University of Nice Sophia Antipolis
INRIA, Athena

An Internship Report

Jitter Adaptive Dictionary Learning for multidimensional Data

Author:
Jelena Mladenović

Supervisor:
Dr. Théodore Papadopoulo
Tutor:
Dr. Éric Debreuve

A thesis submitted in fulfillment of the requirements
for the degree of Master of Science

in the

Computational Biology and Bio-Medicine
University of Nice Sophia Antipolis

April 1 - August 15, 2015

Acknowledgments

I wish to thank my supervisor, Théodore Papadopoulo for his guidance, patience and understanding. My tutor Éric Debreuve for taking the time to help with writing this paper.

I wish to thank my friends and colleagues for being supportive, and inspiring me to work, especially Brahim Belaoucha and Christos Papageorgakis. The whole Athena team, in INRIA made this internship a warm and memorable experience.

A special gratitude I wish to dedicate to my parents for their support, specially my mother for not letting me give up in the hardest moments of my life.

Contents

Preface

Signals acquired by electroencephalography (EEG) and magnetoencephalography (MEG) have high dimensionality and low SNR, thus are hard to interpret. Sophisticated mathematical calculations and machine learning methods are constantly being developed for reducing the dimensionality and finding right representations for such complex signals.

First issue, denoising a signal by simply averaging it through multiple trials showed to have a considerable drawback in cancelling some important variability of the signal. This problem is approached in various ways, e.g., Consensus Matching Pursuit [2] that matches some predefined waveforms with the acquired signal to ensure keeping the signal variability of those specific waveforms.

Second issue, finding a correct representation of the signal while trying to reduce its dimensionality showed that it needed sophisticated mathematical calculations. In other words, for many signal classes, designing a good representation must be met, such as wavelets [21], curvelets [3] and many more. Unfortunately, finding the mentioned representations can be difficult and time consuming. After realizing the sparse nature of signals, the above problem was solved using sparse coding within a dictionary learning method [19, 21].

Lastly, using the measurements from one single electrode, means neglecting the spatial distribution of the electrical activity. Naturally, this deficit inspired the creation of methods such as differentially Variable Component Analysis [16] and Jitter Adaptive Dictionary Learning (JADL) for multidimensional data [22]. Introducing spatial distribution has enabled us to relate this problem to source localization ones. This brings us to our topic, inspired by the JADL for multidimensional data (mJADL). We will inspect whether by introducing the leadfield as a constraint, i.e. by using the information from the source space we could get a more robust approach. We will do so by showing the correlation between the leadfield columns and the learned coefficients in mJADL, as they have analogous roles. Both the leadfield and the coefficient matrix are linear operators, only mapping different sources or atoms to the EEG measurement.

Chapter 1

2.1 Introduction

The most popular methods for measuring brain activity are electroencephalography (EEG) and magnetoencephalography (MEG). Synchronous activity of millions of neurons create constructive interference that enable the appearance of different potentials that are strong enough to be measured noninvasively on the skull with a number of electrodes. In other words, constructive interference is a result of the linear superposition of well aligned features of individual neuronal activity. EEG measures this electrical property of neurons, whereas MEG uses very sensitive magnetometers to measure the magnetic field arising from the electrical currents. Thus, the sensors record the linear mixture of the source activity which is spread spontaneously within the head (volume conductor). Volume conduction can be defined as the transmission of electric or magnetic fields from an electric current source through biological tissue towards the sensors. When the electric or magnetic activity reaches the sensors, it is drastically attenuated because it is being severely space averaged within the distance from the source to the skull.

Reconstruction of the source current distribution from the measured surface fields is called the inverse problem. Its solution requires simulation of the field distribution for a current dipole in the corresponding volume conductor using the quasi-static Maxwell equations, the so-called forward problem. We will discuss these problems in more detail in chapter 2.

Both EEG and MEG (M/EEG) are noninvasive, but the EEG is preferred as it is portable and less expensive. Throughout this work we will be dealing with EEG signals although the same conclusions can be applied to MEG.

We can also mention that the signals can be evoked, in that case they are called event-related potentials (ERPs), or they can be created spontaneously. The event-related potentials are associated with the occurrence of a specific and observable event, such as muscle movements or the processing of a stimulus. The non evoked activity generated by neurons can be divided in different brain rhythms, such as: Delta (0.1-3.5 Hz, produced during deep sleep), Alpha (8-13 Hz, state of physical relaxation, but awake), Beta (14-30Hz, full wakefulness and

alert state) and others, see [24]. This division is important as different brain states or body functions can be associated to these rhythms, allowing to isolate them for further analysis. So, EEG is used for medical purposes, for discovering symptoms from insomnia to epilepsy, and many more.

EEG is also used for Brain- Computer interfaces (BCI), in which the brain activity is interpreted in order to give commands to various devices. Technological advances have enabled scientists to use BCI in Bionics, i.e. a research field which aims in providing people with high mobility disorder the opportunity to interact with their environment by manipulating artificial limbs or other devices with only their brain activity. BCI is also used in entertainment, such as video gaming and various applications sensitive to neural commands. Even though it has advanced a lot since the first BCI project [28], on account of the noisy and highly irregular nature of the signals, it is still not reliable enough to be used out of the laboratories. For that reason signal acquisition (EEG or MEG) and signal processing techniques have been improved to avoid errors, allowing the development of more robust systems and less tiring tasks for the users. There are three main steps in any BCI design [29, 18] corresponding to *signal acquisition, signal processing and application interface*, as shown in Figure 1.

