG-like signals $\mathbf{X} = \{X^1, \dots, X^m\}$. This data generation coupled with the Amari distance explained above allows us to finally evaluate our algorithms.



(a) True sources.

(b) Delayed sources of 4 subjects with noise.

Figure 9: Example of sources generation.

4.1.2 • Results

Now that we detailed how the Amari distance and the reconstruction error work, we are able to compare our algorithms.

As explained in the previous chapter, we proposed two solutions in order to deal with the subject specific-delays. One solution is only implemented during the algorithm's initialization, and the other one during the algorithm's initialization and also during the core optimization, alongside the unmixing matrices optimization. We start this section by comparing these two solutions. Then, we will compare one of the two solutions to the MVICA algorithm. This will be the main result of the internship.

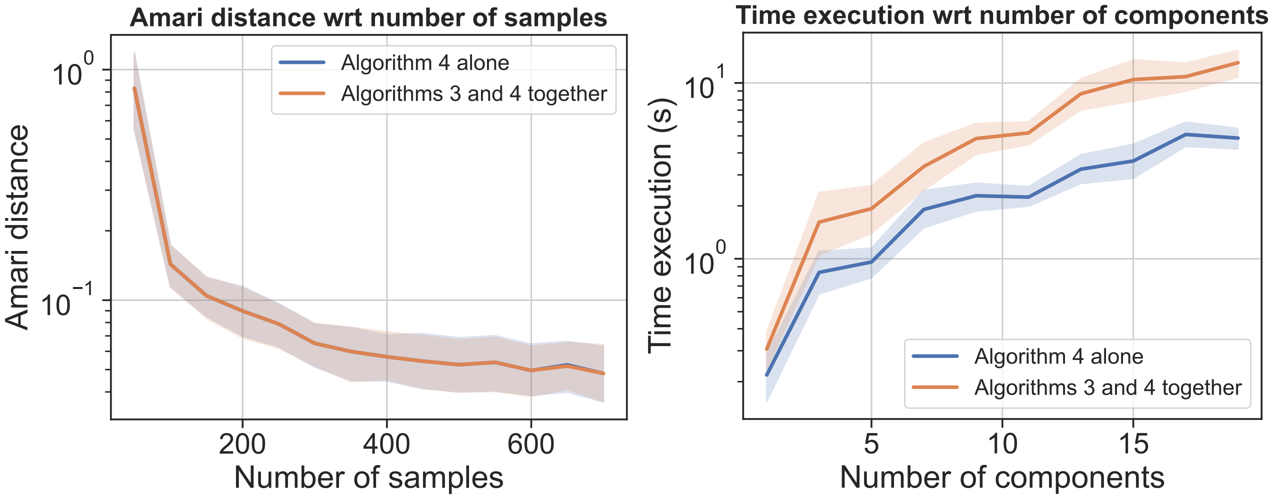
Comparison of the two solutions Let us compare the two solutions described in chapter 3. We chose to use the Amari distance for now, instead of the reconstruction error, because it is more usual. The idea is that also optimizing delays alongside unmixing matrices is computationaly more expensive than only optimizing them at the beginning. So, we want to know if this second delay optimization step is really useful.

In the figures below, we used the default parameters of MVICAD. The blue curve represents MVICAD with delays optimization only during initialization, and the orange curve represents MVICAD with both delays optimization methods.

Figure 10a shows that using both optimization methods does not increase performance, compared to only optimizing delays at the beginning. Indeed, the two solutions' Amari distance seem to be equal, regardless of the number of samples. If we look in detail, we can still see some very little differences. See Appendix for more plots about the compared performances of the two solutions. Of course, these results are greatly conditioned by our data generation. We tried several parameters and always get comparable performances for the two solutions. But it is possible that, using a particular data generation, adding a second delay optimization step may prove necessary. Unfortunately, we were unable to create such a counter-example. So, for the moment, let us consider that the two solutions have approximately the same performance.

On the other hand, adding a second optimization step clearly takes more time, as shown in figure 10b. Thus, from now on, MVICAD will refer to MVICA with algorithm 4 included at the beginning, but without algorithm 3. In other words, from now on, MVICAD will only optimize the delays during the initialization of the unmixing matrices.

Due the previous plots, we could think that adding a second-step optimization, alongside unmixing matrices optimization, was useless. But we will come back to this solution later. Indeed, it could become really useful when considering source-specific delays, in addition to subject-specific delays. See section 5.3 for more information.



(a) Amari distance of the two solutions. We clearly observe that their performances are equivalent.

(b) Time execution of the two solutions. We clearly observe that adding a second optimization step takes more time.

Figure 10: Comparison of the two solutions.

