Allain Cédric

ENSAE 3e année
Stage de fin d'études
Année scolaire 2019/2020

Modeling Temporal Dependencies between Recurring Patterns in Electrophysiology (M/EEG)

Inria Saclay
Palaiseau

Maître de stage : Alexandre Gramfort
01/07/2020 - 30/11/2020

Contents

List of Figures

Résumé

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam. Maecenas ligula massa, varius a, semper congue, euismod non, mi. Proin porttitor, orci nec nonummy molestie, enim est eleifend mi, non fermentum diam nisl sit amet erat. Duis semper. Duis arcu massa, scelerisque vitae, consequat in, pretium a, enim. Pellentesque congue. Ut in risus volutpat libero pharetra tempor. Cras vestibulum bibendum augue. Praesent egestas leo in pede. Praesent blandit odio eu enim. Pellentesque sed dui ut augue blandit sodales. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae ; Aliquam nibh. Mauris ac mauris sed pede pellentesque fermentum. Maecenas adipiscing ante non diam sodales hendrerit.

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam. Maecenas ligula massa, varius a, semper congue, euismod non, mi. Proin porttitor, orci nec nonummy molestie, enim est eleifend mi, non fermentum diam nisl sit amet erat. Duis semper. Duis arcu massa, scelerisque vitae, consequat in, pretium a, enim. Pellentesque congue. Ut in risus volutpat libero pharetra tempor. Cras vestibulum bibendum augue. Praesent egestas leo in pede. Praesent blandit odio eu enim. Pellentesque sed dui ut augue blandit sodales. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam nibh. Mauris ac mauris sed pede pellentesque fermentum. Maecenas adipiscing ante non diam sodales hendrerit.

1 Introduction

TODO : parler de façon générale du stage, que c'est un stage de recherche, effectué à l'Inria Saclay. puis dire que l'on va d'abord présenter l'environnement de travail, puis le contexte général du stage

1.1 Inria and Parietal team

I did my internship at Inria Saclay, within the Parietal team, under the supervision of Alexandre Gramfort, senior research scientist, and Thomas Moreau, research scientist, both permanent members of the Parietal team. This section aims to briefly present the research institute of Inria, with a focus on the Parietal team.

Inria The Institute for Research in Computer Science and Automatic (Inria [1]) is a national institute for research in digital science and technology. It has been funded in 1967, within the framework of the "*Plan Calcul*" [2], and employs 3500 researchers and engineers often working in

1. Institut de recherche en informatique et en automatique
2. The Plan Calcul was a French governmental program launched in 1966 by President Charles de Gaulle designed to ensure the country's autonomy in information technology, and to develop European IT.

an interdisciplinary manner and in collaboration with industrial partners. Inria Saclay is one of the 9 research centres spread over the territory that compose Inria. Also, thanks to Inria Learning Lab, Inria designs MOOCs to disseminate knowledge in digital sciences and strengthen the dialogue between science and society.

Parietal team Within Inria, reasearchers are divided among 200 project-teams, the Parietal team being one of them. It is an Inria-CEA joint team part of the Neurospin research center, headed by Bertrand Thirion, that focuses on mathematical methods for statistical modeling of brain function using neuroimaging data (fMRI, MEG, EEG), with a particular interest in machine learning techniques, applications to human cognitive neuroscience, and scientific software development [3]. The different members of Parietal are committers to a variety of open-source projects such as Scikit-Learn, NiLearn and MNE-Python, among others.

1.2 General problematic

As previously mentioned, the data that are mainly used by the person working in the Parietal team come from electroencephalography (EEG) and magnetoencephalography (MEG), two methods used to record neuronal activity of the brain. The M/EEG data we are interested in for the following of the report are acquired during experiments with human subjects, that consist in a recording of the neuronal activity for a duration around 3 to 7 minutes. In the following Section 1.3, more info will be given on how the brain neuronal activity is produced and recorded

During the experiment, different type of external stimuli can be exercised on the subject, such as an auditory signal (left or right side, at different frequencies), a visual signal (left and right eye), the action to press a button, etc. The term external stimuli is opposed to an internal stimuli, where the subject would be ask to think at something or to think at doing something, without really doing it. In Section 3.2, the data and the different types of stimuli will be presented in more detail.

The main objective of this report is to model the temporal dependency between a specific stimulus and its neuronal response. To do so, we decompose the signal into recurring patterns, called *atoms*, using to dictionary learning and convolutional sparse coding, as it will be presented in Section 1.4. More specifically, we would like to model the latency between external stimuli and neuronal responses, or, in other words, model how a given stimulus can "activate" a specific atom in the brain, and thus, find the atom(s) that have a high probability to be linked with a specific external stimulus.

To respond to such a problematic, a new method will be introduced in Section 2, and it is based on temporal point process, more particularly on Hawkes processes. We called this new method *driven temporal point process*. Before presenting our method, a background on point processes with a focus on temporal point processes will be done in Section 1.5.

3. Source: presentation of Parietal, `https://team.inria.fr/parietal/`

1.3 Data in electrophysiology (M/EEG)

Electrophysiology refers to a branch of physiology that study the electrical and electrochemical phenomena that occur in the cells or tissues of living organisms and, in particular, in neurons and muscle fibres. In neuroscience, we are particularly interested in the electrical activity of the neurons and especially in the emission of post-synaptic potentials (PSP). In a neuron, PSP are nerve influx along the axon consecutive to ion exchange at the synapse level, as show in Figure 1.1, that changes the polarity of the cell membrane. As moving charges on a wire induces an electo-magnetic field, a group of neuron in the gray matter [4], the upper part of the brain that is composed essentially of the cell bodies and the dendritic tree of neurons, form a current generator [5] that produces an electrical and a magnetic field.

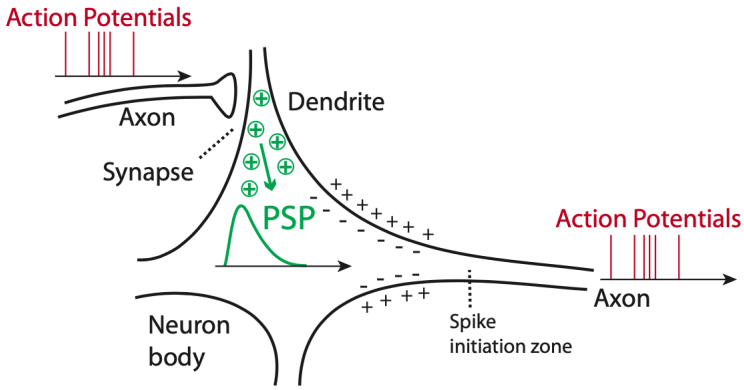


Figure 1.1 – Post-synaptic potentials in a neuron, consecutive to ion exchange
Source: A. Gramfort, MEEG course

Two methods are used to capture both the electrical and the magnetic field, respectively the electroencephalography (EEG) that captures differences in electric potential at the scalp, and the magnetoencephalography (MEG) that captures magnetic flux density outside the head, as shown in Figure 1.2. Both method record synchronised neural activity at a very high temporal resolution, about the millisecond [6], and have the advantage of being non-invasive, unlike ectro-corticography (ECoG) that uses electrodes placed directly on the exposed surface of the brain. Note that a large number of simultaneously active neurons, about 50 000, are needed to generate a measurable M/EEG signal [7].

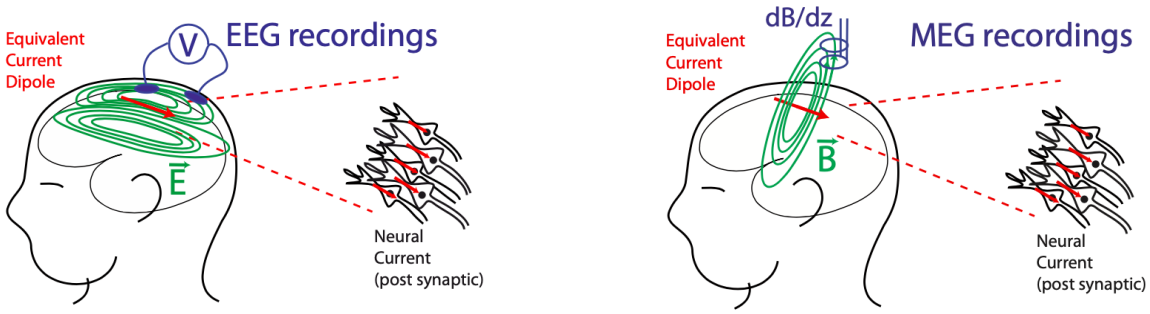
This high temporal resolution is what makes MEG and EEG attractive for the functional study of the brain. The spatial resolution, on the contrary, is rather poor as only a few hundred simultaneous data positions can be acquired simultaneously (about 300 to 400 sensors for MEG and up to 256 electrodes for EEG). With appropriate models and methods, localisation of activity

4. More precisely, the pyramidal cells in the cortex, that constitute approximately 80 % of the neurons of the cortex.

5. Alexandre Gramfort, *Functional Brain Imaging with MEG, EEG and sEEG*, Université Paris-Saclay, MEEG course, `http://bit.ly/meeg_course`

6. Sampling is often between 250 and 1000 Hz.

7. Saskia Helbling, *What are we measuring with M/EEG?*, Goethe University Frankfurt, SPM course, May 2014



(a) EEG recordings (b) MEG recordings

Figure 1.2 – Recordings of the electrical field (a) and the magnetic field (b).
Source: A. Gramfort, MEEG course

from MEG and EEG is nevertheless possible [8]. This is what is called the *inverse problem*, whose objective is to determine the current generators that produced the M/EEG measurements, as opposed to the *forward problem* whose objective is to predict the M/EEG surface signal to current dipoles in the brain.

Finaly, note that neural activity recorded via M/EEG measurements is fundamental to modern experimental neuroscience and in our understanding of human cognitive processes and certain pathologies, thereby motivating the development of computational tools for learning such signals from data. Such recordings consist of dozens to hundreds of simultaneously recorded signals, for duration going from minutes to hour [JLTSG17; DLTMJG18].

In this report we will mainly focus on the data recorded via MEG obtained in the course of experiments in which subjects are exposed to external stimuli. A more comprehensive presentation of the data we work with is done in the Section 3.2. In Python, M/EEG data are easily manipulable thanks to the `MNE` package [GLL+13].

1.4 M/EEG signals decomposition via dictionary learning

As previously explained in Section 1.3, the data recorded from one subject via M/EEG is complex. For example, for P sensors, called *channels*, over T timestamps, the signal observed is $X \in \mathbb{R}^{P \times T}$, that contains heavy noise bursts and have low signal-to-noise ratio, as most of neural signals [JLTSG17]. Thus, we cannot work directly with this result, we have to pre-process it in some way.

It is known that neural time-series data contain a wide variety of prototypical signal waveforms (atoms) that are of significant importance in clinical and cognitive research. One of the goals for analysing such data is hence to extract such 'shift-invariant' atoms, as events can happen at any instant [JLTSG17]. While alpha waves (8 Hz to 12 Hz) are known to closely resemble short sinusoids, and thus are revealed by Fourier analysis or wavelet transforms, there is an

8. Inria: MEG/EEG vs. other functional brain imaging modalities

evolving debate that electromagnetic neural signals are composed of more complex waveforms that cannot be analysed by linear filters and traditional signal representations [DLTMJG18].

To learn such atoms, one method is to use *dictionary learning*, that is a branch of signal processing and machine learning that aims at finding a frame (called dictionary) in which some training data admits a sparse representation. The sparser the representation, the better the dictionary [9]. Applied to brain signals, one method that works well is *convolutional sparse coding* (CSC) [JLTSG17; DLTMJG18; MG19]. This method aims at finding a dictionary of atoms and some associated activation vectors, in order to recover the original signal X by doing a convolution between the atoms and their sparse activation vectors, as shown in Figure 1.3.

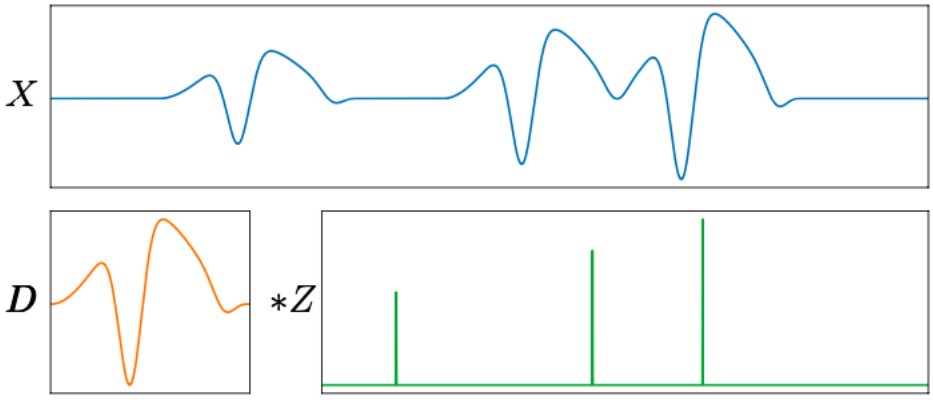


Figure 1.3 – Decomposition of a noiseless univariate signal X (blue) as the convolution $Z * \boldsymbol{D}$ between a temporal pattern $\boldsymbol{D}$ (orange) and a sparse activation signal Z (green).
Source: [MG19]

The optimisation problem is as follow [10]:

$$\min_{D_k, z_k^n} \sum_{n=1}^{N} \frac{1}{2} \left\| X^n - \sum_{k=1}^{K} z_k^n * D_k \right\|_2^2 + \lambda \sum_{k=1}^{K} \|z_k^n\|_1 \qquad (1.1)$$
$$\text{s.t.} \quad \|D_k\|_2^2 \leq 1 \text{ and } z_k^n \geq 0$$

where $\{X^n\}_{n=1}^N \in \mathbb{R}^{P \times T}$ are N observed multivariate signals, $\lambda > 0$ is the regularization parameter, $\{D_k\}_{k=1}^K \in \mathbb{R}^{P \times L}$ are the spatio-temporal atoms, $\{z_k^n\}_{k=1}^K \in \mathbb{R}^{\widetilde{T}}$ are K sparse signals of activations associated with X^n, with $\widetilde{T} := T - L + 1$, and where $z_k^n * D_k$ denotes the convolution between the two signals, obtained by by convolving every row of D_k by z_k^n.

In order to better account for the nature of brain signals, a rank-1 constraint is added to the dictionary: $D_k = u_k v_k^T \in \mathbb{R}^{P \times L}$, with $u_k \in \mathbb{R}^P$ being the pattern over channels and $v_k \in \mathbb{R}^L$ over time. Thus, the $\|D_k\|_2^2 \leq 1$ constraint in (1.1) is now replaced by $\|u_k\|_2^2 \leq 1$ and $\|v_k\|_2^2 \leq 1$. This rank-1 constraint comes from Maxwell's equations and the physical model of electrophysiological signals like EEG or MEG, where each sensor is supposed to instantly receive

9. Team Inria Panama - Dictionary learning: theory and algorithms
10. Note that we are in the case of 1D-convolution, in multivariate CSC.

a linear transformation of every sources, with a constant topographic map, i.e., a signal coming from the same source at two different times will be spread across the sensors with the same linear transformation. Thanks to this rank-1 constraint, the learned atoms have a spacial and a temporal pattern, as shown in Figure 1.4.