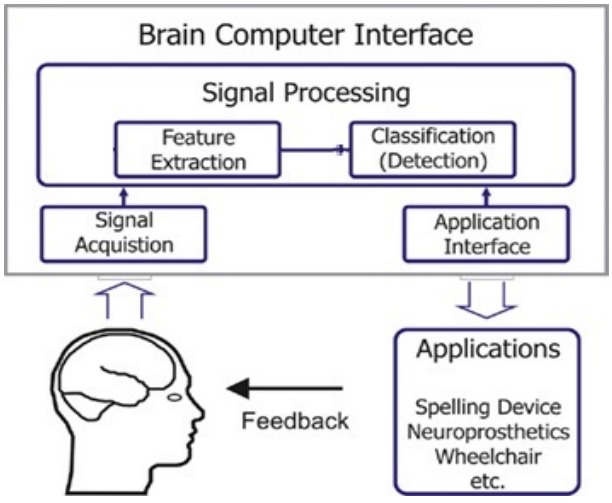

Figure 1: This figure shows the three main steps in Brain Computer Interface, signal acquisition, signal processing and device output or application interface

We are dealing with the second step, *signal processing*, which is analyzing complex signals to extract and interpret the useful information hidden in this very noisy data. More precisely, *signal processing* is composed of *(i) feature extraction*, i.e. finding reduced representation of the input signal, and *(ii) feature translation* which uses the extracted feature vector and classifies/translates it into commands to control a device. Our work is based on enhancing the (i) step, that is, representing complex data as linear combinations of a few components.

In the image bellow we have an example of a BCI spelling device (using specific ERP's called P300) used in INRIA, team Athena. We can have a better view on the *signal acquisition* technique that is used, i.e. a cap with electrodes and we can see the *device output (the 3rd step)* along with the feedback the user responds to.

Figure 2: This image shows an example of a BCI system, called speller checker, used by an Athena team member, in INRIA.

On the other hand, our work will be to introduce the notion of the source space and leadfield, which brings us to another application of the EEG signal, the source localization. Therefore, by advancing the method mentioned earlier, we will later introduce a possibility of tracking the sources of the electrical activity, as well.

To portray different approaches for the process of feature extraction, in the next section we will present the literature overview.

2.2 Literature overview

The obtained M/EEG measurements are very noisy and multi-dimensional, i.e. provide a vector of measurements at each time instant. The intuitive method for extracting waveforms (in this case one deterministic waveform d) and denoising the signal was by averaging it through multiple repetitions (trials), using the basic signal plus noise model (SPN) [6]:

$$x_m = d + \varepsilon_m, \qquad m = 1 \dots M,$$

where x_m are the signals in M number of trials. The random variable ε_m represents noise. It is usually assumed that it has a Gauss distribution. Let's say we are measuring the brain activity of one test-subject and taking into account the signal from only one electrode. Naturally, the signal we acquired is very noisy. The trivial way to induce the seemingly relevant waveform is to repeat the same test and average the signals. So averaging over m will look like:

$$\hat{d} = \frac{1}{M} \sum_{m=1}^{M} x_m = d + \frac{1}{M} \sum_{m=1}^{M} x_m.$$

In this case, we assume that the expected value of noise is zero. Being independent across trials, it will cancel out, whereas the important information will stand out and be easy to extract. We are also assuming that the relevant waveform has same phases and appears at approximately the same time within trials. Unfortunately, this is rarely the case. Because no matter if the test is done on the same subject, through multiple repetitions, we might encounter different reactions from test-subjects, such as fatigue for example. These reactions directly influence the output signal, in this example, in the amplitude change, but one might encounter phase variability and time delays (jitter) which are a lot worse cases.

It is easy to see that by simply superimposing these signals we might lose some important information.

The extension of the SPN model would contain trial-dependent latencies, and would estimate them in an iterative process [30]. Another improvement is

to use different representation for the signals (wavelets, curvelets) for a better conservation of phase and time variabilities. For the time being, we will not go into detail about methods of conserving signal variabilities, but we will remark that in this example, we have taken into account one single electrode. It has been shown that it is important to have a spatial information about the electrical activity spread out within the volume conductor. In other words, it is important to know the weight, influence of an electrode to the entire signal. An explanation is provided, in the subsection 2.5. Source Localization. Before taking into account time variability and multi-channel signal distribution, we will present the simplest approach to model the measured signals as linear combinations of different components.

Given a number of experimental trials $\{x_m\}$, $m = 1, ...M$, it is of interest to extract different components or waveforms $\{d_k\}$, $k = 1, ...K$. The measured signals are linear combinations of these components, which reads:

$$x_m = \sum_{k=1}^{K} a_{km} d_k + \epsilon_m, \quad m = 1 \dots M, \tag{1}$$

with $a_{km} \in \mathbb{R}$ being the coefficients or amplitudes of components d_k and ε_m again presenting the background noise.

If x_m and d_k are discrete signals with $N \in \mathbb{N}$ sampling points, (1) can be formulated as a matrix factorization problem:

$$X = DA + E \tag{2}$$

where the columns of $X, E \in \mathbb{R}^{N \times M}$ and $D \in \mathbb{R}^{N \times K}$ contain the vectors x_m, ε_m, and d_k, respectively, and $A = a_{km} \in \mathbb{R}^{K \times M}$ is the coefficient matrix.

An obvious drawback in the model lies in the fact that the amplitudes may be present both in D and in A. Therefore, all components d_k are normalized in the ℓ_2 -sense, such that their amplitudes (or more precisely ℓ_2 -norms) will be captured exclusively in A. Various methods have been used for reducing the dimensionality of the acquired signal such as Principal Component Analysis (PCA), which projects the input data into a lower dimensional space and Independent Component Analysis (ICA) that separates linearly independent sources. Sparse coding techniques and dictionary learning (DL) will be explained later.

Principal Component Analysis If the number of components $\{d_k\}$ we wish to extract is in a lower dimension than the signal $\{x_m\}$, i.e. $K \leq M$ we could reduce or decompose it with PCA and stay in the lower K-dimensional space. PCA uses the co-variance matrix and the goal is to (i) minimize redundancy, measured by the magnitude of the co-variance (off-diagonal elements), and (ii) maximize the signal, measured by the variance (diagonal elements) in the components $\{d_k\}$. When performing a full PCA, any number K of the first principal components gives a set of normalized, orthogonal vectors. This allows to choose the number K a posteriori without the need to recalculate the principal components. The hierarchical ordering of principal components can be very useful for data compression and dimensionality reduction, as well as for separating signal (first components) from noise (last components). However, this separation of components, variance maximization and orthogonality encourage mixing of signal components and noise. If these are of similar amplitudes and linearly correlated, PCA risks at discarding these high-correlated components and thus, we might again lose relevant variability of the signal.