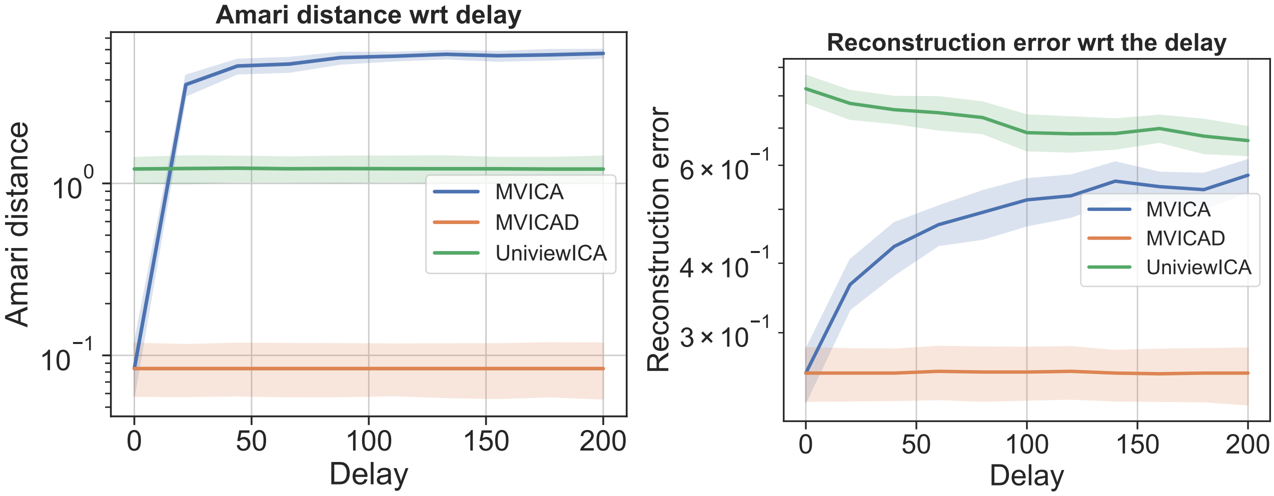
Comparison of MVICA and MVICAD In this section, we finally compare our new MVICAD algorithm to MVICA. It constitutes the main results of the internship, as the original purpose was to modify MVICA in order to overcome the time-delay issue.

Figures 11a and 11b clearly prove that MVICAD outperforms MVICA, when some delay is introduced in the model. Let us explain them. To produce these figures, we used the data generation detailed at the beginning of the chapter to obtain signals $\mathbf{X} = \{X^1, \dots, X^m\}$, mixing matrices $\mathbf{A} = \{A^1, \dots, A^m\}$ and sources $\mathbf{S} = \{S^1, \dots, S^m\}$. Then, we applied MVICA and MVICAD to the signals $\mathbf{X}$, which gave us estimated unmixing matrices $\mathbf{W} = \{W^1, \dots, W^m\}$. Finally, it remained to compute the Amari distance, using $\mathbf{W}$ and $\mathbf{A}$ as inputs, and the reconstruction error, using $\mathbf{W}$, $\mathbf{X}$ and $\mathbf{S}$ as inputs. See the tools description 4.1.1 for a better understanding of the Amari distance and the reconstruction error. We used this procedure with various quantities of delays introduced in the model and multiple random states. Also, we took 6 subjects, 10 sources and 400 samples.

In figure 11a, UniviewICA consists in applying the Picard algorithm on each subject independently. Each call to Picard gives as output an unmixing matrix W^i, which is then used to compute the Amari distance. By construction, this method is thus delay-independent, hence the constant curve. Moreover, UniviewICA doesn't take advantage of the group-structure, unlike MVICA, so it only serves as a baseline.

The conclusion that MVICAD outperforms MVICA comes from the blue and orange curves shape. When there is no delay, these curves start from the same point. Indeed, in this case, MVICAD correctly finds that the delays equal 0, so the two procedures are equivalent. But quickly, as some delay is introduced, the blue curve increases, while the orange one remains constant. In other words, as some delay is introduced, MVICA's performance decreases, while the one of MVICAD remains constant.

These results are easily explained. Assume that each subject has a specific delay. MVICA starts by initializing the unmixing matrices, using the Picard algorithm on each subject's data independently. These unmixing matrices allow us to compute estimated sources for each subject, but these sources are delayed and not necessarily in the same order and with the same signs. So, MVICA tries to match the sources of the subjects. In other words, MVICA wants all subjects to have the same sources' order and signs. Although this matching works fine when there is no delay, it often fails when some delay is introduced, as shown in Appendix. Thus, MVICA's sources often are in the wrong order after the unmixing matrices initialization. In addition, as MVICA doesn't take into account the delays, the sources of different subjects are time-shifted. For those two reasons, when averaging the subjects' sources, it results in a bad estimation of the source matrix. This explains why MVICA's



(a) Amari distance of MVICA and MVICAD.

(b) Rec. error of MVICA and MVICAD.

Figure 11: Comparison of MVICA and MVICAD.

Amari distance and reconstruction error are high when some delay is introduced.

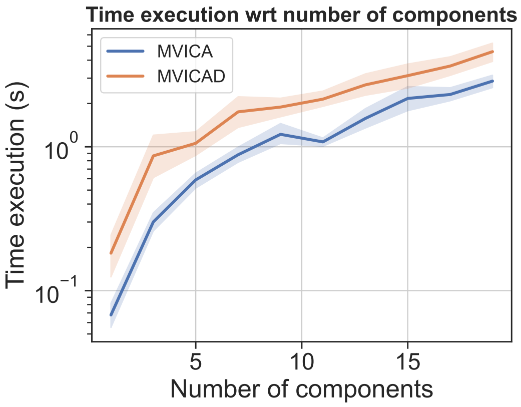


Figure 12: Time execution of MVICA and MVICAD.