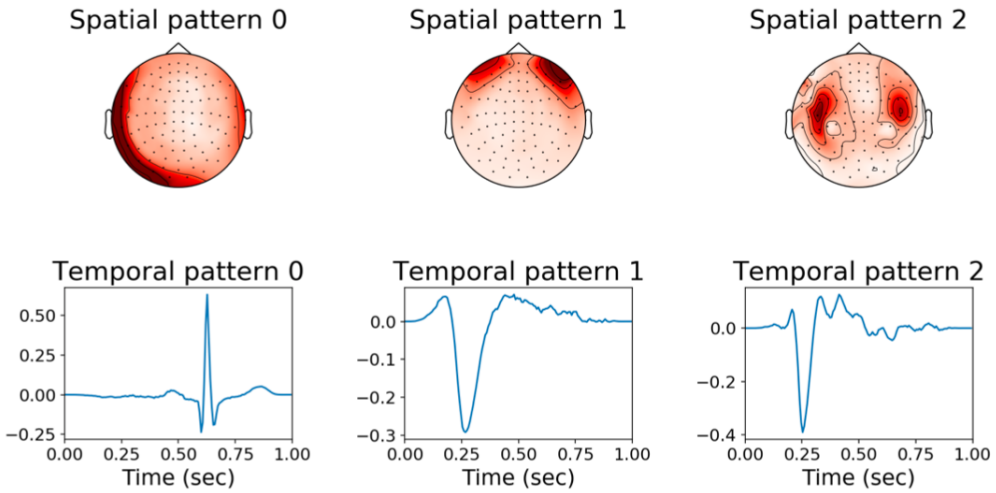


Figure 1.4 – Spacial (up) and temporal (down) representation of three atoms obtained by dictionary learning

Source: Example in `alphacsc` package's documentation

In python, the package `alphacsc` makes it possible to carry out such an operation. In practice, we do not have N signals, as the brain of every person is different, it would make no sense to consider that the atoms of one subject are strictly identical to another. Thus, we split the recorded signal $X \in \mathbb{R}^{P \times T}$ into several smaller signals, in order to take advantage of the factorised computations describe in [DLTMJG18] that speed up computational time, and eventually to be able to distribute computations [MG19] [11]. Once the original signal is split into N smaller signals, the dictionary of atoms D and their associated activations vectors z_k^n are learned. Finally, an ultimate action is done: with the learned dictionary D and with the original signal X, the matrix $z \in \mathbb{R}^{K \times \widetilde{T}}$ is learned solving (1.1).

1.5 Background on point processes

In this section, we will give a short introduction on point processes, and particularly on temporal point processes with a focus on Hawkes processes. Further details can be found in [DVJ03; DVJ07]. The sub-section 1.5.1 comes from the PhD thesis of Massil Achab [Ach17].

1.5.1 Definitions

A point process is a random element whose values are point patterns on a set S, a locally compact metric space equipped with its Borel σ-algebra $\mathscr{B}$. Let X_S be the set of locally fi-

11. Note that distributed CSC is not implemented yet into `alphacsc`.

nite counting measures on S, and $\mathcal{N}_S$ the smallest σ-algebra on X_S such that all point counts $f_B : X_s \to \mathbb{N}, \omega \mapsto \#(\omega \cap B)$ are measurable for B relatively compact in $\mathscr{B}$, where $\#A$ denotes the cardinality of the set A. A point process on S is a measurable map ξ from a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ to the measurable space $(X_S, \mathcal{N}_S)$.

Every realisation of a point process ξ can be written as $\xi = \sum_{i=1}^{n} \delta_{X_i}$, where δ is the Dirac measure, n is a integer-valued random variable and X_i's are random elements of S. A point process can be equivalently represented by a counting process $N(B) \coloneqq \int_B \xi(x)\,\mathrm{d}x$ which basically is the number of event in each Borel subset $B \in \mathscr{B}$. The mean measure M of a point process ξ is a measure on S that assigns to every $B \in \mathscr{B}$ the expected number of event of ξ in B, i.e., $M(B) \coloneqq \mathbb{E}[N(B)]$ for all $B \in \mathscr{B}$.

1.5.2 Temporal point processes

A temporal point process is a stochastic, or random, process composed of a time series of binary events that occur in continuous time [12]. However, on the contrary of time series, temporal point processes can study multiple time scales at once [Bom19].

In this particular case, S is the time interval $[0\,;T)$, equipped with the Borel σ-field of the real line $\mathscr{B}(\mathbb{R})$. Here, a realisation of a point process is simply a set of time points: $\xi = \sum_{i=1}^{n} \delta_{t_i}$. With a slight abuse of notation, we associate to the set of distinct random timestamps $\xi = \{t_1, \ldots, t_n\}$ occurring before T, the counting process $N_t = \sum_{t_i \in \xi} \mathbb{1}_{\{t_i \leq t\}}$, which is then simply the number of points in the time interval $(0\,;t]$. This counting process is a random process which evolves over time by jumps of size 1. Studying temporal point processes consists in analysing when this jumps occur. The *conditional intensity* function $\lambda(t|\mathscr{F}_t)$ is the usual way to characterise temporal point processes where the present depends on the past. It is defined as the expected infinitesimal rate at which events are expected to occur after t given the information $\mathscr{F}_t$ available up to (but not including) time t, i.e., the history of the counting process N_t prior to t. Namely,

$$\lambda(t|\mathscr{F}_t) = \lim_{\mathrm{d}t \to 0} \frac{\mathbb{P}(N_{t+\mathrm{d}t} - N_t = 1|\mathscr{F}_t)}{\mathrm{d}t} \tag{1.2}$$

where $\mathscr{F}_t = \{t_i,\ t_i < t, i = 1, \ldots, n\}$ is the natural filtration of the process. The conditional intensity function is sometimes denoted $\lambda^*(t)$.

As $\mathrm{d}N_t \coloneqq N_{t+\mathrm{d}t} - N_t \in \{0, 1\}$ can only increase by one event at each $\mathrm{d}t$, it readily follows that $\mathbb{P}(\mathrm{d}N_t = 1|\mathscr{F}_t) = \lambda^*(t)\mathrm{d}t$ and

$$\mathbb{E}[\mathrm{d}N_t|\mathscr{F}_t] = 1 \times \mathbb{P}(\mathrm{d}N_t = 1|\mathscr{F}_t) + 0 \times \mathbb{P}(\mathrm{d}N_t = 0|\mathscr{F}_t) = \lambda^*(t)\mathrm{d}t \tag{1.3}$$

Hence, we can also think of the conditional intensity function $\lambda^*(t)$ as an instantaneous rate of events per time of unit.

12. Liam Paninski, Statistical analysis of neural data, Fall 2019, *Chapter 2: Introduction to Point Processes* - Columbia Statistics

The *homogeneous Poisson process* is the most simple temporal point process, which assumes that the events arrive at a constant rate, which corresponds to a constant intensity function $\lambda\left(t|\mathscr{F}_t\right) = \lambda^*(t) = \lambda > 0$. In other words, it describe a phenomenon with no memory and a constant probability of occurrence in which $N_{t+\Delta t} - N_t$ follows a Poisson distribution of parameter Δt for any $\Delta t > 0$. For this process, $\forall B \in \mathscr{B}\left(\mathbb{R}\right), M\left(B\right) = \lambda\left|B\right|$, where $|\cdot|$ is the Lebesgue measure on $(S, \mathscr{B}\left(\mathbb{R}\right))$.

The *inhomogeneous Poisson process* is a more general process, for which the conditional intensity function is not constant as it depends on t but *not* on the history, i.e. $\lambda\left(t|\mathscr{F}_t\right) = \lambda^*(t) = \lambda(t)$. For this process, $M\left(B\right) = \int_B \lambda(x)\,\mathrm{d}x$, for all $B \in \mathscr{B}\left(\mathbb{R}\right)$.

Let us denote $f^*(t) = f\left(t|\mathscr{F}_t\right)$ the conditional probability density function of the inter-event time, i.e., the probability that the next event will occur during the interval $[t\,; t + \mathrm{d}t)$ conditioned on the history $\mathscr{F}_t$. Let us also denote $F^*(t) = F\left(t|\mathscr{F}_t\right) = \mathbb{P}\left(t_n \leq t_{n+1} \leq t|\mathscr{F}_t\right) = \int_{t_n}^{t} f^*(\tau)\,\mathrm{d}\tau$ the conditional cumulative density function, i.e., the probability that the next event will occur before time t conditioned on the history $\mathscr{F}_t$, where here t_n is the last event in $\mathscr{F}_t$, i.e., the last event before time t. Finally, we denote $S^*(t) = 1 - F^*(t) = \mathbb{P}\left(t_{n+1} \geq t|\mathscr{F}_t\right)$ the complementary cumulative distribution, also called the survival function, i.e., the probability that the next event will not occur before time t conditioned on the history $\mathscr{F}_t$ [DUGR19]. Now,

$$
\begin{aligned}
\lambda^*(t) &= \lim_{\mathrm{d}t \to 0} \frac{\mathbb{P}\left(t \leq t_{n+1} \leq t + \mathrm{d}t | t_{n+1} > t\right)}{\mathrm{d}t} \\
&= \lim_{\mathrm{d}t \to 0} \frac{1}{\mathrm{d}t} \frac{\mathbb{P}\left(t \leq t_{n+1} \leq t + \mathrm{d}t\right)}{\mathbb{P}\left(t_{n+1} > t\right)} \\
&= \lim_{\mathrm{d}t \to 0} \left(\frac{1}{\mathrm{d}t} \frac{f^*(t)\mathrm{d}t}{S^*(t)} + o(1) \right) \\
&= \frac{f^*(t)}{S^*(t)} \\
&= -\frac{1}{S^*(t)} \frac{\mathrm{d}S^*(t)}{\mathrm{d}t} \\
&= -\frac{\mathrm{d}\log S^*(t)}{\mathrm{d}t}
\end{aligned}
\tag{1.4}
$$

By integrating the left and right hand sides in the above equation, we have that

$$
\int_{t_n}^{t} \lambda^*(\tau)\,\mathrm{d}\tau = \int_{t_n}^{t} -\frac{\mathrm{d}\log S^*(\tau)}{\mathrm{d}\tau}\,\mathrm{d}\tau = -\log S^*(t) + \underbrace{\log S^*(t_n)}_{=0} \tag{1.5}
$$

and thus,

$$
S^*(t) = \exp\left(-\int_{t_n}^{t} \lambda^*(\tau)\,\mathrm{d}\tau \right) \tag{1.6}
$$

Finally, we get that

$$
f^*(t) = \lambda^*(t) \exp\left(-\int_{t_n}^{t} \lambda^*(\tau)\,\mathrm{d}\tau \right) \tag{1.7}
$$

1.5.3 Poisson process and likelihood function

This section is taken and adapted from [DVJ03, chap. 2, p. 19-23] and aims to give a more in-depth presentation of the Poisson processes, with a focus on the computation of the likelihood function, as it is a crucial information for the repport.

The stationary Poisson process [13], on the line is completely defined by the following equation, in which we use $N(a_i\,;b_i]$ to denote the number of events of the process falling in the half-open interval $(a_i\,;b_i]$ with $a_i < b_i \leq a_i + 1$:

$$\mathbb{P}\left(N(a_i\,;b_i] = n_i, i = 1, \ldots, k\right) = \prod_{i=1}^{k} \frac{\left[\lambda\left(b_i - a_i\right)\right]^{n_i}}{n_i!} e^{-\lambda(b_i - a_i)} \tag{1.8}$$

This definition embodies three important features:

- the number of points in each finite interval $(a_i\,;b_i]$ has a Poisson distribution of parameter λ;
- the numbers of points in disjoint intervals are independent random variables; and
- the distributions are stationary: they depend only on the lengths $b_i - a_i$ of the intervals.

The likelihood of a finite realisation of a Poisson process may be defined as the probability of obtaining the given number of observations in the observation period, times the joint conditional density for the positions of those observations, given their number.

Suppose that there are N observations on $(0\,;T]$ at time points $t_1, \ldots, t_N$. From 1.8, we can write down immediately the probability of obtaining single events in $(t_i - \Delta\,;t_i]$ and no points on the remaining part of $(0\,;T]$. Let A and B be respectively those events, namely,

$$A = \{N(t_i - \Delta\,;t_i] = 1, i = 1, \ldots, N\}$$

and

$$B = \{N(0\,;t_1 - \Delta] = 0, N(t_N\,;T] = 0, N(t_i\,;t_{i+1} - \Delta] = 0, i = 1, \ldots, N-1\}$$

$$\begin{aligned}
\mathbb{P}(A \cap B) &= \prod_{i=1}^{N} \left(\lambda\Delta e^{-\lambda\Delta}\right) \times e^{-\lambda(t_1 - \Delta)} \times e^{-\lambda(T - t_N)} \times \prod_{i=1}^{N-1} e^{-\lambda(t_{i+1} - \Delta - t_i)} \\
&= \prod_{i=1}^{N} (\lambda\Delta) \times e^{-\lambda\Delta N} \times e^{-\lambda(T - N\Delta)} \\
&= e^{-\lambda T} \prod_{i=1}^{N} \lambda\Delta \\
&= \lambda^N \Delta^N e^{-\lambda T}
\end{aligned}$$

13. What we previously called the homogeneous Poisson process.

Dividing by Δ^N and letting $\Delta \to 0$, to obtain the density, we find as the required likelihood function

$$L_{(0\,;T]}\left(N; t_1, \ldots, t_n\right) = \lambda^N e^{-\lambda T} \tag{1.9}$$

We can extend this result to a Poisson process with time-varying rate $\lambda(t)$, commonly called the *nonhomogeneous* or *inhomogeneous* Poisson process. The process can be defined exactly as in 1.8, ,with the quantities $\lambda\left(a_i\,; b_i\right] = \int_{a_i}^{b_i} \lambda \,\mathrm{d}x$ replaced wherever they occur by quantities

$$\Lambda\left(a_i\,; b_i\right] = \int_{a_i}^{b_i} \lambda(x)\,\mathrm{d}x$$

called the *compensator* of the point process. Thus, the joint distributions are still Poisson, and the independence property still holds. The likelihood function takes the more general form

$$\begin{aligned} L_{(0\,;T]}\left(N; t_1, \ldots, t_n\right) &= e^{-\Lambda(0\,;T]} \prod_{i=1}^{N} \lambda(t_i) \\ &= \exp\left(-\int_0^T \lambda(t)\,\mathrm{d}t + \sum_{i=1}^{N} \log \lambda(t_i)\right) \\ &= \exp\left(-\int_0^T \lambda(t)\,\mathrm{d}t + \int_0^T \log \lambda(t) N\left(\mathrm{d}t\right)\right) \end{aligned} \tag{1.10}$$

Note that this result could also be obtained by using the Equation (1.7).

1.5.4 Hawkes processes

In this section, we give the main definitions and properties of Hawkes processes and multivariate Hawkes processes and set the notations that will be used if the in the rest of the report. Hawkes processes [Haw71; HO74] are temporal point processes in which the intensity depends on the process history with an excitation mechanism. They can be understood as the equivalent of auto-regressive time series models (AR) but in continuous time. This allows to study cross causality that might occur in one or several events series [Bom19].

An Hawkes process is thus defined by a history dependent intensity λ defined as follows:

$$\lambda\left(t|\mathscr{F}_t\right) = \psi\left(\mu + \int_{-\infty}^{t} \phi(t-s)\,\mathrm{d}N_s\right) \tag{1.11}$$

where

$$\int_{-\infty}^{t} \phi(t-s)\,\mathrm{d}N_s = \sum_{i, t_i < t} \phi(t - t_i) \tag{1.12}$$

The $\mu \geq 0$ is referred as the *baseline intensity*[14] and it corresponds to the exogenous intensity of the considered events. The function $\phi(\cdot) : \mathbb{R}^+ \to \mathbb{R}$ is called the *kernel function*, or the

14. Some authors may also call it the background intensity.

transfer function [CSSBW17], and quantifies over time and in magnitude the influence of past events. Note that the occurrence of each event t_i increases the intensity by a certain amount, determined by the kernel, making the intensity history dependent and a stochastic process by itself [DUGR19]. If the *link function* ψ on the right-hand side of Eq. 1.11 is non-linear, then $\lambda(t)$ is the intensity of a non-linear Hawkes process [BM96]. In what follows, we only consider the case where ψ is the identity function.

Multivariate Hawkes process We can extend the univariate Hawkes process to model the interactions of $K \geq 1$ temporal point processes, called *nodes*.