Independent Component Analysis Another way to approach the decomposition (2) is to assume higher order statistical independence between the components $\{d_k\}$. This problem belongs to the class of blind source separation (BSS) methods. A prominent representative of BSS methods is the so called cocktail party problem [14]. ICA algorithms generally rely on the assumption that the components have non-Gaussian distributions, which presents one of the limitations in neuro electrical signal processing. The fact that the distributions of neural sources are often not far from being Gaussian, showed reduced use of ICA in this field of research.

These linear decompositions often discard high-correlated components with the task and impose orthogonality onto the basis vectors. Dictionary Learning method and its variants [8, 19], eliminate these risks and requirements, allowing more flexibility to adapt the representation of the data. However, they do not take into account small time variabilities of the signal, called jitters.

Methods that conserve variabilities We focus on a Dictionary Learning variant, a method called Jitter Adaptive Dictionary Learning (JADL) [13]. Predecessors of JADL such as Consensus Matching Pursuit (CMP) in [2] and differentially Variable Component Analysis (dVCA) proposed in [27] enhanced the approach in conserving different signal variability. For example, CMP uses predefined components, Gabor wavelets, but it limits the data representation, i.e., it can only extract these predefined waveforms from the signal. The dVCA model is based on the Bayes' theorem for finding the maximum probability estimates of the components within the signal. It is an alternate minimization method which starts with learning one signal component with it's estimated latency, waveform and amplitude. In each iteration a new component, along with its parameters is learned, i.e., each mentioned attribute of the component is learned separately in each iteration (alternate heuristic). As long as the residual signal still contains relevant structures, new components are added to the learning procedure. The same authors extended the dVCA model in [16] to include multi-channel recordings.

2.3 Internship Overview

This work is a continuation of a method developed on Dictionary learning, performed at Inria with the leadership of professor Théodore Papadpoulo. A method called Jitter Adaptive Dictionary Learning (JADL) [13], is based on the Alternate heuristic (or approximate solution method), since the nonlinear continuous optimization model considered is not convex and therefore has many local minima. Dictionary learning model has two types of variables: coefficients and a dictionary, which will be described in detail throughout this paper. By applying classical Alternate heuristic, nonlinear problem is decomposed into two convex and easy to solve subproblems: (i) by fixing the value of the dictionary, the coefficients are found; (ii) by fixing the coefficients such obtained, a new dictionary is being found, or simply improved. The procedure of fixing the two sets of variables is repeated until stability or convergence, i.e., until there is no further reduction in error (objective function). In other words, the procedure alternates between solving two subproblems until stability. Unfortunately, such

technique, even being easy to create, sometimes provides a solution very far from the global minimum.

JADL from [13]. JADL method also uses the Alternate idea. However, both models of the two convex subproblems are extended and made more complex to solve. The extension includes problem specific characteristics and knowledge, such as delay in which some waveforms appeared. For those delays, values of a given number of time shifts (called jitters) are added to the model. In that way the size of both the dictionary and the coefficient sets are enlarged and one needs to chose the shift that is the best, with respect to the current solution. Thus, one subproblem becomes combinatorial and hard to solve. However, it appeared to be better to solve such more realistic complex model with approximate solution method, than solving exactly the convex subproblem that does not include problem specifics (in this case jitters). Advantages of JADL over other methods are demonstrated in [13] on synthetic data and on the real data. The drawback is that it is using a single channel. Model considered also includes a repetition of the same experiment on that channel in different time slots. It is called multi-trial model.

JADL from [22]. Continuation of the work is done in [22]. Firstly, the model is extended to the case where there are several electrodes/channels taken into account, not just one. In that way three-dimensional input data (3-D matrices) are obtained: time slots $\times$ trials $\times$ channels. It is called multidimensional data (m-JADL). For such a case, three different adaptations to the original two-dimensional JADL are designed in [22]: simple method, multiplexing method and complete method. Moreover, due to the nice plotting properties of *Python* programming language, the whole *C++ with Matlab* JADL code is transferred to Python. Again, promising results are obtained on both, synthetic and real data sets.

Main purpose of this Internship. The main research is again, done on the continuation of the previous work in [13, 22]. One parallel to Dictionary learning problem is the localization of sources of signals (waves) within the human head. The goal of this internship was to understand the relation between the Dictionary Learning method and the source localization one. By presenting the measurements, or the EEG signals as a linear model, we are able to create

that relation. In the linear model, a source matrix S is mapped by a linear operator, called lead field or gain matrix G. In other words, the gain matrix G maps the active sources (signals in time) onto the electrodes/channels, obtaining the EEG measurements; in the same way the coefficient matrix maps the atoms from the Dictionary to the EEG measurements. An interesting question that was not fully addressed in [22], but mentioned in Conclusion section, is the correlation between these two approaches: Dictionary learning and Source localization. More precisely, the question is *-could the number of sources in Source localization be equal to the number of atoms in Dictionary learning?* If we assume so, then the positive answer to this question will follow if there exists a correlation between G and the coefficient matrix. In this internship we addressed the question whether the coefficient matrix can be correlated with the lead field and then performed a series of experiments on the same data sets as used in [22]. We also showed that with such correlation, we might use that information for the *inverse problem* of the source localization.

Outline. In the next Chapter we first explain how data are collected and represented. Then, in Section 2.2, we give a detailed view of Dictionary learning problem and its connections with some known optimization techniques from Numerical analysis and Operations research. Jitter adaptive dictionary learning method is then explained, together with some possible improvements. This chapter is concluded with the basic facts regarding Source localization problem. In Chapter 3, rules of the new algorithm which connects two approaches: dictionary learning and source localization, are explained. Chapter 4 contains computational results with method described in Chapter 3. It also includes brief description of generating data sets. At the end, in Chapter 4, the conclusions and suggestions for the possible future work are drawn.