On the other hand, MVICAD removes the subject-specific time-shifts, after initializing the unmixing matrices. Consequently, the sources matching function works fine (as if there wasn't no delays), and the resulting estimated sources are in the good order. Then, since the delays are estimated, MVICAD's sources of different subjects aren't time-shifted, unlike with MVICA. Thus, averaging these sources makes sense. So, regardless the quantity of delay introduced, MVICAD removes these delays and acts as if there was no delays at all. Hence the constant Amari distance and reconstruction error. What to remember is that this method removes the delays from the beginning. If the delays estimation at the beginning is good, then this method can be delay-independent.

UniviewICA's Amari distance is constant too, with respect to the quantity of delay introduced, because delay has no impact on the estimated unmixing matrix when there is only one subject. However, the Amari distance is much higher than the one of MVICAD because of the wrong orders and signs of the sources, as explained above. On the other hand, UniviewICA's reconstruction error is not constant. This is normal since we average subjects' sources, which are delayed, to compute it. Thus, it depends on the delay. The fact that the green curve decreases a little bit is not very important. What to remember about figure 11b is especially that it corroborates the results of figure 11a.

Although MVICAD shows better results than MVICA, it is computationaly more expensive. Indeed, figure 12 presents the time execution of both algorithms, with different numbers of components. We observe that this duration is far lower for MVICA than for MVICAD. This makes perfect sense, since MVICAD has more optimization steps than MVICA. Thus, MVICAD should only be used in situations where some delays really appear, as in neurological data. In situations where the different views don't present time-shifts, MVICAD would unnecessarily take more time than MVICA and would have comparable performances.

4.2 Real data experiment

Since ICA is an unsupervised algorithm, it is complicated to draw conclusions from real data experiments. In this section, we will discuss some real data results, but without ranking the algorithms.

The experiment in figure 13 was done using 200 Cam-CAN subjects. We didn't use the 618 available subjects to save some time.

The figure shows time courses of 9 sources (one color per source) for MVICA and MVICAD.

The differences between MVICA's sources and MVICAD's sources are small, so hard to see. But we can still observe some differences.

For example, some MVICAD's sources appear to be a little bit sharper than those of MVICA. It could be a coincidence, so we cannot draw any conclusion from this example alone, but this observation would actually make sense. Indeed, since MVICAD removes possible delays, it never averages delayed sources but only realigned sources. Averaging sources presenting different time-shifts give a flatter result than averaging aligned sources. That is why this observation would make sense.

If we want to observe clearer differences, we will probably have to take source-specific delays into account, as proposed in section 5.3.

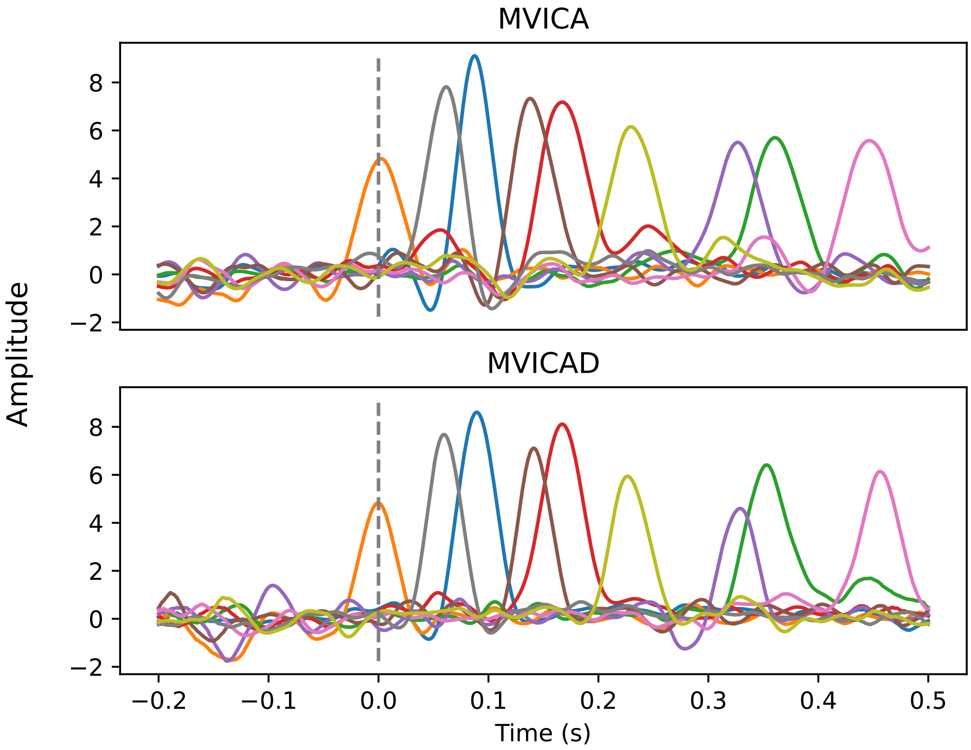


Figure 13: Cam-CAN experiment.

Figure 14 represents 5 of the 9 sources obtained in figure 13, for which we particularly observe a difference in sharpness between MVICA and MVICAD. Once again, this does not prove anything but tends to confirm our point.