Namely, it models timestamps $\left\{t_k^{(i)}\right\}_{k\geq 1}$ of nodes $i = 1, \ldots, K$ associated with a multivariate counting process $N_t = \left[N_t^{(1)}, \ldots, N_t^{(K)}\right]$. Note that for all nodes $i = 1, \ldots, K$, we still have that all of its timestamps $t_k^{(i)}$ occur in the time interval $[0\,;T]$. The excitation dynamic between the nodes is encompassed by the auto-regressive structure of the conditional intensity. For component $N_t^{(i)}$ it writes

$$\lambda_i\left(t|\mathscr{F}_t\right) = \mu_i + \sum_{j=1}^{K} \int_{-\infty}^{t} \phi_{i,j}(t-s)\,\mathrm{d}N_s^{(j)} \tag{1.13}$$

where $\phi_{i,j}(t)$ quantifies the excitation rate of an event of type j on the arrival rate of events of type i after a time lag t. In general it is assumed that each kernel is causal and positive, meaning that Hawkes processes can only account for mutual excitation effects since the occurrence of some event can only increase the future arrival intensity of other events. If the kernels are integrable, each entry of the $K \times K$ matrix $(\Phi)_{i,j} = \int_0^T \phi_{i,j}(t)\,\mathrm{d}t$ denotes the expected number of events of type i directly triggered by an event of type j.

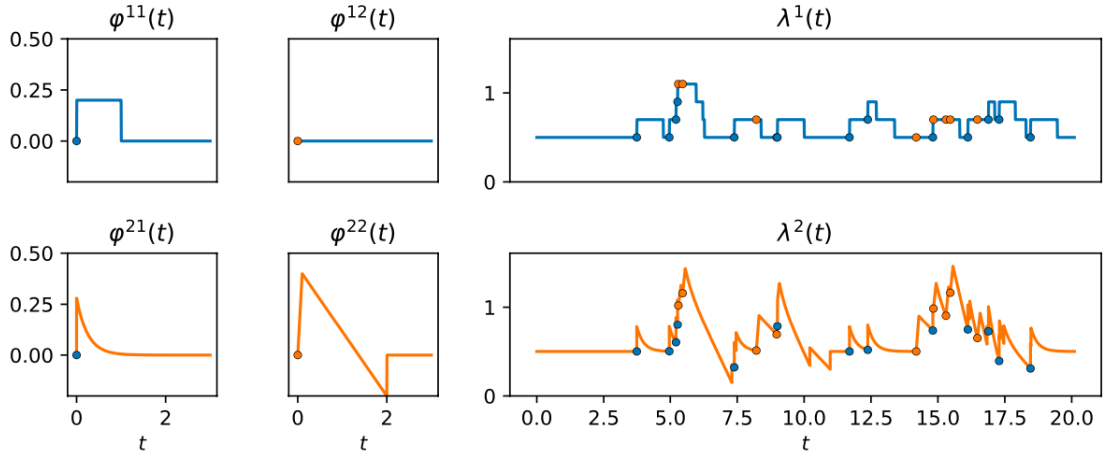


Figure 1.5 – A realisation of a 2 nodes multivariate Hawkes process using `Tick` package. The four excitation kernels are shown on the left hand side. The intensities are displayed on the right hand side (against time, up to time 20), where events are represented by coloured dots (blue corresponding to node 1 and orange to node 2).

Source: [Bom19]

Kernels parametrisation The main parametric model is the so-called *exponential kernel*, in which the kernels have the following form:

$$\phi_{i,j}(t) = \alpha_{i,j}\beta \exp(-\beta t), \quad \alpha_{i,j} > 0, \beta > 0 \tag{1.14}$$

In this model the integral matrix $\Phi = (\alpha_{i,j})_{1 \leq i,j \leq K}$ and $\beta > 0$ is a memory parameter. A more general approach is the *sum of exponentials kernels* [LV14], namely

$$\phi_{i,j}(t) = \sum_{u=1}^{U} \alpha_{i,j}^{(u)} \beta^{(u)} \exp\left(-\beta^{(u)} t\right), \quad \alpha_{i,j}^{(u)} > 0, \beta^{(u)} > 0 \tag{1.15}$$

Similarly, we can define the *gaussian kernel* and the *sum of gaussians kernels*. In python, the `Tick` package allow to easily manipulate Hawkes process with exponential and gaussian kernels [Bom19; BBGP17]. Other kernel functions are presented in [MZ14].

2 Developed method: driven temporal point processes

In the Hawkes processes model presented previously in Section 1.5.4, the assumption is made that all the nodes might have an influence on all the other nodes, in addition of themselves. Recall that in our application to electrophisiology, and in particular to M/EEG data, we would like to capture the potential influence of an external stimulus on the activation of a specific recurring pattern in the cortex, called *atom*. It is easy to understand in that case that the influence can only goes one way, from the stimulus process to the atom process. Furthermore, the stimulus process is determinist as it is controlled by the experimentator and thus cannot be modeled as a classical temporal point process. If we would like to implement such a model using `Tick`, three of the four kernels would be set to the null function, as the assumption is made that the atom do not have a self-excitatory behaviour, thus only leaving the kernel connecting the stimulus and the atom as a non-null function.

For those reasons, we derived a model from the Hawkes processes model, in which a determinist point process has a potential influence on a temporal point process. By "having an influence on", we mean that an event on the determinist point process increases the probability to have an event on the temporal point process. We call this approach *driven temporal point processes*, as the behaviour of a temporal point process might be driven, influenced, by a determinist point process.

2.1 Data and notations

TODO : Faire la distinction entre le modèle théorique (processus guidé et processus "guideur"), et l'application : les z_k sont les processus guidés, et dans l'application on les obtient de telle manière [...]

As explained before, the *driven temporal point processes* approach consists in two sets of timestamps, one for

TODO: dans l'appllication M/EEG, dire que les multiples signaux reçus par les capteurs sont décomposés en K recurring patterns, où K est un hyperparamètre (TODO: dans les fais, comment est déterminé K, est-ce qu'on donne un nombre assez élevé afin d'être sûr de capter tous les atomes intéressants ?) et que l'on fait plusieurs type de stimulis, avec exemples :

- $\{z_k\}_{k=1,\dots,K}$: the family of the K atoms' activation vector (an example of an activation vector is shown in Figure 1.3). $\forall k = 1, \dots, K, z_k \in \mathbb{R}_+^{\mathbf{T}}$, where $\mathbf{T}$ is the total number of possible timestamps. $\mathbf{T} \coloneqq \lfloor T \times s_{freq} \rfloor$, where T is the total duration of the experimentation (e.g., $T = 3\,\text{min}$), and s_{freq} is the sampling frequency for the collect of MEG data (e.g., $s_{freq} = 150\,\text{s}^{-1}$), and where $\lfloor x \rfloor$ denotes the bigger integer smaller or equal to x. We define $\mathcal{A}_k \coloneqq \left\{ t \times s_{freq}^{-1},\ z_k(t) \geq \tau, t = 1, \dots, \mathbf{T} \right\}$ the set of all the timestamps of atom k's activations, where τ is a pre-determined threshold used to filter out insignificant activations (e.g., $\tau = 0.7 \times 10^{-10}$).

- $\{e_p\}_{p=1,\dots,P}$: the family of the P types of tasks performed during the experimentation

$$\forall t = 1, \dots, \mathbf{T}, \quad e_p(t) = \begin{cases} 1 & \text{if a task of type } p \text{ has occurred at the } t\text{-th timestamp} \\ 0 & \text{otherwise} \end{cases}$$

- $t^{(p)} \coloneqq \left\{ t \times s_{freq}^{-1},\ e_p(t) = 1, t = 1, \dots, \mathbf{T} \right\}$ denotes all the timestamps when a task of type p occurred. We thus have $t^{(p)} = \left\{ t_1^{(p)}, \dots, t_{n_p}^{(p)} \right\}$ sorted, i.e., $t_1^{(p)} < t_2^{(p)} < \cdots < t_{n_p}^{(p)}$, where $n_p = \#t^{(p)}$ is the total number of task p that has occurred during the experimentation, as $\#A$ denotes the cardinality of the set A.

TODO: afin de préciser une dernière fois, les e_p sont les determinist point process, qui drivent les z_k. Dans les faits, à la fin de l'expérience, on a toutes les données, les vecteurs z_k sont tous connus. Notre but est alors de déterminer si, pour un coupe (e_p, z_k) donné, si le vecteur d'activation a été influencé au stimulus, et donc si l'atome k est "lié" au stimulus p. A noter que l'on ne peux exhiber qu'une relation de corrélation et non pas de causalité, notamment dû au fait que l'on ne contrôle pas tout l'environnement, il peut donc exister des facteurs exhogènes non contrôlés par le modèle qui peuvent être à l'origine des activations sur z_k, et non pas e_p. Ainsi, dans un premmier temps, on ne considère un atome qu'à la lumière d'un seul stimulus à la fois.

2.2 Model with truncated gaussian kernel

TODO : partir de la forme général du kernel, avec ϕ, comme écrit dans Hawkes, puis dire que l'on choisit un kernel normal tronqué de façon à modéliser la latency entre le stimulus et son éventuelle réponse neuronale. Rajouter que pour l'instant on essaie de modéliser le lien entre un simulus p et un atome k, de façon à voir si ce dernier lui est lié ou non (on considère que toutes les autres influences sur l'atome k sont dans sa baseline). Dire aussi que l'on fait l'hypothèse que, à l'inverse des processus de Hawkes avec un noyau exponentiel, où tous les events passés influent sur l'intensité présente, dans notre modèle seul le dernier event du driver à l'instant t peut avoir une influence sur l'intensité du temporal point processes qui lui est éventuellement lié (on dit toujours éventuellement cas on essaie justement d'exhiber une telle relation).

When driven by e_p, the dynamic of z_k's counting process $N_t^{(k)} \coloneqq \sum_{t_k \in \mathcal{A}_k} \mathbb{1}_{\{t_k \leq t\}}$ can be describe by its intensity: $\forall t \in [0\,;T]$,

$$\lambda_{k,p}(t) \equiv \lambda_k \left(t \middle| \mathscr{F}_t^{(p)} \right) \coloneqq \lim_{\mathrm{d}t \to 0} \frac{\mathbb{P}\left(N_{t+\mathrm{d}t}^{(k)} - N_t^{(k)} = 1 \middle| \mathscr{F}_t^{(p)} \right)}{\mathrm{d}t} \tag{2.1}$$

where $\mathscr{F}_t^{(p)} = \left\{ t_i \in t^{(p)},\ t_i < t \right\}$ is the information available on e_p up to, but not including, t.

Our model As mentioned above, the shape of the intensity is derived from the Hawkes process model, with a baseline intensity and a kernel function. Note that here again, we consider the link function to be the identity function. Namely,

$$\lambda_{k,p}(t) = \mu_k + \alpha_{k,p}\kappa_{k,p}(t - t_*^{(p)}(t))\mathbb{1}_{\left\{t \geq t_1^{(p)}\right\}} \tag{2.2}$$

where $t_*^{(p)}(t) \coloneqq \max\left\{t',\ t' \in t^{(p)}, t' \leq t\right\}$ denotes the timestamp of the last task of type p occurred before ($\leq$) time t, and where $\kappa_{k,p}$ is the probability density function of a truncated gaussian distribution of mean $m_{k,p}$ and standard deviation $\sigma_{k,p}$, with truncation values $[a\,;b]$, noted $\mathcal{N}_{[a,b]}\left(m_{k,p}, \sigma_{k,p}^2\right)$, namely,

$$\kappa_{k,p}(x) \equiv \kappa(x; m_{k,p}, \sigma_{k,p}, a, b) = \frac{1}{\sigma_{k,p}} \frac{\phi\left(\frac{x-m_{k,p}}{\sigma_{k,p}}\right)}{\Phi\left(\frac{b-m_{k,p}}{\sigma_{k,p}}\right) - \Phi\left(\frac{a-m_{k,p}}{\sigma_{k,p}}\right)} \mathbb{1}_{\{a \leq x \leq b\}} \tag{2.3}$$

where

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right)$$

is the probability density function of the standard normal distribution, and

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{1}{2}t^2\right) \mathrm{d}t$$

is its cumulative distribution function.

Note that by definition, if $b = \infty$, then $\Phi\left(\frac{b-m_{k,p}}{\sigma_{k,p}}\right) = 1$, and similarly, if $a = -\infty$, then $\Phi\left(\frac{a-m_{k,p}}{\sigma_{k,p}}\right) = 0$. An example of a truncated normal distribution is presented in Fig. 2.1.

2.3 Hypotheses

In addition to the theoretical model presented above, we do some hypotheses due to the fact that we place ourselves in a particular frame of study, namely the study of M/EEG data. This therefore involves making some specific assumptions in order to take into account the physical reality that applies. In addition, certain assumptions facilitate the calculations, without prejudice to the veracity of the model. Those hypotheses are as follow:

1. In Equation (2.2), the kernel's purpose is to capture the delay of the neurological response following an external stimulation. This kernel is chosen to be a truncated normal of shape parameter m and σ, respectively representing the mean and the standard deviation of such a delay. Precisely, we suppose that we have:

$$a \leq m \leq b \tag{2.4}$$

where a and b are respectively the lower and upper truncation values.

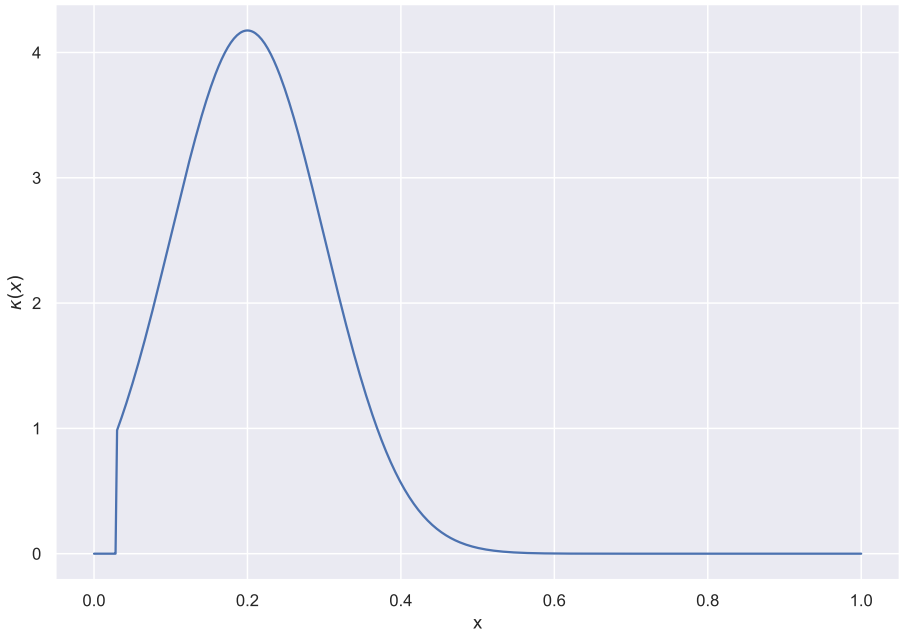


Figure 2.1 – Truncated Normal distribution with $m = 0.2$, $\sigma = 0.1$, $a = 0.03$ and $b = 0.8$.

2. The shape parameters of the kernel depend on the couple atom/task, i.e. $m \equiv m_{k,p}$ and $\sigma \equiv \sigma_{k,p}$, in order to better capture the specificity of a task and its neurological response (in time and in space).

3. The truncation values a and b are the same for every atom k and every task p. The idea behind this hypothesis is that the neurological response cannot, in any case, be shorter than a (e.g., $a = 30 \times 10^{-3}$ s) nor be longer than b (e.g., $b = 800 \times 10^{-3}$ s)

4. The tasks are far enough apart from each other such that the kernel is null right before every tasks, i.e.:

$$t_{i+1}^{(p)} - t_i^{(p)} > b, \quad \forall p = 1, \dots, P, \quad \forall i = 1, \dots, (n_p - 1) \tag{2.5}$$

   In the M/EEG context, we say that the *inter stimulus interval* (ISI) is larger than b.