Chapter 2

In this chapter, in order to provide a full understanding of the signal processing methods and feature extraction, in section 2.1 we discuss some popular data representations and explain the meaning of sparseness, which became an important concept in neural signal processing. Dictionary Learning with its novel variant, Jitter Adaptive Dictionary Learning, is described in 2.2 and 2.3, respectively. In section 2.4 the multi-electrode or multi-channel dictionary learning algorithm will be explained. The introduction of source localization problems, the Forward and Inverse problems are given in section 2.5.

3.1 Data representation

Fourier Transform. Signal processing techniques are selected according to certain signal characteristics. For example, the time domain representation gives information about the maximal time resolution, i.e, the intensity of the signal at time t, but no frequency information. Same implies to the frequency representation, where we can notice the mutual exclusion of each representations. However, a signal can be represented as a linear superposition of sines and cosines, characterized by their frequency f (or period $T = \frac{1}{f}$). The Fourier Transform (FT) allows a stationary signal $x(t)$ to be defined as the inner product of $x(t)$ and the complex sinusoidal function e^{-iwt}

$$\sum_t x(t)e^{-iwt} = \sum_t x(t)(\cos(wt) + i\sin(wt)),$$

where $w = 2\pi f$. FT has two disadvantages, (i) the signal is often not stationary, (ii) the lack of time information about the evolution of frequencies.

Short Time Fourier Transform. A new method was developed to overcome these issues, called Short Time Fourier Transform (STFT) or Gabor Transform. It allows to analyze a small signal section multiplying the signal $x(t)$ by a window function $g(t)$ that slides along the time axis before determining the frequency spectrum. The STFT results in describing the signal in time and frequency, giving a two-dimensional representation. The assumption of this method is that the signal viewed within a window is stationary, and sometimes it is difficult to

find appropriate window size where the signal has that property. The famous uncertainty problem states that it is impossible to know at the same time two physical magnitudes such as the position and the velocity of a particle, meaning that in our case, by reducing the time window one will increase the temporal resolution but also decrease the frequency resolution. The main disadvantages of STFT is that the width of the window is fixed for the entire signal analysis, implying a strict frequency and time resolution. In addition, the STFT is not able to give information if the windowed signal is not stationary. This leads to the development of the Wavelet theory to be able to represent non stationary signals in the time-frequency domain.

Wavelet theory. As EEG signals are not stationary, i.e., several rhythms vary at the same time, the Wavelet theory seems fit for data representation. As STFT, wavelets convolve the signal to be analyzed with the window function, but it differs in the sense that the window size is not fixed. The window size adapts to different frequencies, such as higher frequencies are analyzed using narrow windows to obtain good time resolution, and lower frequencies are analyzed with wide windows to obtain good frequency resolution.

Wavelets have been popular choices not only because of their good mathematical properties such as orthogonality, shift and scaling invariance, but they have shown to be suited for sparsely representing natural signals and images [21]. In addition, they facilitate the interpretation of signals by representing them in the time-frequency domain. An advantage of using complex-valued time-frequency representation is the possibility to treat magnitude and phase of each frequency component separately.

Sparsity. Since natural images and signals contain information from limited frequency spectra, sparsity has shown to be a very useful property, especially in the case of time-frequency representations. The current amplitudes generated by a neural source, for example, have a sparse distribution over recording channels as they are only measurable in sensors nearby. Sparsity may not only occur across channels, but it has been observed that certain waveforms are present only in a subset of trials.

Something being sparse means that a given data can be described in a small number of basis functions chosen out of a larger set. Given a basis of a vector

space V, every element of V can be expressed uniquely as a linear combination of basis vectors. The simplest description would be to consider a linear system of equations $x = D\alpha$, where D is an underdetermined $m \times p$ matrix ($m \ll p$) and $x \in \mathbb{R}^m, \alpha \in \mathbb{R}^p$. D is given and called the dictionary or the design matrix. The goal is to estimate the signal α, subject to the constraint that is sparse. The underlying motivation for sparse decomposition problems is that even though the observed values are in high-dimensional (m) space, the actual signal is organized in some lower-dimensional subspace ($k \ll m$). Sparsity implies that only a few components of α are non-zero. This further implies that x can be decomposed as a linear combination of only a few column vectors in D, called atoms. The sparse decomposition problem is represented as,

$$\min_{\alpha \in R^p} \|\alpha\|_0 \quad s.t. \quad x = D\alpha$$

where $\|\alpha\|_0$ is a pseudo-norm, l_0, which counts the number of non-zero components of α. This problem is NP-Hard, being a subset of selection problems in combinatorial optimization. A convex relaxation of the problem can instead be obtained by taking the l_1 norm instead of the l_0 norm, where $\|\alpha\|_1 = \sum_{i=1}^{p} |\alpha_i|$. The l_1 norm induces sparsity under certain conditions which we will explain in more detail in the next subsection when describing the sparse coding method.

Traditionally, sparse decomposition was performed over predefined dictionaries that were known to yield sparse representations, such as windowed sinusoidal functions or different types of wavelets, such as Gabor wavelets in Consensus Matching Pursuit [2]. Finding the optimal design of the dictionary and calculating the sparse representation in one alternating method, is known as dictionary learning. Both concepts are presented in the following subsection.

3.2 Dictionary learning

We first explain the simple model that covers the experiment with a single electrode, without repetition. Then we extend this basic model in two directions: multi-trial case and multiple electrode case.