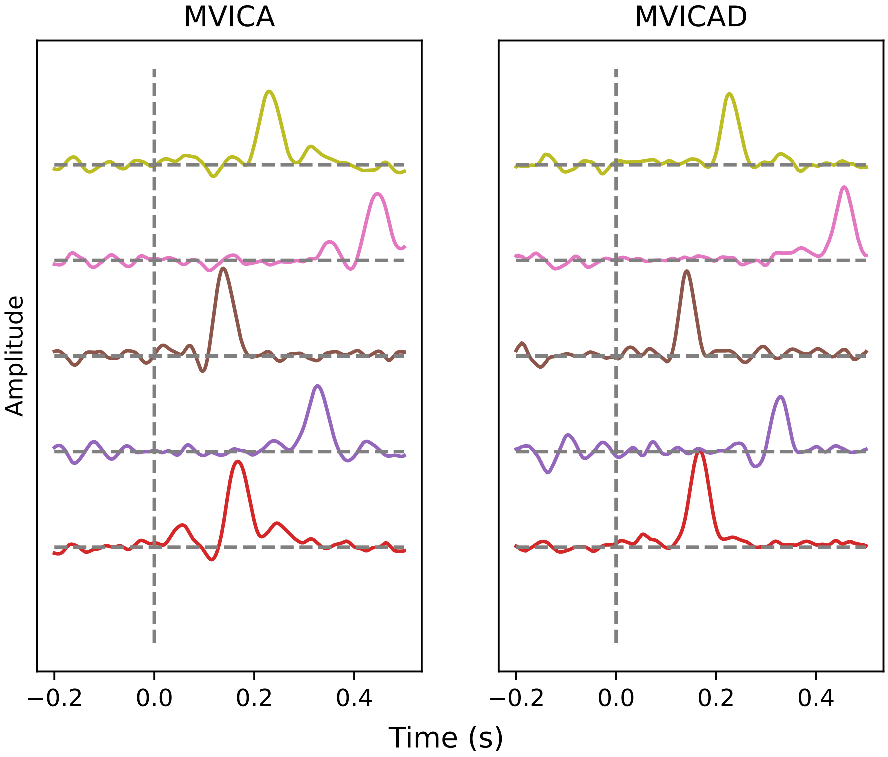


Figure 14: 5 of the 9 sources. We observe that MVICAD sources look sharper than MVICA ones.

5
CONCLUSION AND FUTURE WORKS

5.1 Conclusion

In this work, we introduced a modified version of the already existing MVICA and showed improved performance on synthetic data. We also tested our algorithm on real data and obtained similar results to MVICA. Importantly, our model estimates subject-specific delay and could therefore be used to explore relations between age and delay.

5.2 Reduce MVICAD computation time

Figure 12 showed that MVICAD takes more time than MVICA. Although it makes perfect sense, reducing MVICAD time execution would be great. An idea would be to only consider small delays instead of all possible delays between 0 and n, since having important delays is not very realistic.

Indeed, algorithms 2 and 4 make computations for each possible delay between 0 and n. Instead, we could only look at delays between 0 and $n/6$ and delays between $5n/6$ and n for example. For algorithm 4, this would divide time execution by 3.

5.3 Source-specific delays

For the moment, we only took into account subject-specific delays, but never source-specific delays. However, this would make sense in a neurobiological way, since temporal variability in the neural responses of subjects could also come from different regions of the brain.

Moreover, algorithm 2 could easily be adapted in order to take these new delays into account. In total, we would consider pm different delays instead of m.

5.4 Auto-supervision

Another very important line of research is self-supervision. Self-supervised learning (SSL) is used in situations where obtaining labeled data is difficult. SSL main idea is to learn a representation on a supervised "pretext task" that has to be invented, in order to reuse later this representation in unsupervised contexts. This kind of method greatly reduces the number of labeled examples required to train a model.

Recently, SSL has been applied to clinical EEG data [Banville et al., 2021], revealing insights on the data without any human supervision. However, such approaches are not generalized to multi-view learning yet. Finding pretext tasks that use the views indices as supervision could be an interesting idea in order to generalize SSL to multi-view learning.

REFERENCES

Hugo Richard, Luigi Gresele, Aapo Hyvarinen, Bertrand Thirion, Alexandre Gramfort, and Pierre Ablin. Modeling shared responses in neuroimaging studies through multiview ica. *Advances in Neural Information Processing Systems*, 33:19149–19162, 2020.

Arnaud Delorme, Terrence Sejnowski, and Scott Makeig. Enhanced detection of artifacts in eeg data using higher-order statistics and independent component analysis. *Neuroimage*, 34(4):1443–1449, 2007.

Anthony J Bell and Terrence J Sejnowski. The "independent components" of natural scenes are edge filters. *Vision research*, 37(23):3327–3338, 1997.

Pamela K Douglas, Edward Lau, Ariana Anderson, Austin Head, Wesley Kerr, Margalit Wollner, Daniel Moyer, Wei Li, Mike Durnhofer, Jennifer Bramen, et al. Single trial decoding of belief decision making from eeg and fmri data using independent components features. *Frontiers in human neuroscience*, 7:392, 2013.

MS Lewicki. A review of methods for spike sorting: the detection and classification of neural action potentials. network 9. *R53–R78*, 1998.

Vince D Calhoun, Tulay Adali, Godfrey D Pearlson, and James J Pekar. A method for making group inferences from functional mri data using independent component analysis. *Human brain mapping*, 14(3):140–151, 2001a.