   Note that with this hypothesis, the intensity can be rewritten similar as a auto-regressive point process:

$$\lambda_{k,p}(t) = \mu_k + \alpha_{k,p} \sum_{t_i^{(p)} \leq t} \kappa_{k,p}(t - t_i^{(p)}) \tag{2.6}$$

indeed, as $t \geq t_*^{(p)}(t) > \cdots > t_2^{(p)} > t_1^{(p)}$, then $\forall i, t_i^{(p)} < t_*^{(p)}(t)$

$$\begin{aligned} t - t_i^{(p)} &= t - t_{i+1}^{(p)} + t_{i+1}^{(p)} - t_i^{(p)} \\ &> t - t_{i+1}^{(p)} + b \\ &\geq b, \quad \text{as } t - t_{i+1}^{(p)} \geq 0 \end{aligned}$$

and as by definition, $\forall x \geq b, \kappa_{k,p}(x) = 0$, thus $\kappa_{k,p}(t - t_1^{(p)}) = 0$, and finally,

$$\begin{aligned} \sum_{t_i^{(p)} \leq t} \kappa_{k,p}(t - t_i^{(p)}) &= \kappa_{k,p}(t - t_1^{(p)}) + \kappa_{k,p}(t - t_2^{(p)}) + \cdots + \kappa_{k,p}(t - t_*^{(p)}(t)) \\ &= 0 + 0 + \cdots + \kappa_{k,p}(t - t_*^{(p)}(t)) \end{aligned}$$

5. The experimentation ends after every possible neurological response driven by a task, i.e.,

$$\forall p = 1, \ldots, P, \quad T > t_{n_p}^{(p)} + b \tag{2.7}$$

$$\Leftrightarrow T > \max_{p=1,\ldots,P} t_{n_p}^{(p)} + b$$

6. An atom is driven by only one task, i.e., we only consider the effect of one task for an atom in the model, as opposed to including several tasks.

7. The baseline intensity μ_k is fixed in time. The baseline coefficient aims at capturing every exhogeneous sources of activation that are not controlled by the model, i.e., all other sources of activation apart from the considered task. Other sources can be other tasks, other atoms, spontaneous activations or even sources we are not aware of. Finally, it should be pointed out that, unlike Hawkes processes, we do not model any self-excitatory behaviour, i.e., an activation of the considered atom does not increase by itself the probability to have another activation.

These assumptions can easily be called into question, in particular their restrictive character on the control of endogenous sources of activation, as we only consider an atom along a unique task. That is why they we be discussed later on, and a brief overview of an extension of our model will be presented in Section 2.6.

2.4 Simulation

TODO: présenter pourquoi on utilise cet algorithme, citer aussi le papier pour l'algo de simulation des processus de hawkes, et redire pourquoi ce n'est pas exactement notre cas (on est déterministe sur un des processus temporel). Mettre l'algorithme de simulation des processus de Hawkes (thinning algo, Ogata) en annexe ?

[Oga81; Che16; LS79]

Algorithm 1: [LS79], p.7, Algorithm 1, One-dimensional nonhomogeneous Poisson process

Data: $\lambda(t)$, T
initialize $n = m = 0$, $t_0 = s_0 = 0$, $\bar{\lambda} = \max_{0 \leq t \leq T} \lambda(t)$;
while $s_m \leq T$ **do**
  Draw $u \sim \mathcal{U}nif_{[0\,;1]}$;
  $w \leftarrow -\ln u / \bar{\lambda}$;
  $s_{m+1} \leftarrow s_m + w$;
  Draw $D \sim \mathcal{U}nif_{[0\,;1]}$;
  if $D \leq \lambda(s_{m+1})/\bar{\lambda}$ **then**
    $t_{n+1} \leftarrow s_{m+1}$;
    $n \leftarrow n + 1$;
  end
  $m \leftarrow m + 1$;
end
if $t_n \leq T$ **then**
  return $\{t_k\}_{k=1,2,\ldots,n}$;
else
  return $\{t_k\}_{k=1,2,\ldots,n-1}$;
end

Remark Thanks to hypotheses 2.4 and 2.5, it is easy to compute $\bar{\lambda}$:

$$\bar{\lambda} = \max_{0 \leq t \leq T} \lambda(t) = \mu + \kappa(m) \tag{2.8}$$

TODO: calculer la complexité de cet algorithme, ou du moins ça complexité maximale, comme fait dans la thèse de Martin

2.5 EM-based algorithm

TODO: dire ce que l'on veut faire, Inspired by [LM11; XFZ16]

2.5.1 Computations of the coefficients update

Rewriting of the kernel κ

$$\begin{aligned}
\kappa(x; m, \sigma, a, b) &= \frac{1}{\sigma} \frac{\phi\left(\frac{x-m}{\sigma}\right)}{\Phi\left(\frac{b-m}{\sigma}\right) - \Phi\left(\frac{a-m}{\sigma}\right)} \mathbb{1}_{\{a \le x \le b\}} \\
&= \frac{1}{\sigma} \frac{\exp\left(-\frac{1}{2}\frac{(x-m)^2}{\sigma^2}\right)}{\int_{\frac{a-m}{\sigma}}^{\frac{b-m}{\sigma}} \exp\left(\frac{-t^2}{2}\right) \mathrm{d}t} \mathbb{1}_{\{a \le x \le b\}} \\
&= \frac{\exp\left(-\frac{1}{2}\frac{(x-m)^2}{\sigma^2}\right)}{\int_a^b \exp\left(-\frac{1}{2}\frac{(u-m)^2}{\sigma^2}\right) \mathrm{d}u} \mathbb{1}_{\{a \le x \le b\}}, \quad t = \frac{u-m}{\sigma} \\
&= \frac{\exp\left(-\frac{1}{2}\frac{(x-m)^2}{\sigma^2}\right)}{C\left(m, \sigma, a, b\right)} \mathbb{1}_{\{a \le x \le b\}}
\end{aligned}$$

where

$$\begin{aligned}
C\left(m, \sigma, a, b\right) &\coloneqq \int_a^b \exp\left(-\frac{1}{2}\frac{(u-m)^2}{\sigma^2}\right) \mathrm{d}u \\
&= \sigma\sqrt{2\pi}\left(\Phi\left(\frac{b-m}{\sigma}\right) - \Phi\left(\frac{a-m}{\sigma}\right)\right)
\end{aligned}$$

and similarly, we denote by subscripts the partials derivatives:

$$\begin{aligned}
C_m\left(m, \sigma, a, b\right) &\coloneqq \frac{\partial}{\partial m} C\left(m, \sigma, a, b\right) \\
&= \int_a^b \frac{u-m}{\sigma^2} \exp\left(-\frac{1}{2}\frac{(u-m)^2}{\sigma^2}\right) \mathrm{d}u \\
&= \left[-\exp\left(-\frac{1}{2}\frac{(u-m)^2}{\sigma^2}\right)\right]_a^b \\
&= \exp\left(-\frac{1}{2}\frac{(a-m)^2}{\sigma^2}\right) - \exp\left(-\frac{1}{2}\frac{(b-m)^2}{\sigma^2}\right)
\end{aligned}$$

and

$$\begin{aligned}
C_\sigma\left(m, \sigma, a, b\right) &\coloneqq \frac{\partial}{\partial \sigma} C\left(m, \sigma, a, b\right) \\
&= \int_a^b \frac{(u-m)^2}{\sigma^3} \exp\left(-\frac{1}{2}\frac{(u-m)^2}{\sigma^2}\right) \mathrm{d}u \\
&= \left[-\frac{u-m}{\sigma} \exp\left(-\frac{(u-m)^2}{2\sigma^2}\right)\right]_a^b + \frac{1}{\sigma}\int_a^b \exp\left(-\frac{1}{2}\frac{(u-m)^2}{\sigma^2}\right) \mathrm{d}u \\
&= \frac{a-m}{\sigma} \exp\left(-\frac{(a-m)^2}{2\sigma^2}\right) - \frac{b-m}{\sigma} \exp\left(-\frac{(b-m)^2}{2\sigma^2}\right) + \frac{1}{\sigma} C\left(m, \sigma, a, b\right)
\end{aligned}$$

Negative log-likelihood From Equation (1.10), we can define the negative log-likelihood adapted for our specific problem:

$$\mathcal{L}_{k,p}\left(\mu_k, \alpha_{k,p}, m_{k,p}, \sigma_{k,p}\right) \coloneqq -\log L\left(\lambda_k, \mathscr{F}_T^{(p)}\right) = \int_0^T \lambda_{k,p}(s)\mathrm{d}s - \sum_{t\in\mathcal{A}_k} \log \lambda_{k,p}(t) \tag{2.9}$$

Hypothesis (2.5) implies that $\forall i = 1, \ldots, n_p - 1$,

$$\begin{aligned}\int_{t_i^{(p)}}^{t_{i+1}^{(p)}} \kappa_{k,p}(s - t_*^{(p)}(s))\mathbb{1}_{\left\{s\geq t_1^{(p)}\right\}}\,\mathrm{d}s &= \underbrace{\int_{t_i^{(p)}}^{t_i^{(p)}+a} \kappa_{k,p}(s - t_i^{(p)})\,\mathrm{d}s}_{=0} + \underbrace{\int_{t_i^{(p)}+a}^{t_i^{(p)}+b} \kappa_{k,p}(s - t_i^{(p)})\,\mathrm{d}s}_{=1} \\ &\quad + \underbrace{\int_{t_i^{(p)}+b}^{t_{i+1}^{(p)}} \kappa_{k,p}(s - t_i^{(p)})\,\mathrm{d}s}_{=0} \\ &= 1\end{aligned}$$

and hypothesis (2.7) implies that

$$\begin{aligned}\int_{t_{n_p}^{(p)}}^{T} \kappa_{k,p}(s - t_*^{(p)}(s))\mathbb{1}_{\left\{s\geq t_1^{(p)}\right\}}\,\mathrm{d}s &= \underbrace{\int_{t_{n_p}^{(p)}}^{t_{n_p}^{(p)}+a} \kappa_{k,p}(s - t_{n_p}^{(p)})\,\mathrm{d}s}_{=0} + \underbrace{\int_{t_{n_p}^{(p)}+a}^{t_{n_p}^{(p)}+b} \kappa_{k,p}(s - t_{n_p}^{(p)})\,\mathrm{d}s}_{=1} \\ &\quad + \underbrace{\int_{t_{n_p}^{(p)}+b}^{T} \kappa_{k,p}(s - t_{n_p}^{(p)})\,\mathrm{d}s}_{=0} \\ &= 1\end{aligned}$$

Thus,

$$\begin{aligned}\int_0^T \lambda_{k,p}(s)\,\mathrm{d}s &= \int_0^T \mu_k + \alpha_{k,p}\kappa_{k,p}(s - t_*^{(p)}(s))\mathbb{1}_{\left\{s\geq t_1^{(p)}\right\}}\,\mathrm{d}s \\ &= \mu_k T + \alpha_{k,p}\underbrace{\int_0^{t_1^{(p)}} \kappa_{k,p}(s - t_*^{(p)}(s))\mathbb{1}_{\left\{s\geq t_1^{(p)}\right\}}\,\mathrm{d}s}_{=0} + \alpha_{k,p}\sum_{i=1}^{n_p-1}\int_{t_i^{(p)}}^{t_{i+1}^{(p)}} \kappa_{k,p}(s - t_i^{(p)})\,\mathrm{d}s \\ &\quad + \alpha_{k,p}\int_{t_{n_p}^{(p)}}^{T} \kappa_{k,p}(s - t_{n_p}^{(p)})\,\mathrm{d}s \\ &= \mu_k T + \alpha_{k,p} n_p\end{aligned}$$

Finally,

$$\begin{aligned}\mathcal{L}_{k,p}\left(\mu_k, \alpha_{k,p}, m_{k,p}, \sigma_{k,p}\right) &= \mu_k T + \alpha_{k,p} n_p \\ &\quad - \sum_{t \in \mathcal{A}_k} \log\left(\mu_k + \alpha_{k,p}\kappa_{k,p}(t - t_*^{(p)}(t); m_{k,p}, \sigma_{k,p}) \mathbb{1}_{\left\{t \geq t_1^{(p)}\right\}}\right)\end{aligned} \tag{2.10}$$

TODO : calculer la complexité du calcul de la nll, comme fait dans la thèse de Martin

Then the derivatives of the inverse log-likelihood with respect to the parameters are given by

$$\frac{\partial}{\partial \mu_k}\mathcal{L}_{k,p}\left(\mu_k, \alpha_{k,p}, m_{k,p}, \sigma_{k,p}\right) = T - \sum_{t \in \mathcal{A}_k} \frac{1}{\lambda_{k,p}(t)} \tag{2.11}$$

$$\begin{aligned}\frac{\partial}{\partial \alpha_{k,p}}\mathcal{L}_{k,p}\left(\mu_k, \alpha_{k,p}, m_{k,p}, \sigma_{k,p}\right) &= n_p - \sum_{t \in \mathcal{A}_k} \frac{\kappa_{k,p}(t - t_*^{(p)}(t))}{\lambda_{k,p}(t)} \mathbb{1}_{\left\{t \geq t_1^{(p)}\right\}} \\ &= n_p - \sum_{t \in \mathcal{A}_k, t \geq t_1^{(p)}} \frac{\kappa_{k,p}(t - t_*^{(p)}(t))}{\lambda_{k,p}(t)}\end{aligned} \tag{2.12}$$

$$\begin{aligned}\frac{\partial}{\partial m_{k,p}}\mathcal{L}_{k,p}\left(\mu_k, \alpha_{k,p}, m_{k,p}, \sigma_{k,p}\right) &= -\sum_{t \in \mathcal{A}_k} \frac{\alpha_{k,p}}{\lambda_{k,p}(t)} \frac{\partial}{\partial m_{k,p}} \kappa_{k,p}(t - t_*^{(p)}(t); m_{k,p}, \sigma_{k,p}, a, b) \mathbb{1}_{\left\{t \geq t_1^{(p)}\right\}} \\ &= -\sum_{t \in \mathcal{A}_k, t \geq t_1^{(p)}} \frac{\alpha_{k,p}}{\lambda_{k,p}(t)} \frac{\partial}{\partial m_{k,p}} \kappa_{k,p}(t - t_*^{(p)}(t); m_{k,p}, \sigma_{k,p}, a, b)\end{aligned} \tag{2.13}$$

where

$$\frac{\partial}{\partial m}\kappa(x; m, \sigma, a, b) = \left(\frac{x - m}{\sigma^2} - \frac{C_m\left(m, \sigma, a, b\right)}{C\left(m, \sigma, a, b\right)}\right) \kappa(x; m, \sigma, a, b) \tag{2.14}$$

$$\begin{aligned}\frac{\partial}{\partial \sigma_{k,p}}\mathcal{L}_{k,p}\left(\mu_k, \alpha_{k,p}, m_{k,p}, \sigma_{k,p}\right) &= -\sum_{t \in \mathcal{A}_k} \frac{\alpha_{k,p}}{\lambda_{k,p}(t)} \frac{\partial}{\partial \sigma_{k,p}} \kappa_{k,p}(t - t_*^{(p)}(t); m_{k,p}, \sigma_{k,p}, a, b) \mathbb{1}_{\left\{t \geq t_1^{(p)}\right\}} \\ &= -\sum_{t \in \mathcal{A}_k, t \geq t_1^{(p)}} \frac{\alpha_{k,p}}{\lambda_{k,p}(t)} \frac{\partial}{\partial \sigma_{k,p}} \kappa_{k,p}(t - t_*^{(p)}(t); m_{k,p}, \sigma_{k,p}, a, b)\end{aligned} \tag{2.15}$$

where

$$\frac{\partial}{\partial \sigma}\kappa(x; m, \sigma, a, b) = \left(\frac{(x - m)^2}{\sigma^3} - \frac{C_\sigma\left(m, \sigma, a, b\right)}{C\left(m, \sigma, a, b\right)}\right) \kappa(x; m, \sigma, a, b) \tag{2.16}$$

To lighten notations, we define $\mathcal{A}_{k,p} = \left\{t,\ t \in \mathcal{A}_k, t \geq t_1^{(p)}\right\}$ as the set of atom k's activation timestamps that occur after the first timestamps of task p. In other words, $\mathcal{A}_{k,p}$ is the set of all atom k's activations that might have been driven by a task of type p.