Simple model. Data obtained by measuring signals on one electrode in N time segments are denoted by $x_\ell \in R$, $\ell = 1, \ldots, N$. One wants to map them from

R^N to R^K, with K number of vectors d_i and unknown coefficients a_i. In other words, it is necessary to consider the linear transformation D that maps R^N to R^K:

$$Da = x, \tag{3}$$

where linear operator D is represented by matrix $D \in R^{N\times K}$. If D and x are given, then the problem transforms to solving the system of linear equations with K unknowns and N equations. Of course, if $K = N$ and $det(D) \neq 0$, we have a single solution a_i^* that satisfies $Da = x$. If $K < N$, which is our case, then the system is over-determined and has a solution if the over-determination is due to redundancy. One would like to get a solution whose sum of square differences of right and left hand sides of (3) is minimum

$$(\min_a) f = \frac{1}{2}\|x - Da\|_2^2. \tag{4}$$

or, after including the definition od Euclidean norm, we have

$$(\min_a)\ f = \frac{1}{2}\sum_{\ell=1}^{N}\Big(x_\ell - \sum_{i=1}^{K} d_{\ell i}a_i\Big)^2 \tag{5}$$

Note that f is a convex function, since its Hessian is equal to the unit matrix E and thus is positive definite. Note also that in the case $K = N$, the solution of (5) is also the solution of (3), obtained by solving the equivalent system $(D^TD)a = D^Tx$. It appears that the solution of (5) is not stable, i.e., a small change in data, that are also noisy, produce a large change in coefficients a_i. Therefore, some regularization technique is welcome. The usual way to overcome this difficulty is by adding the so called Lasso part in the minimization of f:

$$(\min_a) f = \frac{1}{2}\|x - Da\|_2^2 + \lambda\|a\|_1, \tag{6}$$

where $\lambda > 0$ is a regularization parameter. By using the definition of the Euclidean norm, the latest minimization problem may be written as

$$(\min_a)\ f = \frac{1}{2}\sum_{\ell=1}^{N}\Big(x_\ell - \sum_{i=1}^{K} d_{\ell i}a_i\Big)^2 + \lambda\sum_{i=1}^{K}|a_i|, \tag{7}$$

In addition, the large value of λ will force more zero values in the final solution. λ is a penalty/regularization parameter as it makes the balance between data

fidelity and constraint enforcement. In other words, λ is controlling the trade-off between the data fitting and regularization. We have empirically seen that by changing λ, we control the sparsity, which is an increasing function of λ. When λ is near zero, there is no sparsity and as λ increases the solution becomes sparse.

Up to now, we assumed that the matrix D which maps R^N to R^K is known. From the problem nature, this is not the case. We would like to minimize the objective function (3) with respect to a, but for the best possible matrix D. Therefore, the real model becomes more complex. Nonlinear programming model obtained contains products of variables a_i and $d_{\ell i}$.

$$(\min_{a,D})\ f = \frac{1}{2}\sum_{\ell=1}^{N}\Big(x_\ell - \sum_{i=1}^{K} d_{\ell i}a_i\Big)^2 + \lambda\sum_{i=1}^{K}|a_i|, \tag{8}$$

This constrained nonlinear problem is no more convex since it contains a product of variables, and therefore, it can not be easily solved. Exact solution methods are still not proposed in the literature to solve it. One possible way in that direction could be to transform the problem into biliner form and then apply some bilinear exact solution method [1].

Alternate heuristic. On the other hand, one can see that a heuristic that alternatively solves two convex problems may lead to some feasible solution:

(i) for a given matrix D find vector a that minimizes f;

(ii) for such obtained coefficients a_i, find the matrix D that minimizes f;

(iii) repeat steps (i) and (ii) until there is no improvement of f when compared with its value in the previous repetition (iteration).

Note that the idea of solving alternatively two easier problems made from a complex one, is used in many scientific fields. For example, in solving the minimum sum of squares clustering problem. The Alternate method is known as k-means heuristic [17]; in solving the location-allocation problem, it is known as Cooper's heuristic; [5]; in solving the pooling problem in the oil industry, it is known as refirement method [11], etc. Unfortunately, Alternate heuristic methods sometimes stop in a local minimum with bad quality, i.e., solution whose

function value f is very different from the global exact minimum. Therefore, a lot of research efforts in the literature are devoted to improve the quality of a pure Alternate heuristic.

The further development of this alternate heuristic, in our problem, includes introducing new terms and notations. As we already know, matrix D is called *dictionary* and the process of getting their new values from iteration to iteration is called *dictionary learning*. Columns of matrix D are called *atoms*, thus there are K atoms. What we haven't mentioned is that finding coefficients a_i in step (i) of the Alternate heuristic is called *sparse coding*. Finding matrix D in step (ii) of the Alternate heuristic is called *dictionary update*. Thus, the problem is to construct a dictionary learning procedure that includes solving two more simple problems alternatively, and whose error f will be as small as possible.

Just to recapitulate, the coefficients are learned using the sparse coding, which yields to maximize the number of zero elements, reducing the number of active components within the dictionary. As we mentioned earlier, the l_0 norm is solving this issue, but as it is NP hard we are introducing a relaxation with l_1 norm, which is called the Lasso problem and is solved with the Least Angle Regression approach (LARS) in [7].

Extended model with more trials. The discussion in the previous subsection covers the case of getting signals using a single electrode in N time intervals (seconds). The natural extension is to repeat the same experiment M times. Repetition of the same experiment, in the different time intervals, increases the reliability of input data that are influenced by noise. So, input values are $x_\ell^{(j)}$ ($j = 1, \ldots, M$, $\ell = 1, \ldots, N$). Then the extended model with M trials, and a given matrix D, has a form

$$(\min_{a_j \in R^K})\ f_j = \frac{1}{2}\|x_j - Da_j\|_2^2 + \lambda\|a_j\|_1,\ \ \forall j = 1, \ldots, M, \tag{9}$$

or

$$(\min_{a_j})\ f_j = \frac{1}{2}\sum_{\ell=1}^{N}\left(x_\ell^{(j)} - \sum_{i=1}^{K} d_{\ell i}a_i^{(j)}\right)^2 + \lambda\sum_{i=1}^{K}|a_i^{(j)}|,\ \forall j = 1, \ldots, M. \tag{10}$$

The natural way to get one common objective function out of M is to sum them. In addition, a constraint that prevents atoms from growing arbitrarily large should be taken into account. The resulting constrained nonlinear program is as follows:

$$(\min_{a_j, D}) \sum_{j=1}^{M} f_j = \sum_{j=1}^{M} \left(\frac{1}{2} \sum_{\ell=1}^{N} \left(x_\ell^{(j)} - \sum_{i=1}^{K} d_{\ell i} a_i^{(j)} \right)^2 + \lambda \sum_{i=1}^{K} |a_i^{(j)}| \right), \qquad (11)$$

subject to

$$\sum_{\ell=1}^{N} d_{\ell i}^2 = 1, \ \forall i = 1, \ldots, K. \qquad (12)$$

Note that in problem (11) - (12) unknown variables are atoms of the dictionary as well as vector a. Therefore, Alternate algorithm from above, as well as its variants can be applied as a mean to solve this problem.