Michal Teplan et al. Fundamentals of eeg measurement. *Measurement science review*, 2(2):1–11, 2002.

David Cohen. Magnetoencephalography: evidence of magnetic fields produced by alpha-rhythm currents. *Science*, 161(3843):784–786, 1968.

Eric Racine, Ofek Bar-Ilan, and Judy Illes. fmri in the public eye. *Nature Reviews Neuroscience*, 6(2):159–164, 2005.

Jason R Taylor, Nitin Williams, Rhodri Cusack, Tibor Auer, Meredith A Shafto, Marie Dixon, Lorraine K Tyler, Richard N Henson, et al. The cambridge centre for ageing and neuroscience (cam-can) data repository: Structural and functional mri, meg, and cognitive data from a cross-sectional adult lifespan sample. *neuroimage*, 144:262–269, 2017.

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

Shiliang Sun. A survey of multi-view machine learning. *Neural computing and applications*, 23(7):2031–2038, 2013.

Hugo Richard, Pierre Ablin, Bertrand Thirion, Alexandre Gramfort, and Aapo Hyvarinen. Shared independent component analysis for multi-subject neuroimaging. *Advances in Neural Information Processing Systems*, 34: 29962–29971, 2021.

Darren Price, Lorraine K Tyler, R Neto Henriques, Karen L Campbell, Nitin Williams, Matthias S Treder, Jason R Taylor, and Richard NA Henson. Age-related delay in visual and auditory evoked responses is mediated by white-and grey-matter differences. *Nature communications*, 8(1):1–12, 2017.

Timothy PL Roberts, Sarah Y Khan, Mike Rey, Justin F Monroe, Katelyn Cannon, Lisa Blaskey, Sarah Woldoff, Saba Qasmieh, Mike Gandal, Gwen L Schmidt, et al. Meg detection of delayed auditory evoked responses in autism spectrum disorders: towards an imaging biomarker for autism. *Autism Research*, 3(1):8–18, 2010.

Pierre Ablin. *Exploration of multivariate EEG/MEG signals using non-stationary models*. PhD thesis, Université Paris-Saclay (ComUE), 2019.

Jonathon Shlens. A tutorial on independent component analysis. *arXiv preprint arXiv:1404.2986*, 2014.

Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

George Darmois. Analyse générale des liaisons stochastiques: etude particulière de l'analyse factorielle linéaire. *Revue de l'Institut international de statistique*, pages 2–8, 1953.

Aapo Hyvärinen. New approximations of differential entropy for independent component analysis and projection pursuit. *Advances in neural information processing systems*, 10, 1997a.

Aapo Hyvärinen. Independent component analysis by minimization of mutual information. 1997b.

D Pham, P Garrat, and C Jutten. Separation of a mixture of independent sources through a maximum likelihood approach, in 'european signal processing conference'. 1992.

Tzyy-Ping Jung, Colin Humphries, Te-Won Lee, Scott Makeig, Martin McKeown, Vicente Iragui, and Terrence J Sejnowski. Extended ica removes artifacts from electroencephalographic recordings. *Advances in neural information processing systems*, 10, 1997.

Pierre Ablin, Jean-François Cardoso, and Alexandre Gramfort. Faster independent component analysis by preconditioning with hessian approximations. *IEEE Transactions on Signal Processing*, 66(15):4040–4049, 2018.

Aapo Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE transactions on Neural Networks*, 10(3):626–634, 1999.

Gaël Varoquaux, Sepideh Sadaghiani, Jean Baptiste Poline, and Bertrand Thirion. Canica: Model-based extraction of reproducible group-level ica patterns from fmri time series. *arXiv preprint arXiv:0911.4650*, 2009.

Hejia Zhang, Po-Hsuan Chen, Janice Chen, Xia Zhu, Javier S Turek, Theodore L Willke, Uri Hasson, and Peter J Ramadge. A searchlight factor model approach for locating shared information in multi-subject fmri analysis. *arXiv preprint arXiv:1609.09432*, 2016.

Ying Guo and Giuseppe Pagnoni. A unified framework for group independent component analysis for multi-subject fmri data. *NeuroImage*, 42(3):1078–1093, 2008.

Olivier Bermond and Jean-François Cardoso. Approximate likelihood for noisy mixtures. In *Proc. ICA*, volume 99, pages 325–330. Citeseer, 1999.

Kaare Brandt Petersen, Ole Winther, and Lars Kai Hansen. On the slow convergence of em and vbem in low-noise linear models. *Neural computation*, 17(9):1921–1926, 2005.

Po-Hsuan Cameron Chen, Janice Chen, Yaara Yeshurun, Uri Hasson, James Haxby, and Peter J Ramadge. A reduced-dimension fmri shared response model. *Advances in Neural Information Processing Systems*, 28, 2015.

Christian F Beckmann and Stephen M Smith. Tensorial extensions of independent component analysis for multisubject fmri analysis. *Neuroimage*, 25(1):294–311, 2005.

Fengyu Cong, Zhaoshui He, Jarmo Hämäläinen, Paavo HT Leppänen, Heikki Lyytinen, Andrzej Cichocki, and Tapani Ristaniemi. Validating rationale of group-level component analysis based on estimating number of sources in eeg through model order selection. *Journal of neuroscience methods*, 212(1):165–172, 2013.