Expectation step Let $P_{t,k}$ be the probability that the activation at time t has been triggered by the baseline intensity of atom k, and $P_{t,p}$ be that probability that the activation at time t has been triggered by the driver p (the task p in our case).

$$P_{t,k} = \frac{\mu_k}{\lambda_{k,p}(t)} \tag{2.17}$$

$$P_{t,p} = \frac{\alpha_{k,p}\kappa_{k,p}\left(t - t_*^{(p)}(t)\right)}{\lambda_{k,p}(t)} \mathbb{1}_{\left\{t \geq t_1^{(p)}\right\}} \tag{2.18}$$

Maximisation step Details of the computation are present in Appendix B.

$$\mu_k^{(n+1)} = \frac{1}{T} \sum_{t \in \mathcal{A}_k} P_{t,k}^{(n)} \tag{2.19}$$

$$\alpha_{k,p}^{(n+1)} = \frac{1}{n_p} \sum_{t \in \mathcal{A}_{k,p}} P_{t,p}^{(n)} \tag{2.20}$$

$$m_{k,p}^{(n+1)} = \frac{\sum_{t \in \mathcal{A}_{k,p}} \left(t - t_*^{(p)}(t)\right) P_{t,p}^{(n)}}{\sum_{t \in \mathcal{A}_{k,p}} P_{t,p}^{(n)}} - {\sigma_{k,p}^{(n)}}^2 \frac{C_m\left(m_{k,p}^{(n)}, \sigma_{k,p}^{(n)}, a, b\right)}{C\left(m_{k,p}^{(n)}, \sigma_{k,p}^{(n)}, a, b\right)} \tag{2.21}$$

$$\sigma_{k,p}^{(n+1)} = \left(\frac{C\left(m_{k,p}^{(n)}, \sigma_{k,p}^{(n)}, a, b\right)}{C_\sigma\left(m_{k,p}^{(n)}, \sigma_{k,p}^{(n)}, a, b\right)} \frac{\sum_{t \in \mathcal{A}_{k,p}} \left(t - t_*^{(p)}(t) - m_{k,p}\right)^2 P_{t,p}^{(n)}}{\sum_{t \in \mathcal{A}_{k,p}} P_{t,p}^{(n)}} \right)^{1/3} \tag{2.22}$$

Remark If $\alpha_{k,p} = 0$, then the intensity is reduced to its baseline, $\lambda_{k,p}(t) = \mu_k$, and then the negative log-likelihood is:

$$\mathcal{L}_{k,p}\left(\mu_k, \alpha_{k,p}, m_{k,p}, \sigma_{k,p}\right) = \mu_k T - \sum_{t \in \mathcal{A}_k} \log \mu_k = \mu_k T - \#\mathcal{A}_k \log \mu_k \tag{2.23}$$

Thus, we can directly compute the maximum likelihood estimator (MLE) for μ_k, as follow:

$$\mu_k^{(MLE)} = \frac{\#\mathcal{A}_k}{T} \tag{2.24}$$

2.5.2 Parameters initialisation

In this section, we will describe two methods on how to initialise the parameters for the EM algorithm, in other words, how to determine $\theta^{(0)} = \left(\mu^{(0)}, \alpha^{(0)}, m^{(0)}, \sigma^{(0)}\right)$.

Random initialisation The first initialisation method is simply done by random sampling the parameters according to a specific uniform distribution:

- $\mu^{(0)} \sim \mathcal{U}nif_{[\mu_{min}\,;\mu_{max}]}$, e.g. $\mu_{min} = 0$ and $\mu_{max} = 5$;
- $\alpha^{(0)} \sim \mathcal{U}nif_{[\alpha min\,;\alpha max]}$, e.g. $\alpha min = 0.1$ and $\alpha max = 5$;
- $m^{(0)} \sim \mathcal{U}nif_{[a\,;b]}$, were a and b are the truncation values of the kernel, Eq. (2.3);
- $\sigma^{(0)} \sim \mathcal{U}nif_{[\sigma_{min}\,;\sigma_{max}]}$, e.g. $\sigma_{min} = 0.01$ and $\sigma_{max} = 1$.

Smart initialisation An other method, more intuitive, would be to compute the empirical values of those parameters. We call this method "smart initialisation". More specifically, the initial values of the parameters are computed as follow:

- Let $\mathcal{D}_{k,p} \coloneqq \left\{t - t_*^{(p)}(t),\ t \in \mathcal{A}_{k,p}\right\} \cap [a\,;b]$ be the set of all empirical delays possibly linked to the task p. Then, we define $m^{(0)}$ as the mean of the empirical delays:

$$m^{(0)} = \frac{1}{\#\mathcal{D}_{k,p}} \sum_{d \in \mathcal{D}_{k,p}} d$$

  where $\#A$ denotes the cardinality of the set A.

- Similarly, we define $\sigma^{(0)}$ as the standard deviation of the empirical delays:

$$\sigma^{(0)} = \sqrt{\frac{1}{\#\mathcal{D}_{k,p}} \sum_{d \in \mathcal{D}_{k,p}} \left|d - m^{(0)}\right|^2}$$

- The activations that do not lend on any kernel support come from the baseline intensity, which is fixed, so it is similar to say that those activations are the result of a homogeneous Poisson process. Thus, to initialise the baseline value $\mu^{(0)}$, we simply compute the Poisson process parameter's estimator, which is the average number of activation over the process' duration:

$$\begin{aligned} \mu^{(0)} &= \frac{\#\left\{\mathcal{A}_k \setminus \left\{t,\ t \in \mathcal{A}_{k,p}, a \leq t - t_*^{(p)}(t) \leq b\right\}\right\}}{T - n_p\,(b-a)} \\ &= \frac{\#\mathcal{A}_k - \#\left\{t,\ t \in \mathcal{A}_{k,p}, a \leq t - t_*^{(p)}(t) \leq b\right\}}{T - n_p\,(b-a)} \end{aligned} \tag{2.25}$$

because there can be no duplicates, and where $n_p\,(b-a)$ is the sum of all kernels support.

- Finally, we set $\alpha^{(0)}$ as follow:

$$\alpha^{(0)} = f(p_a) \tag{2.26}$$

with

$$f(p_a) = -e^{\mu} \ln\left(\frac{l(\mu) - p_a}{l(\mu) - p_s}\right) \tag{2.27}$$

and

$$l(\mu) = \left(\frac{e^{\mu} - 1}{5} + \frac{1}{1 - p_s}\right)^{-1} + p_s \tag{2.28}$$

where

$$p_a \coloneqq \frac{\#\left\{t,\ t \in \mathcal{A}_{k,p}, a \leq t - t_*^{(p)}(t) \leq b\right\}}{\#\mathcal{A}_k} \tag{2.29}$$

is the proportion of activations that lend in a kernel support, and

$$p_s \coloneqq \frac{n_p\,(b-a)}{T} \tag{2.30}$$

is the ratio of all the kernels' support time over the total duration.

More details on the role of the α coefficient and how its initialisation function was determined are presented in Appendix A.

2.6 Model extension to multiple point processes drivers

TODO: dire que l'on peut facilement étendre le modèle de façon à y include différents drivers pour un même atome. Mettre alors l'équation de la nouvelle fonction d'intensité, préciser que ça fait partie du futur work. Dire aussi que l'on peut éventuellement continuer en y incluant les activations des autres atomes, de façon à déterminer les liens temporels entre eux. Mettre alors de nouveau l'équation de l'intensité, où pour chaque atome, il y a un groupe de drivers (les différents tasks) et un groupe de processus "classiques" comme dans Hawkes

3 Results

In this section, we will present several results of interest in order to show the performances of the EM based algorithm, on both simulated and real data. For the sake of brevity, some other figures are presented in Appendix C.

3.1 Simulated data

The results presented in this section will be based on simulated data. The main advantage to work with simulated data in our situation, is the possibility to have a ground truth to compare our results to. However, we do not have a baseline method to compare the performances of our algorithm with. For this reason, the results presented will focus on the capacity of our algorithm to converge toward a solution.

To simulate the data, two steps are necessary: first, the simulation of the task timestamps $t^{(p)} = \left\{t_1^{(p)}, \dots, t_{n_p}^{(p)}\right\}$, that allows to determine an intensity function $\lambda_{k,p}$, and second, thanks to this function, the simulation of the atom's activation's timestamps $\mathcal{A}_k$ using the simulation algorithm of one-dimensional nonhomogeneous Poisson process previously introduced (Algo. 1). To simulate the task timestamps, we simply generate regular timestamps during the entire duration of the experimentation T with a given interval (inter stimulus interval) that will be denoted by the variable `isi`. Then, we randomly select a percentage of those timestamps to keep, e.g., 80 %. This percentage is denoted in the following by the variable `n_tasks`.

In the following, we call hyper-parameters the variables T, `isi`, `n_tasks` and the truncation values a and b of the kernel, as they are not parameters that are learnt by the model. The parameters μ, α, m and σ, that are learnt by the model, will simply be called parameters or variables, interchangeably.

For the first experimentation, we are interested in determining to what extent the EM algorithm makes it possible to retrieve the true values of the parameters, and at what speed. In other words, we are looking to find out whether or not the algorithms converges, both in the loss function and parameter values. We fix the hyper-parameters at the following values, in order to generate data similar to an M/EEG recording: $T = 4\,\text{min}$, `isi` $= 1.2$, `n_tasks` $= 80\,\%$, $[a\,;b] = [30\,;800]\times10^{-3}\,\text{s}$. The loss history and the parameters recovery are respectively presented in Figures 3.1 and 3.2. Other results of similar experiences are presented in Appendix C.1.

Another experimentation of interest is to determine what is the influence of the hyper-parameters on the algorithm's capacity to converge toward the true value of the parameters. In order to continue to generate data similar to a real M/EEG recording, in the following, unless otherwise stated, the inter-stimulus interval and the truncation values will be kept to reasonable values, i.e., `isi` $= 1.2$ and $[a\,;b] = [30\,;800] \times 10^{-3}\,\text{s}$. Thus, the two main hyper-parameters we are interested in are the duration of the experimentation T and the number of tasks timestamps kept `n_tasks`. This will allow us to see whether or not the algorithm needs a lot data to be efficient. This is a crucial point to determine as in reality, M/EEG data are costly to acquire,

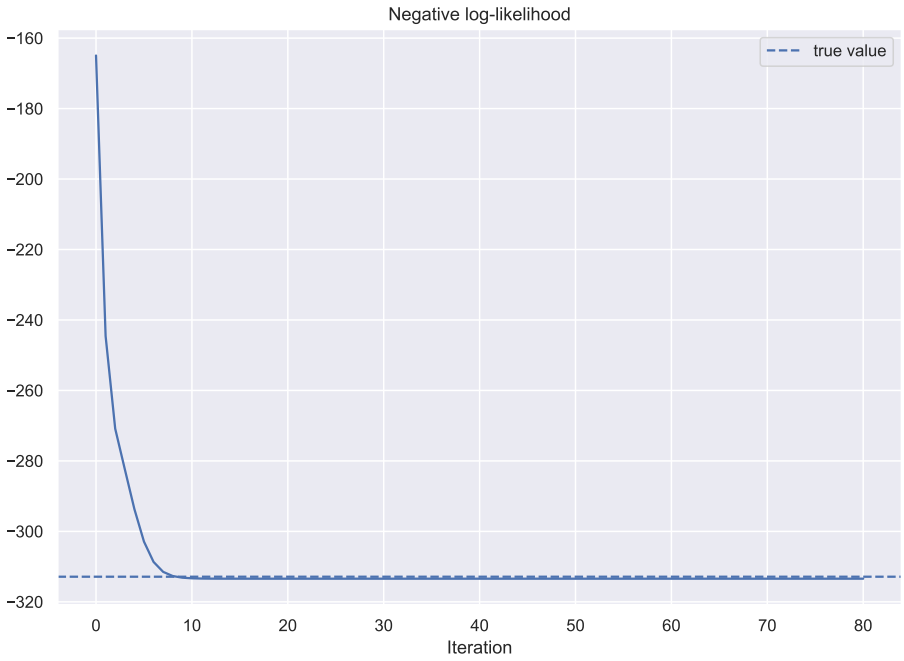


Figure 3.1 – Loss history over 80 iterations with a *smart* initialisation, with $T = 4\,\text{min}$, `isi` $= 1.2$, `n_tasks` $= 80\,\%$, $[a\,;b] = [30\,;800] \times 10^{-3}\,\text{s}$.

generally are of a relatively short duration, and the stimuli exerted on the subject count only a few repetitions per type, a hundred in the best of cases.

For those experimentations, we vary `n_tasks` and T, keeping fixed all other parameters. For every value of `n_tasks` and T, we run 50 EM algorithms with 80 iterations, and plot the boxplots of the estimated parameters μ, α, m and σ, with their true value, alongside the boxplots for the squared error of the loss and for the computation time. Results are shown in Fig. 3.3 and 3.4.

3.2 Real data

In this section, the results that will be presented are obtained on real data. The main dataset that we will be using is called *sample*, as it simply is the sample data available on the `MNE` package. In this experiment, checkerboard patterns were presented to the subject into the left and right visual field, interspersed by tones to the left or right ear. The interval between the stimuli was 750 ms. Occasionally a smiley face was presented at the centre of the visual field. The subject was asked to press a key with the right index finger as soon as possible after the appearance of the face [15]. In the following, we are only interested in the four main stimuli types: auditory left, auditory right, visual left and visual right. The experiment lasts about 4.6 min and

15. Source: MNE documentation

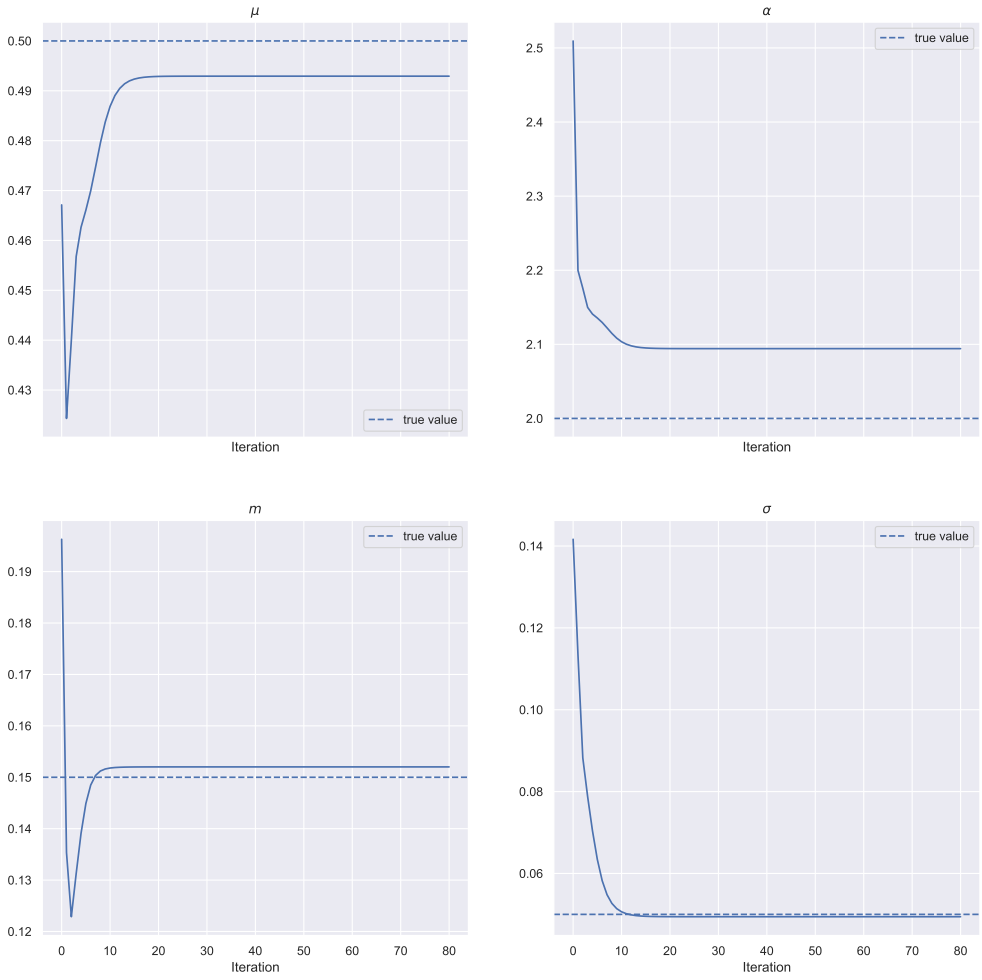


Figure 3.2 – Parameters recovery over 80 iterations with a *smart* initialisation, with $T = 4\,\text{min}$, `isi` $= 1.2$, `n_tasks` $= 80\,\%$, $[a\,;b] = [30\,;800] \times 10^{-3}\,\text{s}$.

approximately 70 stimuli per type was presented, with a minimum of 2.5 s between two stimuli of the same type.