Dictionary Learning in literature. In dictionary-based approaches, an expert selects a specific family of basis functions (atoms), known to capture important features of the input data, for example wavelets [21] or curvelets [3]. When no specific expert knowledge is available, dictionary learning algorithms could learn those atoms from a given dataset. The problem is then stated as an optimization procedure (as defined in (8) or (11)-(12)) usually under some sparsity constraints. Thanks to the pioneering works on sparsity constraint decomposition of Mallat and Zhang, see in [20], on the Lasso problem [26] and on sparse signal recovery [4], efficient algorithms are available both for projection on overcomplete representations [7] and dictionary learning [8, 19].

Learning a dictionary instead of using predefined basis, has shown to improve dramatically the signal reconstruction [8]. A sparse approximation of a signal $x \in R^M$ over a dictionary $D \in R^{M \times K}$ with K columns or atoms is when we can find a linear combination of a few atoms that is "close" to the signal x.

While sparse coding uses a model of linear combinations of signal components (like PCA and ICA), it strongly differs from these techniques qualitatively. This is due to the fact that a dictionary typically contains a large number of atoms and is often shift- and scale-invariant. In this sense, these dictionaries are well suited to capture temporal variability such as latency jitter, change

of duration, frequency or phase. However, in many cases the shapes of the atoms do not well represent the characteristic ERPs. These potentials are often asymmetric and not well-localized in time-frequency domain. Hence, in the case of a Gabor dictionary, for instance, a large number of these symmetric, time-frequency localized atoms is needed to encode the ERP shape. As the shapes of the ERPs are usually not exactly known a priori, it is not clear how to design the optimal dictionary. Instead of defining it beforehand, it may therefore be beneficial to learn it together with the decomposition.

We have presented the main concepts of Dictionary Learning (DL), taking first into account the single electrode/single trial case. After expanding the model to the single electrode/multi-trial case, we are ready to show the multi-electrode/multi-trial one. However, this model was used onto a variant of DL which takes into account time variabilities. Therefore, we will primarily provide the description of this DL variant, called Jitter Adaptive Dictionary Learning (JADL) [13] and then introduce the multi-electrode JADL model [22].

3.3 Jitter Adaptive Dictionary Learning

Description. One possible way to reduce the error of original Alternate procedure, proposed in [13] is called Jitter Adaptive Dictionary Learning (JADL). It represents a brilliant way of taking into account the time variabilities of the signal without inducing much the computational complexity of the Dictionary Learning Algorithm. The D matrix is being enlarged by S possible different time occurrences of a waveform, and then reduced back to its initial dimension once the new time occurrence (delay) has been chosen. In other words, the dictionary is being expanded with the shifted versions of each atom; then, with sparse coding, we are choosing the best shift operation and cancelling other unnecessary shifts that were added, thus reducing the dictionary to it's initial dimension.

The shift operation δ is a function $\delta \in \Delta$, where Δ is a set of small shifts (called jitters) relative to the size of time window. Let us denote its cardinality with $S = |\Delta|$. Formally speaking, each element of the matrix D may be chosen

among S possible proposed values (shifts). In that way, the continuous problem of finding new atoms in the next iteration of Alternate heuristic (i.e., within dictionary learning step of the algorithm), is transferred to discrete one; we need to find one, out of S possible values around the current value of $d_{\ell i}$.

$$D^S = \begin{bmatrix} d_{111} & d_{112} \dots d_{11S} & d_{121} \dots d_{12S} & \dots & d_{1K1} \dots d_{1KS} \\ d_{211} & d_{212} \dots d_{21S} & d_{221} \dots d_{22S} & \dots & d_{2K1} \dots d_{2KS} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{N11} & d_{N12} \dots d_{N1S} & d_{N21} \dots d_{N2S} & \dots & d_{NK1} \dots d_{NKS} \end{bmatrix}$$

In other words

$$d_{\ell is} = d_{\ell i} + \delta_{is},\ \delta_{is} \in \Delta,\ \ell = 1, \dots, N,\ i = 1, \dots, K,\ s = 1, \dots, S.$$

Thus, the matrix D^S has dimension $N \times K \cdot S$. The implementation of dictionary learning JADL method consists of two basic steps: (i) sparse coding; (ii) dictionary update.

Sparse coding step. By introducing set Δ and jitters, the LARS method that was used to solve step (i) in Alternate heuristic should be modified. Since the number of columns of matrix D increases S times, the size of the unknown vector a should increases to $K \cdot S$ as well to allow multiplication $D^S a_j^S$:

$$a_j^S = (a_{11}^{(j)}, \dots, a_{1S}^{(j)},\ |\ a_{21}^{(j)}, \dots, a_{2S}^{(j)}, | \dots, | a_{K1}^{(j)}, \dots, a_{KS}^{(j)})^T$$

The problem is as follows:

$$(\min_{a_j^S}) f_j = \frac{1}{2} \|x_j - D^S a_j^S\|_2^2 + \lambda \|a_j^S\|_1, \tag{13}$$

subject to

$$\|a_{ij}^S\|_0 \leq 1,\ i = 1, \dots, K. \tag{14}$$

The constraints (14) present that the number of non-zero elements in each segment of vector a_j^S (out of K of them) can be at most one. In [19] is used an additional constraint in which these elements can be only 1 or 0. This condition in fact makes problem (13)-(14) non convex. In other words, this problem belongs to combinatorial optimization, since unknown values $a_{is}^{(j)}$ belongs to the set {0,1}. It can be reformulated as follows:

$$(\min_{a_j^S}) \sum_{j=1}^{M} f_j = \sum_{j=1}^{M} \left(\frac{1}{2} \sum_{\ell=1}^{N} \left(x_\ell^{(j)} - \sum_{i=1}^{K} \sum_{s=1}^{S} d_{\ell is} a_{is}^{(j)} \right)^2 + \lambda \sum_{i=1}^{K} \sum_{s=1}^{S} |a_{is}^{(j)}| \right), \quad (15)$$

subject to

$$\sum_{s=1}^{S} a_{is}^{(j)} \leq 1, \ \forall i, j \quad (16)$$

$$a_{is}^{(j)} \in \{0, 1\}, \ \forall i, j, s. \quad (17)$$

Constraint set (16) indicates that in each part of vector a_i^j with S elements, at most one should be equal to 1. Constraint set (17) indicates that all variables are Boolean.