Vera A Grin-Yatsenko, Ineke Baas, Valery A Ponomarev, and Juri D Kropotov. Independent component approach to the analysis of eeg recordings at early stages of depressive disorders. *Clinical Neurophysiology*, 121(3):281–289, 2010.

Vince D Calhoun, Tülay Adali, Vince B McGinty, James J Pekar, Todd D Watson, and Godfrey D Pearlson. fmri activation in a visual-perception task: network of areas detected using the general linear model and independent components analysis. *NeuroImage*, 14(5):1080–1088, 2001b.

Ricardo Pio Monti and Aapo Hyvärinen. A unified probabilistic model for learning latent factors and their connectivities from high-dimensional data. *arXiv preprint arXiv:1805.09567*, 2018.

Petr Tichavsky and Zbynek Koldovsky. Optimal pairing of signal components separated by blind techniques. *IEEE Signal Processing Letters*, 11(2):119–122, 2004.

Po-Hsuan Chen, Xia Zhu, Hejia Zhang, Javier S Turek, Janice Chen, Theodore L Willke, Uri Hasson, and Peter J Ramadge. A convolutional autoencoder for multi-subject fmri data aggregation. *arXiv preprint arXiv:1608.04846*, 2016.

Jacek P Dmochowski, Paul Sajda, Joao Dias, and Lucas C Parra. Correlated components of ongoing eeg point to emotionally laden attention–a possible marker of engagement? *Frontiers in human neuroscience*, 6:112, 2012.

Simon Kamronn, Andreas Trier Poulsen, and Lars Kai Hansen. Multiview bayesian correlated component analysis. *Neural computation*, 27(10):2207–2230, 2015.

Julien Mairal. *Large-Scale Machine Learning and Applications*. PhD thesis, UGA-Université Grenoble Alpes, 2017.

Hubert Banville, Omar Chehab, Aapo Hyvärinen, Denis-Alexander Engemann, and Alexandre Gramfort. Uncovering the structure of clinical eeg signals with self-supervised learning. *Journal of Neural Engineering*, 18 (4):046020, 2021.

APPENDIX

Modify algorithm 2 to guarantee likelihood decrease

In section 3.1.2, we saw a way to minimize:

$$\mathcal{L}'(\tau^1, \dots, \tau^m) = \sum_{i=1}^m \|Y^i - \tilde{S}\|^2 \ ,$$

with respect to $\tau = \{\tau^1, \dots, \tau^m\}$. This way allowed to choose the source reference Y at the first iteration but did not guarantee that $\mathcal{L}'$ decreased at each iteration. We introduce another method for minimizing $\mathcal{L}'$, that has this propriety.

We can observe that:

$$\begin{aligned}\mathcal{L}'(\tau^1, \dots, \tau^m) &= \sum_{i=1}^m \left\| Y^i - \frac{1}{m}\sum_{i=1}^m Y^j \right\|^2 \\ &= \sum_{i=1}^m \left\| \frac{m-1}{m} Y^i - \frac{1}{m}\sum_{j\neq i} Y^j \right\|^2 \\ &= \left(\frac{m-1}{m}\right)^2 \sum_{i=1}^m \left\| Y^i - \frac{1}{m-1}\sum_{j\neq i} Y^j \right\|^2 \ .\end{aligned}$$

Hence, we only need to optimize:

$$\sum_{i=1}^m \left\| Y^i - \frac{1}{m-1}\sum_{j\neq i} Y^j \right\|^2 \ .$$

Let $Y^{-i} := \frac{1}{m-1}\sum_{j\neq i} Y^j$ be the mean of the estimated sources of all subjects except subject i. Then, the goal is to minimize:

$$\sum_{i=1}^m \left\|Y^i - Y^{-i}\right\|^2 = \sum_{i=1}^m \|Y^i\|^2 + \|Y^{-i}\|^2 - 2\left\langle Y^i, Y^{-i}\right\rangle \tag{12}$$

with respect to τ. But $\|Y^i\|^2$ and $\|Y^{-i}\|^2$ are independent from τ. So, minimizing (12) corresponds to maximizing:

$$\sum_{i=1}^m \langle Y^i, Y^{-i}\rangle \ . \tag{13}$$

Maximizing (13) with respect to all the entries of $\tau = \{\tau^1, \dots, \tau^m\}$ at once can be complicated, so we optimize it iteratively with respect to each τ^i, $i = 1, \dots, m$. Let us look at the maximization of (13) with respect to τ^1 for example. We have:

$$\begin{aligned}\sum_{i=1}^m \langle Y^i, Y^{-i}\rangle &= \langle Y^1, Y^{-1}\rangle + \sum_{i=2}^m \langle Y^i, \frac{1}{m-1}\sum_{j\neq i} Y^j\rangle \\ &= \langle Y^1, Y^{-1}\rangle + \frac{1}{m-1}\sum_{i=2}^m \sum_{j\neq i} \langle Y^i, Y^j\rangle \ . \end{aligned} \tag{14}$$

The second term in (14) depends on τ^1 only when $j = 1$. So, the optimization boils down to maximizing:

$$\langle Y^1, Y^{-1}\rangle + \frac{1}{m-1}\sum_{i=2}^{m}\langle Y^i, Y^1\rangle = \langle Y^1, Y^{-1}\rangle + \langle\frac{1}{m-1}\sum_{i=2}^{m} Y^i, Y^1\rangle = 2\,\langle Y^1, Y^{-1}\rangle$$

with respect to τ^1.