As previously explained, the recorded signals must first of all be decomposed in K atoms. In our case, we put $K = 40$, and some examples of obtained atoms were presented in Fig. 1.4.

Unlike the previous section where the data were simulated and thus the real values of the parameters were known, now we do not have any ground truth to compare our results to. Thus, we will focus on showing that the algorithm converges and real data too. However, with some domain expertise, one is able to determine the origin of a given atom. For example, the atom 0 in Fig. 1.4, the one on the left, is characteristic of an heartbeat, which is confirmed by its spatial

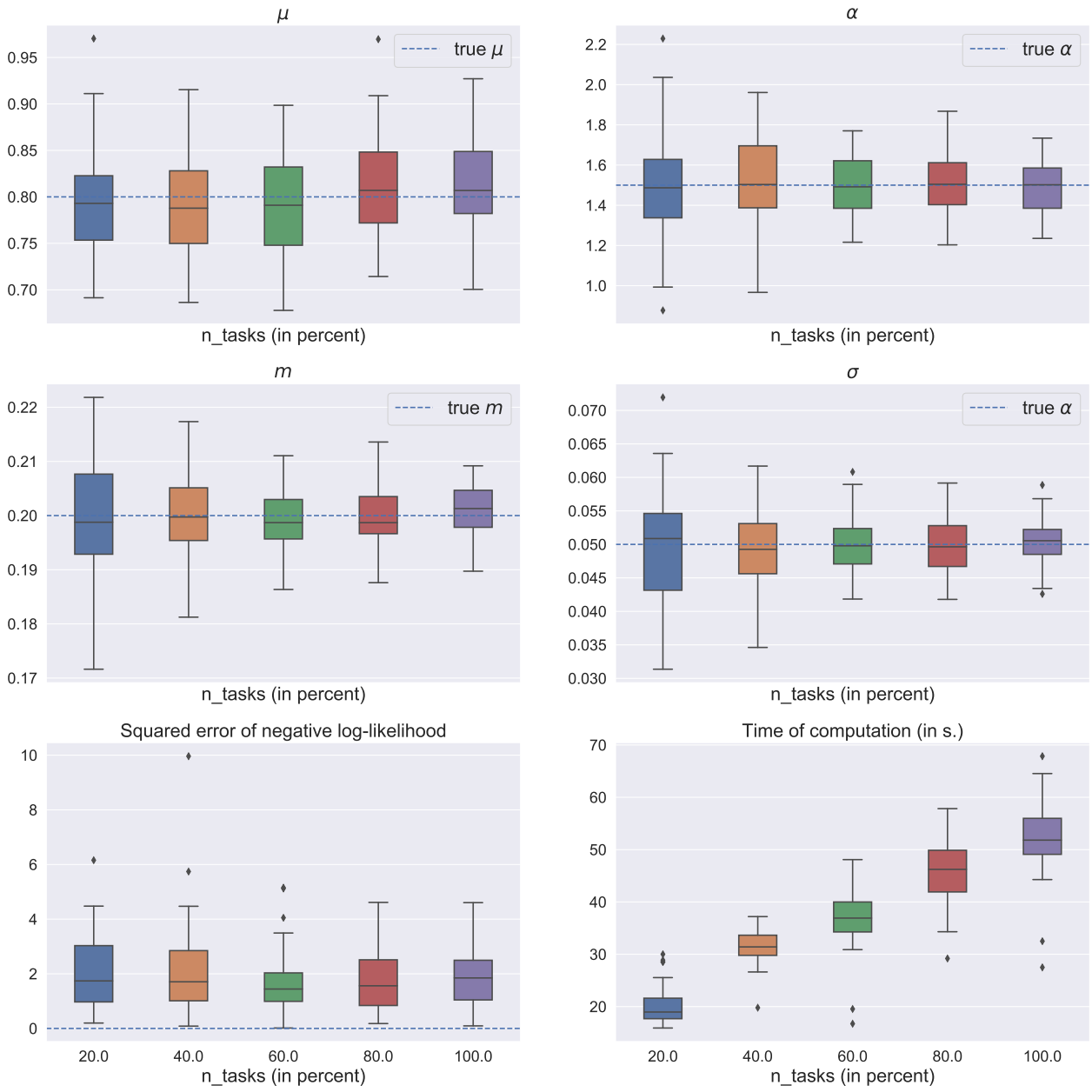


Figure 3.3 – Influence of `n_tasks` on parameters recovery, loss error and computation time, 50 simulations for each value, with 80 EM iterations with a *smart* initialisation, with $T = 4.6\,\text{min}$, `isi` $= 2.5$,, $[a\,;b] = [30\,;800] \times 10^{-3}\,\text{s}$.

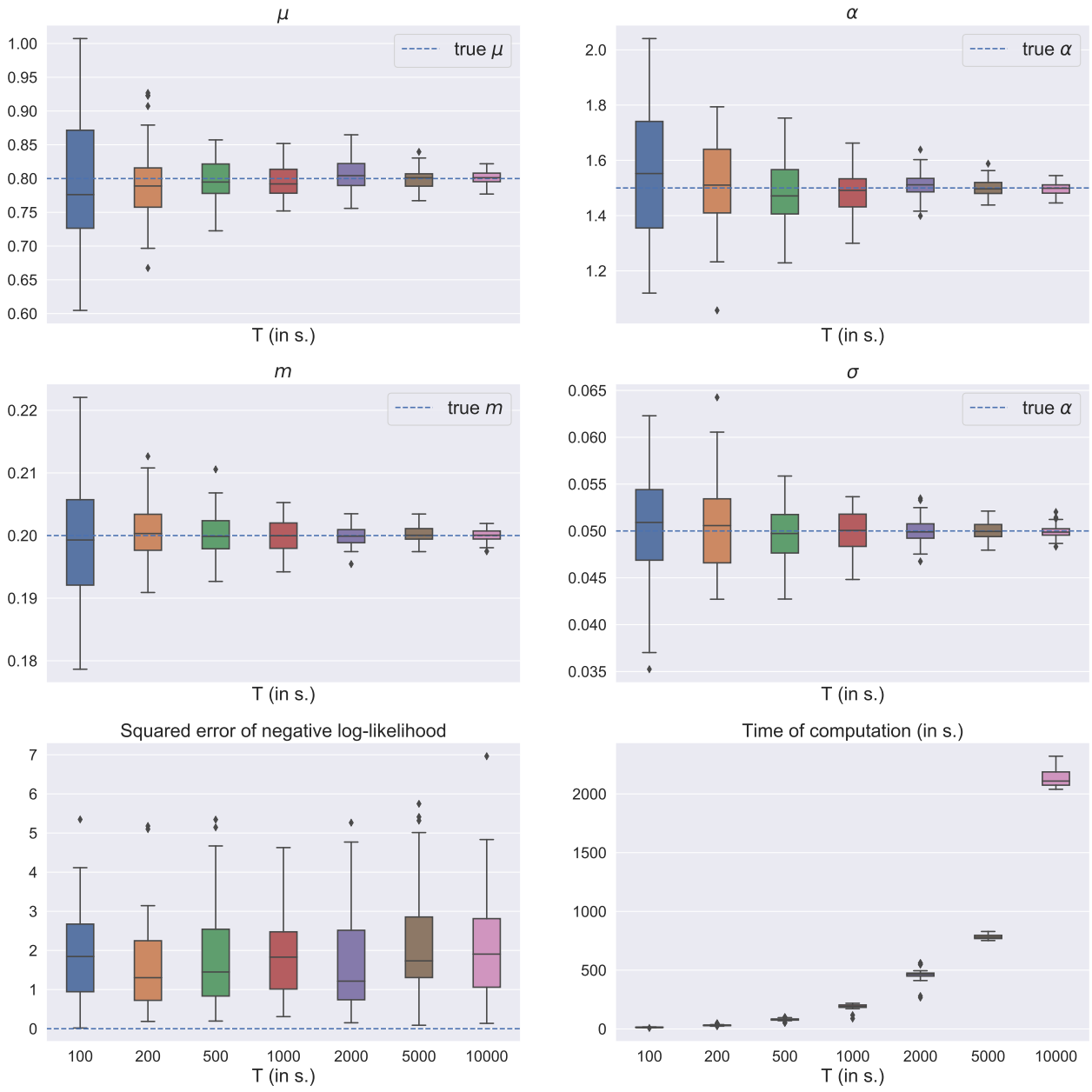


Figure 3.4 – Influence of T on parameters recovery, loss error and computation time, 50 simulations for each value, with 80 EM iterations with a *smart* initialisation, with `n_tasks` $= 80\,\%$, `isi` $= 2.5$,, $[a\,;b] = [30\,;800] \times 10^{-3}$ s.

pattern which represents a diffuse area rather than a specific zone, as the heartbeat comes from "under" the brain, approximately form the middle left. Similarly, the atom 1, the one in the middle, can easily be associated with the subject's eye blinks. Thus, even if we do not have a ground truth, as we do not know which atom is linked to which stimulus, it can be said that some specific atoms are not linked to any stimulus, as it is the case with the heartbeat atom, for which it is reasonable to say that it is not influenced by any stimulus. Note that such a statement is not as clear for the eye blinks atom, as the subject can eventually anticipate the incoming visual stimuli and therefore make an effort to not blink accordingly.

TODO: Présentation rapide des deux/trois datasets (sample, camcan, somato), avec quelques stats des. éventuellement. Performances de l'algorithme sur des données réelles : lesquelles (sample, camcan et somato, présenter ce que sont ces différents jeux de données), comment évaluer la pertinence des résultats (absence ground truth et pas possible d'avoir une baseline avec Tick) ? Dire que si on connaît déjà le lien possible, on peut interpréter les résultats, parler de la mesure ration p_a/p_s exemple sur sample de l'atoms lié aux battements de coeurs, qui par définition, ne doit pas avoir de lien avec aucun des task

4 Discussion

TODO: discuss results. Futur work: say that we know need a statistical test to determine whether or not an atom is linked to a stimulus, based on the model fitted; in alphacsc add a feature to give in input the timestamps of interest, i.e., where we are interested in finding correlated atoms;

5 Conclusion

Appendices

A The role of the α coefficient in the intensity function

Here we seek to determine an intuitive approach to the role of the coefficient α in intensity, Eq. (2.1). The goal of this approach is to determine a function that will allow to initialize the α parameter for the EM algorithm, for the *smart init*, as it has been done for the μ coefficient of the baseline. The starting point is the Algorithm 1, which allows us to simulate a non-homogenous Poisson process using a given intensity function. The coefficient α intervenes at two places in this algorithm: when determining $\bar{\lambda} = \max_{0 \leq t} \lambda(t)$, and upon acceptance, or not, of the points s_{m+1}.

As previously mentioned in Eq. (2.8), $\bar{\lambda} = \mu_k + \alpha_{k,p}\kappa_{k,p}(m_{k,p})$, thus, the greater the α coefficient, the higher the maximum intensity over the $[0\,;T]$ interval. For the selected points s_{m+1}, two cases are to be considered: when the points are on the baseline and when the points are on the supports of the different kernels. For more clarity, let us note the set of the supports of the various kernels $\mathcal{S}_p$.:

$$\mathcal{S}_p \coloneqq \bigcup_{i=1,\dots,n_p} \left[t_i^{(p)} + a\,; t_i^{(p)} + b\right] \tag{A.1}$$

In case a selected point s_{m+1} is on the baseline, i.e., $s_{m+1} \in [0\,;T] \setminus \mathcal{S}_p$, the probability to accept this point is equal to $\lambda(s_{m+1})/\bar{\lambda} = \mu_k/\bar{\lambda}$, and thus decreases when α increases. Therefore, as α increases, the proportion of activations on the baseline decreases, and conversely, the proportion of activations on the kernel supports, i.e., in $\mathcal{S}_p$ is greater. This relationship between the value of α and the proportion of activations in $\mathcal{S}_p$ is represented in Figure A.1, where for each value of the coefficient α, from 0 to 5 with a step of 0.1, the said proportion is calculated on the ordinate as follows:

$$p_a \coloneqq \frac{\#\left\{t \in \mathcal{A}_{k,p},\ t \in \mathcal{S}_p\right\}}{\#\mathcal{A}_k} \tag{A.2}$$

As mentioned before, we are looking for a function, which, from the data, would allow us to have a first estimate of α, in order to initialise the EM algorithm intuitively. A first observation that can be made with the help of this figure is that there is most probably a relationship between the value of the coefficient α and p_a. If we manage to establish such a relation, then, by taking its inverse, we will be able to determine a first value of α from the proportion p_a, which can be calculated immediately from the data. We can then initiate α in the following way:

$$\alpha^{(0)} = f(p_a) \tag{A.3}$$

A first observation that can be made from the outset is that, whatever the values of the parameters, α, μ, T, etc., p_a will always be below 1, by definition. So, looking at the shape of

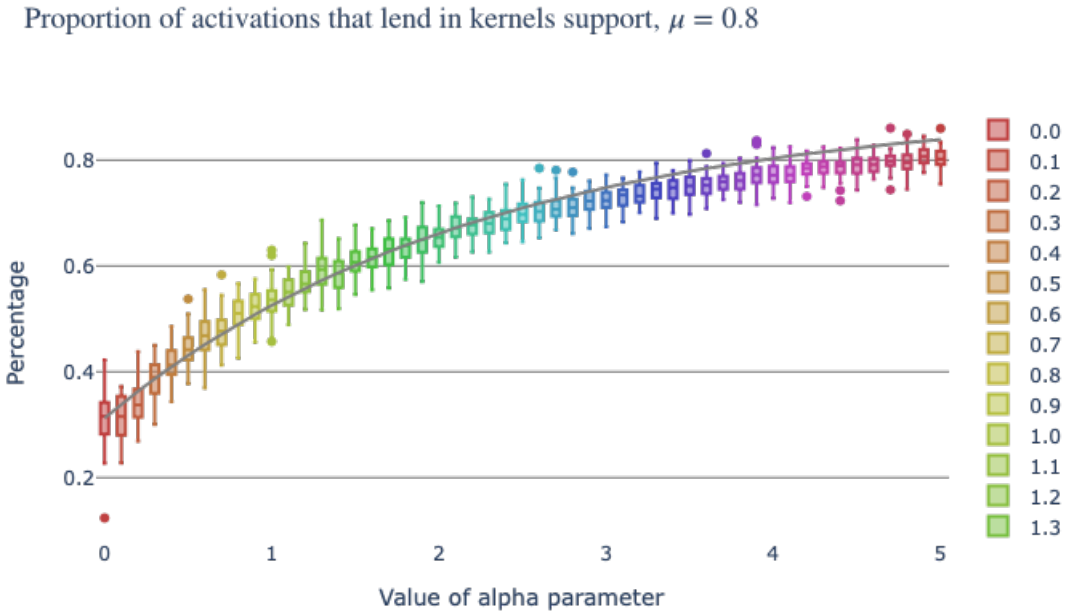


Figure A.1 – Proportion of generated activations within $\mathcal{S}$ based on the α parameter, where $\mu = 0.8, T = 180, n_p = 73, [a\,;b] = [30, 800]\,10^{-3}$, with 50 simulations for each value of α. In black, the function $f^{-1}(p_a)$.

the relationship in Figure A.1, a first idea is the following:

$$f(p_a) = -\ln(1 - p_a) \tag{A.4}$$

When $\alpha = 0$, the intensity is reduced to its baseline, i.e., $\lambda_{k,p}(t) = \mu_k, \forall t \in [0\,;T]$, the proportion of acceptance in the simulation algorithm is therefore 1. Thus, we find ourselves in the simple case of a simulation of a homogeneous Poisson process, i.e., with constant intensity. Thus, the proportion of activations found in the supports of the different kernels must be simply equal to the proportion represented by all the supports in the total duration. We note this proportion p_s:

$$p_s \coloneqq \frac{n_p\,(b-a)}{T} \tag{A.5}$$

Replacing with the numerical values used for the Figure A.1 gives $p_s = 0.31$, which is consistent with the median obtained for $\alpha = 0$.