Sparse coding problem (15) - (17) is defined as nonlinear 0-1 program, which is clearly not easy to solve exactly. In [13], a local search (LS) heuristic is used to get a local minimum. The solution is represented by vector which has K arrays with length S. The neighborhood is defined by swapping each variable whose current value is 1 with another variable with zero value, within the corresponding array with S elements. The cardinality of such neighborhood is obviously $K \cdot S$. This number is much smaller than S^K, being the cardinality of the whole solution space of this sub-problem. The step of finding the best swap is repeated until there is no improvement in the error f.

LS example. To explain better the idea of solving hard sparse coding step by local search heuristic, we will use an example. Assume that $S = 4$ and $K = 2$. Assume further that the current value of vector a of size 8 is: $a = (0100|0010)$. We want to check if change of some element of a from 0 to 1 will improve the objective function f. Since constraint (16) should be satisfied, change of any zero to one means that one, in each of the two parts of the solution a should become zero. The complete neighborhood $\mathcal{N}$ of solution a is then defined as

$$\mathcal{N}(0100|0010) = \{(1000|0010), (0010|0010), (0001|0010), (0000|0010)$$

$$(0100|1000), (0100|0100), (0100|0001), (0100|0000)\}.$$

The cardinality of $\mathcal{N}(a)$ is obviously equal to 8 ($|\mathcal{N}(a)| = K \cdot S = 2 \cdot 4 = 8$). Within LS, we check the value of function f in each shifted value from $\mathcal{N}(a)$.

If there is an improvement, we move to that solution and repeat the same step (define its neighborhood, etc). In case that the current solution is the best among their 8 neighboring solutions, we get local minimum of the problem and proceed to the dictionary update step.

Dictionary update. When the sparse coding step is executed, the dictionary learning step is transformed into the following combinatorial optimization problem:

$$(\min_{D^S}) \sum_{j=1}^{M} f_j = \sum_{j=1}^{M} \left(\frac{1}{2} \sum_{\ell=1}^{N} \left(x_\ell^{(j)} - \sum_{i=1}^{K} \sum_{s=1}^{S} d_{\ell i s} a_{is}^{(j)} \right)^2 \right) \tag{18}$$

subject to

$$\sum_{\ell=1}^{N} \sum_{s=1}^{S} d_{\ell i s}^2 = 1, \ \forall i = 1, \ldots, K. \tag{19}$$

It should be noted that the mixed integer nonlinear problem (18)-(19) is now NP-hard, since there are exponential number (S^K) of possible values of D after choosing δ_{is} values, for each $d_{\ell i}$. Therefore, the subproblem (ii) from dictionary learning algorithm becomes hard to solve as well. However, it could be solved approximately, if S and K are not so large.

In [13] a local search approach of this problem is also implemented. Based on the first order conditions, the updating formula for the new set of atoms is derived. Jitter values δ are also chosen in the vicinity (neighborhood) of the current value of d_{ijs}.

Computational results in [13] show how results gained by JADL method are better than with the original alternate algorithm.

In the next subsection we present the multi-dimensional JADL (mJADL)[22], the multi-electrode/multi-trial case, we mentioned earlier. Three models were proposed presenting the signals in three dimensions, that follows: *different channels, trials and time instances.* We will discuss these models in detail in the next section.

3.4 Dictionary learning from multi-dimensional data

During the EEG recording, different parts of the brain are active at each time instant but do not produce the same pattern of activity. By using a single

electrode, only a subset of the brain activities was considered and modelled at a time instant. Multiple repetition approach degrades the shapes and timings of the activities and hides their inherent variability. And EEG are inherently multi-dimensional, i.e. provide a vector of measurements at each single time instant. It is obvious that increasing the number of channels (electrodes), much more signals can be collected. They should give more precise information regarding brain activities we are trying to model and recognize. We talk about *multi-dimensional* EEG data measurements, as presented in Figure 3.

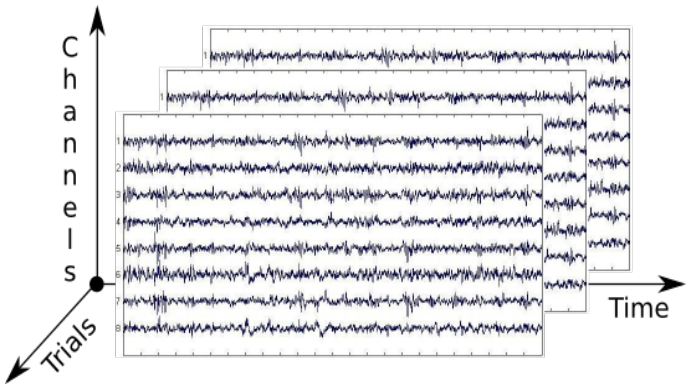

Figure 3: Multi-dimensional data: three dimensions are taken into account: time segments, trials and channels/electrodes, from [22]

Each point in this 3-D space has coordinates $(i, m, c) \in R^3$, $(i = 1, ..., N$, $m = 1, ..., M$, $c = 1, ..., C)$ representing the time i when a signal is kept, in trial m and by electrode c, respectively.

In [22] the spatial dimension of the activity (its distribution across electrodes) is considered. Dictionary learning methods explained earlier are adapted for solving this multidimensional case. In fact, the jitter adaptive dictionary learning (JADL) method is modified to handle multidimensional measurements. The basic question is how to use the additional data set within JADL. In [22], three different ways are proposed:

Simple model. The simplest one is to consider additional electrodes as increase of the number of trials. The input data belongs to $X \in R^{N \times M \cdot C}$. Therefore, the only change is to set new M to be equal to $M \cdot C$. All other parameters remain the same, including the K atoms (learned dictionary). A basic drawback

of this model is the fact that the same set of jitters Δ is used for all channels.