In other words, for all $i \in \{1, \dots, m\}$, minimizing (12) with respect to τ^i boils down to maximizing $\langle Y^i, Y^{-i}\rangle$ with respect to this τ^i. We have:

$$\langle Y^i, Y^{-i}\rangle = \sum_{j=1}^{p}(Y_j^i)^\top Y_j^{-i} \quad , \tag{15}$$

where $Y_j^i \in \mathbb{R}^n$ is the j-th row of the estimated sources Y^i of subject i and $Y_j^{-i} \in \mathbb{R}^n$ is the j-th row of Y^{-i} and doesn't depend on τ^i.

To maximise (15) with respect to τ^i, it suffices to keep the delay τ^i that minimizes the correlation between Y^i and Y^{-i}. It is easily done in Python.

Remark that we could also maximise (13) with respect to τ directly, instead of maximizing it with respect to each τ^i iteratively. But it would require to evaluate n^m times the quantity (13). With our procedure, we only evaluate it nm times.

This new method gives us algorithm 5.

Algorithm 5: Delay optimization algorithm with decrease guarantee

```
Input: Estimated sources $\mathbf{S} = \{S^1, \dots, S^m\}$, number of iterations $n\_iter\_delay$
Set $\mathbf{Y} = \mathbf{S}$, $Y = \frac{1}{m}\sum_{i=1}^{m} Y^i$, $\tau = (0, \dots, 0)$
for $i = 1 \dots n\_iter\_delay$ do
    Set $\tau' = (0, \dots, 0)$
    for $j = 1 \dots m$ do
        Set $Y' = \frac{m}{m-1}(Y - Y^j/m)$
        Compute the argmax of the sum of correlations between the rows of $Y^j$ and $Y'$ and put it in $\tau'^j$
        Set $Y_{\text{old}} = Y^j$
        Delay $Y^j$ by $\tau'^j$
        Set $Y = Y + (Y^j - Y_{\text{old}})/m$
    end
    $\tau = (\tau + \tau')\%n$
end
return Estimated delays $\tau$
```

Unfortunately, this algorithm forces us to take $Y = \frac{1}{m}\sum_{i=1}^{m} Y^i$ as the source reference, at first iteration, which does not really make sense if the sources are delayed. We would prefer to take $Y = Y^1$ as the source reference at the beginning, like with algorithm 2. This motivates the creation of algorithm 6, which combines the advantages of algorithms 2 and 5.

Algorithm 6: Delay optimization algorithm with decrease guarantee and initialization of the source reference at the sources of the first subject

```
Input: Estimated sources $\mathbf{S} = \{S^1, \dots, S^m\}$, number of iterations $n\_iter\_delay >= 2$
Set $\mathbf{Y} = \mathbf{S}$, $Y = Y^1$, $\tau = (0, \dots, 0)$
for $i = 1 \dots m$ do
    Compute the argmax of the sum of correlations between the rows of $Y^i$ and $Y$ and put it in $\tau^i$
    Delay $Y^i$ by $\tau^i$
end
Set $Y = \frac{1}{m} \sum_{i=1}^m Y^i$
$\tau = \tau \% n$
for $i = 2 \dots n\_iter\_delay$ do
    Set $\tau' = (0, \dots, 0)$
    for $j = 1 \dots m$ do
        Set $Y' = \frac{m}{m-1}(Y - Y^j/m)$
        Compute the argmax of the sum of correlations between the rows of $Y^j$ and $Y'$ and put it in $\tau'^j$
        Set $Y_{\text{old}} = Y^j$
        Delay $Y^j$ by $\tau'^j$
        Set $Y = Y + (Y^j - Y_{\text{old}})/m$
    end
    $\tau = (\tau + \tau')\% n$
end
return Estimated delays $\tau$
```

Why is algorithm 3 alone not effective?

In section 3.2, we said that optimizing delays with algorithm 3, but without optimizing them during unmixing matrices initialization, sometimes gives bad results. We also explained that it comes from the fact that, when some delay is introduced, the function that matches subjects' sources can fail. We show here an example of such a failure.

Figure 15a presents estimated sources of four subjects, after unmixing matrices initialization. We clearly observe three facts.

First, sources are not necessarily in the same order: for example, the first source of subject one corresponds to the second source of subject three. Second, sources have not necessarily the same sign: for example, the first source of subject one corresponds to the first source of subject four, but with inverted signs. Third, sources are delayed: we see that peaks of different subjects do not occur at the same time. In the figure, red rectangles only serve to make observation easier.