Thus, a second observation that can be made is that if α is always positive, then p_a is always higher than p_s. We then modify A.4 accordingly:

$$f(p_a) = -\ln(1 - p_a) + \ln(1 - p_s) = -\ln\left(\frac{1 - p_a}{1 - p_s}\right) \tag{A.6}$$

In addition, the stronger the baseline, the less p_a increases, so we want the curvature to decrease when μ increases:

$$f(p_a) = -e^{\mu} \ln\left(\frac{1-p_a}{1-p_s}\right) \tag{A.7}$$

Likewise, the higher μ is, the harder it is for p_a to reach the limit of 1. However, we do not consider values of α too high, in order to remain realistic, and so we especially want a good estimate of α for values lower than 5. Therefore, we propose to adapt the limit according to μ:

$$f(p_a) = -e^{\mu} \ln\left(\frac{l(\mu)-p_a}{l(\mu)-p_s}\right) \tag{A.8}$$

where $l(\mu)$ is a function which is worth 1 when $\mu = 0$ (in this case all the activations are in $\mathcal{S}$, $p_a = 1$), and which tends towards p_s when μ increases (because we recall that p_a cannot be systematically lower than p_s for positive values of α).

Thus, we propose:

$$l(\mu) = \left(\frac{e^{\mu}-1}{5} + \frac{1}{1-p_s}\right)^{-1} + p_s \tag{A.9}$$

In order to test our initialisation function, and to show that it is better than chance, we calculate $\alpha^{(0)} = f(p_a)$, for different values of μ and α, for 50 simulations each time. The RMSE (*Root Mean Square Error*) is then calculated. The result is shown in Figure A.2.

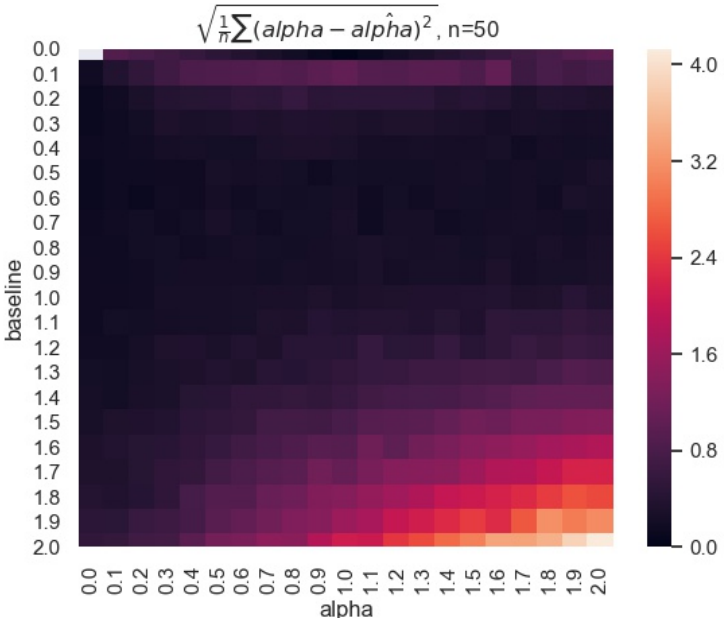


Figure A.2 – RMSE between the calculated value for $\alpha^{(0)}$ and its true value, for different values of μ and α.

It can be seen from this figure that the estimate made of α is more than correct (i.e., cases where the RMSE is smaller than 1) in two main situations [16]:

16. We do not take into account the degenerate case where $\mu = \alpha = 0$.

- when the μ baseline is low, but greater than 0.2, for all values of α;
- when the α coefficient is low for all values in the baseline.

Conversely, the more μ and α increase simultaneously, the less accurate the initial estimate of α is.

There is a second degenerate case, in addition to the case where $\mu = \alpha = 0$: when $\mu = 0$ and $\alpha > 0$. In this case, since the baseline is null, all activations are present on the kernel supports, because in the Algorithm 1, if $s_{m+1} \notin \mathcal{S}$, then the intensity is null and therefore the probability of accepting such a point is also null. Thus, we have $p_a = 1$, and $l(\mu) = 1$, and then $f(p_a)$ is worth $+\infty$. So, when one is in this situation, one decides to return 1:

$$f(p_a; \mu, p_s) = \begin{cases} f(p_a) & \text{si } \mu > 0 \text{ ou } p_a > 0 \\ 1 & \text{sinon} \end{cases} \tag{A.10}$$

Remarque In practice, the μ parameter of the function $f(p_a; \mu, p_s)$ is not the true value of the baseline, but the calculated value for its initialisation, as defined in Eq. (2.25).

B Details of EM-based algorithm computations

Recall that we denote with the upper script the iteration step, e.g., $\theta^{(n)}$ is the value of the parameter θ at step n of the EM algorithm. By extension, if f is a function of $\theta_1, \dots, \theta_m$, then we define $f^{(n)}(\theta_1, \dots, \theta_m) \coloneqq f\left(\theta_1^{(n)}, \dots, \theta_m^{(n)}\right)$.

Recall also that we defined $P_{t,k}^{(n)}$ and $P_{t,p}^{(n)}$ as follow:

$$P_{t,k} = \frac{\mu_k}{\lambda_{k,p}(t)}$$

$$P_{t,p} = \frac{\alpha_{k,p}\kappa_{k,p}\left(t - t_*^{(p)}(t)\right)}{\lambda_{k,p}(t)} \mathbb{1}_{\left\{t \geq t_1^{(p)}\right\}}$$

For μ_k

$$\begin{aligned}
&\frac{\partial}{\partial \mu_k} \mathcal{L}_{k,p}\left(\mu_k, \alpha_{k,p}, m_{k,p}, \sigma_{k,p}\right) = 0 \\
\Leftrightarrow &\sum_{t \in \mathcal{A}_k} \frac{1}{\lambda_{k,p}(t)} = T \\
\Leftrightarrow &\sum_{t \in \mathcal{A}_k} \frac{P_{t,k}^{(n)}}{\mu_k} = T \\
\Leftrightarrow &\mu_k^{(n+1)} = \frac{1}{T} \sum_{t \in \mathcal{A}_k} P_{t,k}^{(n)}
\end{aligned} \tag{B.1}$$

For $\alpha_{k,p}$

$$\begin{aligned}
&\frac{\partial}{\partial \alpha_{k,p}} \mathcal{L}_{k,p}\left(\mu_k, \alpha_{k,p}, m_{k,p}, \sigma_{k,p}\right) = 0 \\
\Leftrightarrow &\sum_{t \in \mathcal{A}_{k,p}} \frac{\kappa_{k,p}(t - t_*^{(p)}(t))}{\lambda_{k,p}(t)} = n_p \\
\Leftrightarrow &\sum_{t \in \mathcal{A}_{k,p}} \frac{P_{t,p}^{(n)}}{\alpha_{k,p}} = n_p \\
\Leftrightarrow &\alpha_{k,p}^{(n+1)} = \frac{1}{n_p} \sum_{t \in \mathcal{A}_{k,p}} P_{t,p}^{(n)}
\end{aligned} \tag{B.2}$$

For $m_{k,p}$

$$\begin{aligned}\frac{\partial}{\partial m_{k,p}}\mathcal{L}_{k,p}\left(\mu_k,\alpha_{k,p},m_{k,p},\sigma_{k,p}\right) &= -\sum_{t\in\mathcal{A}_{k,p}}\left(\frac{t-t_*^{(p)}(t)-m_{k,p}}{\sigma_{k,p}^2}-\frac{C_m\left(m_{k,p},\sigma_{k,p},a,b\right)}{C\left(m_{k,p},\sigma_{k,p},a,b\right)}\right)\underbrace{\frac{\alpha_{k,p}\kappa_{k,p}(t)}{\lambda_{k,p}(t)}}_{=P_{t,p}}\\ &= -\frac{1}{\sigma_{k,p}^2}\sum_{t\in\mathcal{A}_{k,p}}\left(t-t_*^{(p)}(t)\right)P_{t,p}\\ &+\left(\frac{m_{k,p}}{\sigma_{k,p}^2}+\frac{C_m\left(m_{k,p},\sigma_{k,p},a,b\right)}{C\left(m_{k,p},\sigma_{k,p},a,b\right)}\right)\sum_{t\in\mathcal{A}_{k,p}}P_{t,p}\end{aligned} \tag{B.3}$$

hence,

$$\begin{aligned}&\frac{\partial}{\partial m_{k,p}}\mathcal{L}_{k,p}\left(\mu_k,\alpha_{k,p},m_{k,p},\sigma_{k,p}\right)=0\\ \Leftrightarrow m_{k,p}^{(n+1)} &= \frac{\sum_{t\in\mathcal{A}_{k,p}}\left(t-t_*^{(p)}(t)\right)P_{t,p}^{(n)}}{\sum_{t\in\mathcal{A}_{k,p}}P_{t,p}^{(n)}}-{\sigma_{k,p}^{(n)}}^2\frac{C_m\left(m_{k,p}^{(n)},\sigma_{k,p}^{(n)},a,b\right)}{C\left(m_{k,p}^{(n)},\sigma_{k,p}^{(n)},a,b\right)}\end{aligned} \tag{B.4}$$

For $\sigma_{k,p}$

$$\begin{aligned}\frac{\partial}{\partial \sigma_{k,p}}\mathcal{L}_{k,p}\left(\mu_k,\alpha_{k,p},m_{k,p},\sigma_{k,p}\right) &= -\sum_{t\in\mathcal{A}_{k,p}}\left(\frac{\left(t-t_*^{(p)}(t)-m_{k,p}\right)^2}{\sigma_{k,p}^3}-\frac{C_\sigma\left(m_{k,p},\sigma_{k,p},a,b\right)}{C\left(m_{k,p},\sigma_{k,p},a,b\right)}\right)\underbrace{\frac{\alpha_{k,p}\kappa_{k,p}(t)}{\lambda_{k,p}(t)}}_{=P_{t,p}}\\ &= -\frac{1}{\sigma_{k,p}^3}\sum_{t\in\mathcal{A}_{k,p}}\left(t-t_*^{(p)}(t)-m_{k,p}\right)^2P_{t,p}\\ &+\frac{C_\sigma\left(m_{k,p},\sigma_{k,p},a,b\right)}{C\left(m_{k,p},\sigma_{k,p},a,b\right)}\sum_{t\in\mathcal{A}_{k,p}}P_{t,p}\end{aligned} \tag{B.5}$$

hence,

$$\begin{aligned}&\frac{\partial}{\partial \sigma_{k,p}}\mathcal{L}_{k,p}\left(\mu_k,\alpha_{k,p},m_{k,p},\sigma_{k,p}\right)=0\\ \Leftrightarrow \sigma_{k,p}^{(n+1)} &= \left(\frac{C\left(m_{k,p}^{(n)},\sigma_{k,p}^{(n)},a,b\right)}{C_\sigma\left(m_{k,p}^{(n)},\sigma_{k,p}^{(n)},a,b\right)}\frac{\sum_{t\in\mathcal{A}_{k,p}}\left(t-t_*^{(p)}(t)-m_{k,p}\right)^2P_{t,p}^{(n)}}{\sum_{t\in\mathcal{A}_{k,p}}P_{t,p}^{(n)}}\right)^{1/3}\end{aligned} \tag{B.6}$$

Link to gradient descent As done in [LM11], it is possible to see the coefficient update in the EM bases algorithm as a gradient descent, in which the coefficients are updated as follow:

$$x^{(n+1)}=x^{(n)}-\eta^{(n)}\nabla f(x^{(n)}) \tag{B.7}$$

where f is the loss function to minimise, η the step size and where ∇ denotes the gradient. In our case, it can easily be done for the parameters μ, α and m.

$$\mu_k^{(n+1)} - \mu_k^{(n)} = -\frac{\mu_k^{(n)}}{T} \frac{\partial}{\partial \mu_k} \mathcal{L}_{k,p} \left(\mu_k, \alpha_{k,p}, m_{k,p}, \sigma_{k,p} \right) \tag{B.8}$$

$$\alpha_{k,p}^{(n+1)} - \alpha_{k,p}^{(n)} = -\frac{\alpha_{k,p}^{(n)}}{n_p} \frac{\partial}{\partial \alpha_{k,p}} \mathcal{L}_{k,p} \left(\mu_k, \alpha_{k,p}, m_{k,p}, \sigma_{k,p} \right) \tag{B.9}$$

$$\begin{aligned}
m_{k,p}^{(n+1)} - m_{k,p}^{(n)} &= \frac{\sum_{t \in \mathcal{A}_{k,p}} \left(t - t_*^{(p)}(t) \right) P_{t,p}^{(n)}}{\sum_{t \in \mathcal{A}_{k,p}} P_{t,p}^{(n)}} - {\sigma_{k,p}^{(n)}}^2 \frac{C_m \left(m_{k,p}^{(n)}, \sigma_{k,p}^{(n)}, a, b \right)}{C \left(m_{k,p}^{(n)}, \sigma_{k,p}^{(n)}, a, b \right)} - m_{k,p}^{(n)} \\
&= \frac{1}{\sum_{t \in \mathcal{A}_{k,p}} P_{t,p}^{(n)}} \sum_{t \in \mathcal{A}_{k,p}} \left(t - t_*^{(p)}(t) - {\sigma_{k,p}^{(n)}}^2 \frac{C_m \left(m_{k,p}^{(n)}, \sigma_{k,p}^{(n)}, a, b \right)}{C \left(m_{k,p}^{(n)}, \sigma_{k,p}^{(n)}, a, b \right)} - m_{k,p}^{(n)} \right) P_{t,p}^{(n)} \\
&= \frac{{\sigma_{k,p}^{(n)}}^2}{\sum_{t \in \mathcal{A}_{k,p}} P_{t,p}^{(n)}} \sum_{t \in \mathcal{A}_{k,p}} P_{t,p}^{(n)} \left(\frac{t - t_*^{(p)}(t) - m_{k,p}^{(n)}}{{\sigma_{k,p}^{(n)}}^2} - \frac{C_m \left(m_{k,p}^{(n)}, \sigma_{k,p}^{(n)}, a, b \right)}{C \left(m_{k,p}^{(n)}, \sigma_{k,p}^{(n)}, a, b \right)} \right) \\
&= -\frac{{\sigma_{k,p}^{(n)}}^2}{\sum_{t \in \mathcal{A}_{k,p}} P_{t,p}^{(n)}} \frac{\partial}{\partial m_{k,p}} \mathcal{L}_{k,p} \left(\mu_k, \alpha_{k,p}, m_{k,p}, \sigma_{k,p} \right)
\end{aligned} \tag{B.10}$$

C Extra results

In this appendix, we present some extra results that are of interest in order to demonstrate the performances of our method and algorithm.

C.1 Simulated data

TODO: des données qui ne ressemblent pas du tout à des données M/EEG.

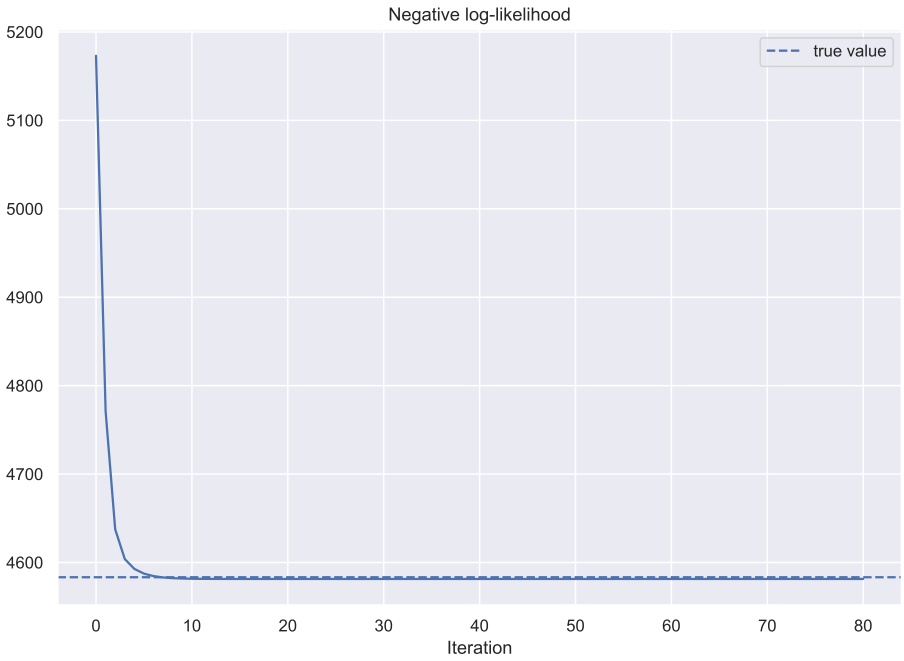


Figure C.1 – Loss history over 80 iterations with a *smart* initialisation, with $T = 100\,\text{min}$, `isi` $= 5$, `n_tasks` $= 50\,\%$, $[a\,;b] = [0\,;2000] \times 10^{-3}\,\text{s}$.

TODO: des données très similaires aux données réelles utilisées (*sample*). Note that in order to simulate data as similar as the *sample* data, the hyper-parameter `n_tasks` is no longer a percentage, but directly an integer, denoting the exact number of tasks timestamps we want to keep.

C.2 Real data

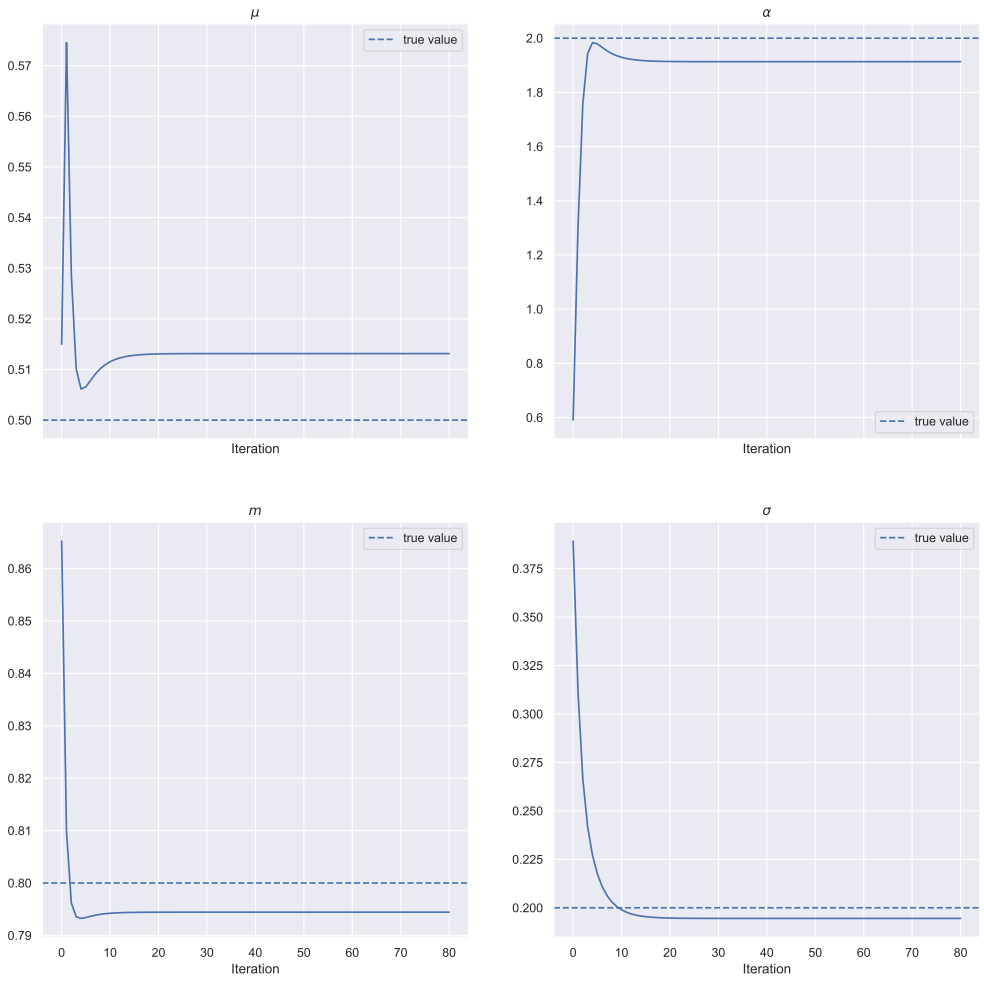


Figure C.2 – Parameters recovery over 80 iterations with a *smart* initialisation, with $T = 100\,\text{min}$, `isi` $= 5$, `n_tasks` $= 50\,\%$, $[a\,;b] = [0\,;2000] \times 10^{-3}\,\text{s}$.

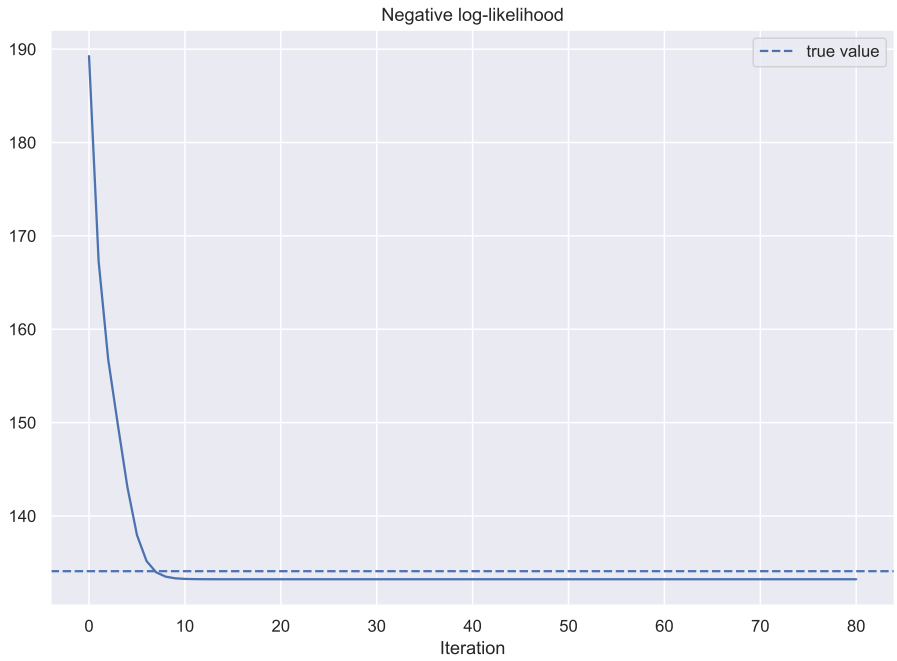


Figure C.3 – Loss history over 80 iterations with a *smart* initialisation, with $T = 4.6\,\text{min}$, `isi` $= 2.5$, `n_tasks` $= 70$, $[a\,;b] = [30\,;800] \times 10^{-3}\,\text{s}$.

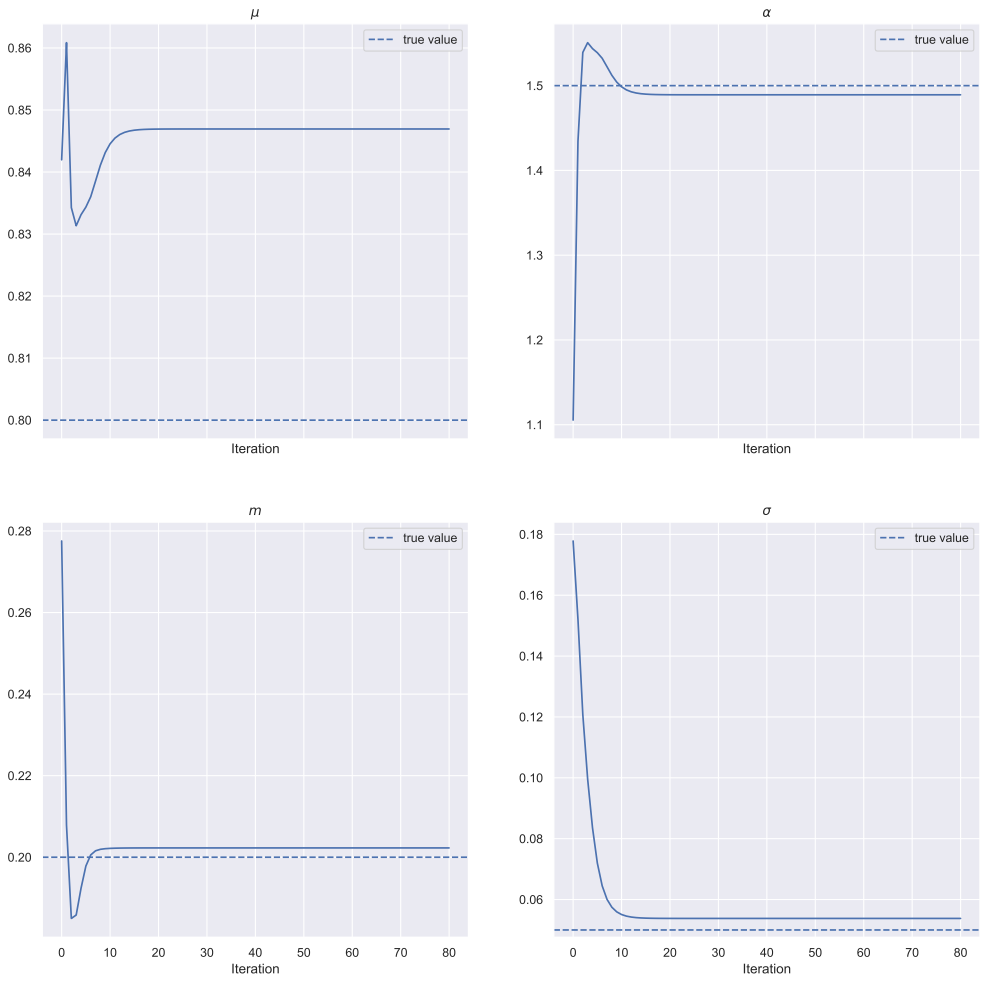


Figure C.4 – Parameters recovery over 80 iterations with a *smart* initialisation, with $T = 4.6\,\text{min}$, `isi` $= 2.5$, `n_tasks` $= 70$, $[a\,;b] = [30\,;800] \times 10^{-3}\,\text{s}$.

References

[Ach17] Massil Achab. *Learning from Sequences with Point Processes.* PhD thesis, Université Paris-Saclay, 2017.

[BBGP17] Emmanuel Bacry, Martin Bompaire, Stéphane Gaïffas, and Soren Poulsen. Tick: a python library for statistical learning, with a particular emphasis on time-dependent modelling. *arXiv preprint arXiv:1707.03003*, 2017.

[BM96] Pierre Brémaud and Laurent Massoulié. Stability of nonlinear hawkes processes. *The Annals of Probability*, pages 1563–1588, 1996.

[Bom19] Martin Bompaire. *Machine learning based on Hawkes processes and stochastic optimization.* PhD thesis, Université Paris-Saclay, 2019.

[Che16] Yuanda Chen. Thinning algorithms for simulating point processes, 2016.

[CSSBW17] Shizhe Chen, Ali Shojaie, Eric Shea-Brown, and Daniela Witten. The multivariate hawkes process in high dimensions: Beyond mutual excitation. *arXiv preprint arXiv:1707.04928*, 2017.

[DLTMJG18] Tom Dupré La Tour, Thomas Moreau, Mainak Jas, and Alexandre Gramfort. Multivariate convolutional sparse coding for electromagnetic brain signals. In *Proceeding conference on neural information processing systems (NIPS)*, 2018.

[DUGR19] Abir De, Utkarsh Upadhyay, and Manuel Gomez-Rodriguez. Temporal point processes. Technical report, Notes for Human-Centered ML, Saarland University, 2019.

[DVJ03] Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes. Volume I: Elementary theory and methods.* Probability and Its Applications. Springer-Verlag New York, 2003.

[DVJ07] Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes. Volume II: general theory and structure.* Probability and Its Applications. Springer-Verlag New York, 2007.

[GLL+13] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A Engemann, Daniel Strohmeier, Christian Brodbeck, Roman Goj, Mainak Jas, Teon Brooks, Lauri Parkkonen, et al. Meg and eeg data analysis with mne-python. *Frontiers in neuroscience*, 7:267, 2013.

[Haw71] Alan G Hawkes. Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 33(3):438–443, 1971.

[HO74] Alan G Hawkes and David Oakes. A cluster process representation of a self-exciting process. *Journal of Applied Probability*, pages 493–503, 1974.

[JLTSG17] Mainak Jas, Tom Dupré La Tour, Umut Simsekli, and Alexandre Gramfort. Learning the morphology of brain signals using alpha-stable convolutional sparse coding. In *Advances in Neural Information Processing Systems*, pages 1099–1108, 2017.

[LM11] Erik Lewis and George Mohler. A nonparametric em algorithm for multiscale hawkes processes. *Journal of Nonparametric Statistics*, 1(1):1–20, 2011.

[LS79] PA W Lewis and Gerald S Shedler. Simulation of nonhomogeneous poisson processes by thinning. *Naval research logistics quarterly*, 26(3):403–413, 1979.

[LV14] Remi Lemonnier and Nicolas Vayatis. Nonparametric markovian learning of triggering kernels for mutually exciting and mutually inhibiting multivariate hawkes processes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 161–176. Springer, 2014.

[MG19] Thomas Moreau and Alexandre Gramfort. Distributed convolutional dictionary learning (dicodile): Pattern discovery in large images and signals. *arXiv preprint arXiv:1901.09235*, 2019.

[MZ14] Behzad Mehrdad and Lingjiong Zhu. On the hawkes process with different exciting functions. *arXiv preprint arXiv:1403.0994*, 2014.

[Oga81] Yosihiko Ogata. On lewis' simulation method for point processes. *IEEE transactions on information theory*, 27(1):23–31, 1981.

[XFZ16] Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha. Learning granger causality for hawkes processes. In *International conference on machine learning*, pages 1717–1726, 2016.

Allain Cédric

ENSAE 3e année
Stage de fin d'études
Année scolaire 2019/2020

Modélisation des dépendances temporelles dans la détection des modèles récurrents en électrophysiologie (M/EEG)

—

Note de synthèse

Première section

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam. Maecenas ligula massa, varius a, semper congue, euismod non, mi. Proin porttitor, orci nec nonummy molestie, enim est eleifend mi, non fermentum diam nisl sit amet erat. Duis semper. Duis arcu massa, scelerisque vitae, consequat in, pretium a, enim. Pellentesque congue. Ut in risus volutpat libero pharetra tempor. Cras vestibulum bibendum augue. Praesent egestas leo in pede. Praesent blandit odio eu enim. Pellentesque sed dui ut augue blandit sodales. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae ; Aliquam nibh. Mauris ac mauris sed pede pellentesque fermentum. Maecenas adipiscing ante non diam sodales hendrerit.

— item 1 ;
— item 2 ?

Allain Cédric

ENSAE 3rd year
Graduation internship
Academic year 2019/2020

Modeling Temporal Dependencies between Recurring Patterns in Electrophysiology (M/EEG)

—

Summary note

First section

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam. Maecenas ligula massa, varius a, semper congue, euismod non, mi. Proin porttitor, orci nec nonummy molestie, enim est eleifend mi, non fermentum diam nisl sit amet erat. Duis semper. Duis arcu massa, scelerisque vitae, consequat in, pretium a, enim. Pellentesque congue. Ut in risus volutpat libero pharetra tempor. Cras vestibulum bibendum augue. Praesent egestas leo in pede. Praesent blandit odio eu enim. Pellentesque sed dui ut augue blandit sodales. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam nibh. Mauris ac mauris sed pede pellentesque fermentum. Maecenas adipiscing ante non diam sodales hendrerit.

- item 1;
- item 2?