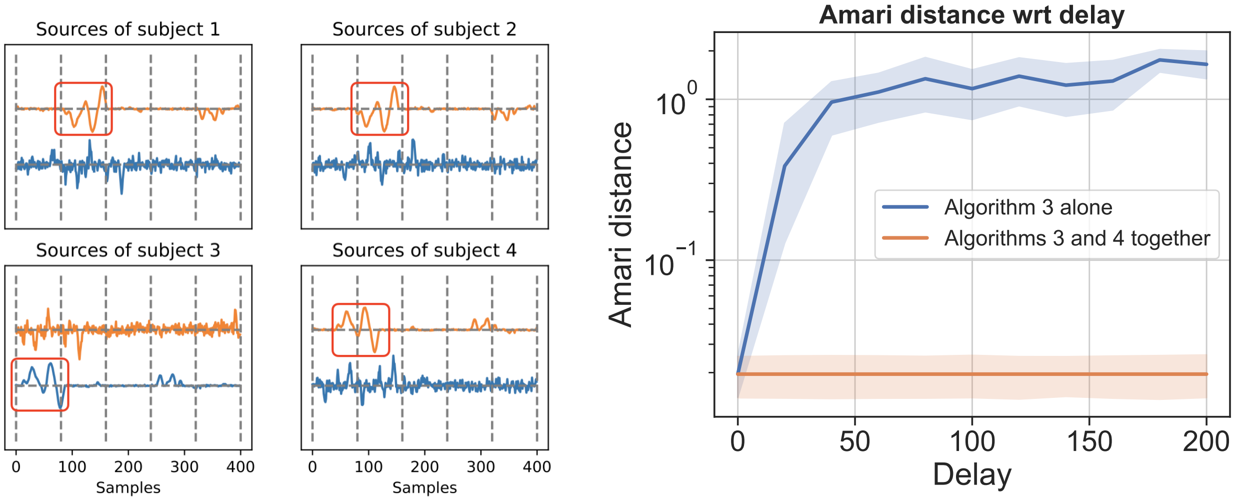
Figure 15b shows the Amari distance when we apply algorithm 3 alone, compared to when we apply algorithms 3 and 4 together. We clearly see that using algorithm 3 alone strongly increases the Amari distance when there is some delay.

MVICAD hyperparameters

We added two hyperparameters in MVICAD, related to the function that finds delays and unmixing matrices alternatively.

The first parameter serves as early stopping for the optimization of the delays. By default it is None, but if an iteration number is specified, then from this iteration MVICAD only optimizes unmixing matrices and not delays anymore.

The second parameter looks like the first one. By default it is None, but if a positive number N is specified, then delays optimization is performed every N iterations of unmixing matrices optimization.



(a) Sources of 4 subjects after unmixing matrices initialization. The sources present different orders, signs and delays.

(b) Amari distance of algorithm 3 alone and algorithms 3 and 4 together, with respect to the delay.

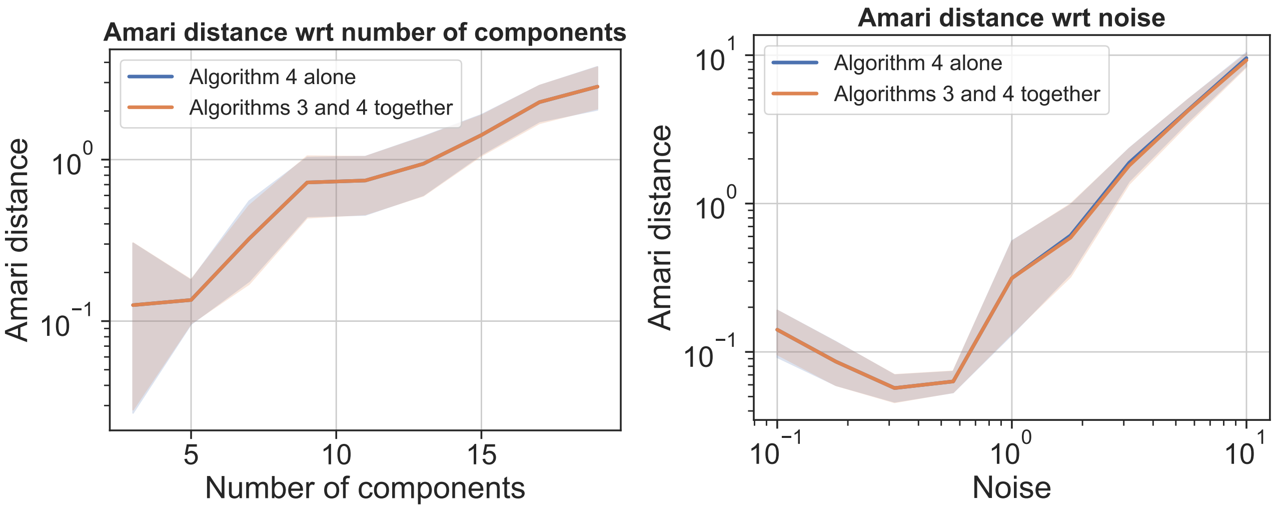
Figure 15: Plots that aim to show that algorithm 3 alone is not effective.

Algorithm 4 with and without algorithm 2: other plots

We show here two other plots that aims to tell if adding a second-step optimization of the delays is useful.

Figure 16a shows the Amari distance of MVICA and MVICAD for several number of components. In the model, there are 6 subjects and 400 samples (i.e. time-courses). Figure 16b shows the Amari distance of MVICA and MVICAD for several noise levels.

In both figures, we clearly observe that the two lines are merged. Thus, we can conclude as in section 4.1.2 that, using our data generation, the two solutions have approximately the same performance.



(a) Amari distance of MVICA and MVICAD with respect to the number of components.

(b) Amari distance of MVICA and MVICAD with respect to noise.

Figure 16: Comparison of the two solutions.