Multiplexing model. In this model projection to 2-dimensional case is done in different way. The input matrix X is presented as $X \in R^{N \cdot C \times M}$. This means that the number of rows is increased C times. In other words, for each time unit, the values of signals over all C electrodes are averaged. Dictionary also increased its dimension: $D \in R^{N \cdot C \times M}$. More details could be found in [22].

Complete model. This model learns a single dictionary over multidimensional recording that have the same waveform and jitter, but different coefficients over the channels. Such an approach requires methodological changes within two JADL steps: more in (i) Sparse coding and less in (ii) Dictionary update.

Sparse coding. The LARS algorithm that was used earlier for finding coefficient a_j is modified into two steps as follows.

1. *Atom selection.* The JADL is applied S times for each channel, in the way explained in the previous subsection. As result, S different dictionaries are obtained and the best one D_S selected (called *compressed dictionary*). In other words, the set of atoms are obtained by the following formula:

$$(\max_{d_j^S \in D^S}) g = \sum_{c=1}^{C} \|s_c d_j^S\|, \tag{20}$$

   where C is the number of channels (electrodes), s_c represents the signal of channel c and d_j^S is the j-th atom of the extended dictionary D^S.

2. *Standard LARS.* Standard LARS sparse coding is implemented, taking a set of atoms obtained in the first step. Hence, multidimensional coefficients a_{ijc} are found using the compressed dictionary D_S for the current trial.

Dictionary update. Since the number of the learned coefficients is increased, when compared to original JADL, the dictionary update step (ii) should be modified as well. The more precise implementation of this step may be found in [22].

The advantages of multidimensional complete model over the single channel JADL is the fact that all electrodes are used in decision making process, finding one that best represents the model. Computational results on synthetic, as well as on real data sets confirm this claim.

3.5 Source Localization and Leadfield

As we mentioned earlier, the purpose of this internship is to find a relation between two fundamental problems in signal processing, i.e., *Dictionary Learning* and *Source Localization*. So far, we were addressing the dictionary learning approaches and now we must introduce the main concepts in source localization.

The aim of source localization is to find the brain areas responsible for EEG waves of interest. It consists of solving forward and inverse problems, presented as

$$M = G \cdot S, \tag{21}$$

where $M \in R^{C \times N}$ contains the EEG measurements in a single trial (or average measurements of all trials), matrix $G \in R^{C \times U}$ is called *lead field* or *gain* matrix; matrix $S \in R^{U \times N}$ is called source matrix. Also, C are number of channels, N time segments and U number of sources. The source matrix represents the signals created in a number of active sources within a time window. The gain matrix is a linear operator that maps the source matrix with a number of channels, onto the EEG measurements. In other words, it consists the influence of each active source onto each channel.

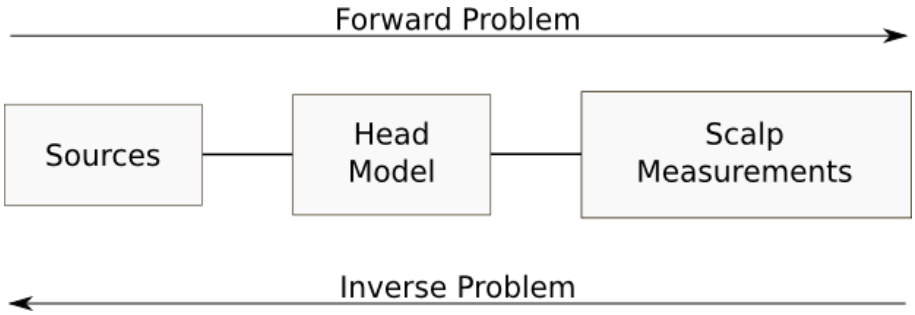

Figure 4: This figure depicts the Forward and Inverse Problem of M/EEG, from [22]

Basically, the source localization means, as the name reveals already, finding the locations of sources that were active during some brain process, within a time window. As it is an ill-posed problem, in order to get a solution, we must

first solve the forward problem. It means to find the source representation and calculate the EEG with a knowledge of the head conductivity and the gain. Then we can deal with the inverse problem, which is the actual concern, i.e., searching for active sources. Let us separately, in short explain the concepts of the Forward and Inverse problems.

Forward problem. To solve the forward problem we need the *head model* and the *source model*. The first can be represented as (i) a single layer or 3-4 layer sphere or (ii) realistically, obtained from 3D medical images, such as MRI. After gaining medical images (the (ii) case), certain numerical methods are implemented in order to get a realistic geometry of the head, a mesh. Those methods are quite famous in the source localization field, and are called as follows, the Boundary Element Methods (BEM), Finite Element Methods (FEM) and Finite Difference Methods (FDM). [10]

As for the sources, in the case of (i), they are often presented as dipoles that generate electrical currents. This method is quite fast but inaccurate. Each dipole has 7 variables, 3 for location and orientation (in 3D) and 1 for magnitude. To make the forward problem easier to solve, one fixes one or two of these properties of dipoles. The Figures 6 and 7 might clarify this explanation, and more about dipoles can be found here [25].

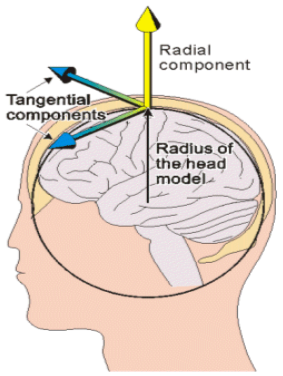

Figure 5: A representation of the spherical model, showing an orientation of a source/dipole orthogonal to the skull

As for the case of (ii) there is an approximation of a realistic source config-

uration, calculated with methods mentioned above (BEM, FEM, FDM), we get a distributed source model. Within these methods, the location and orientation of sources or dipoles are usually fixed.

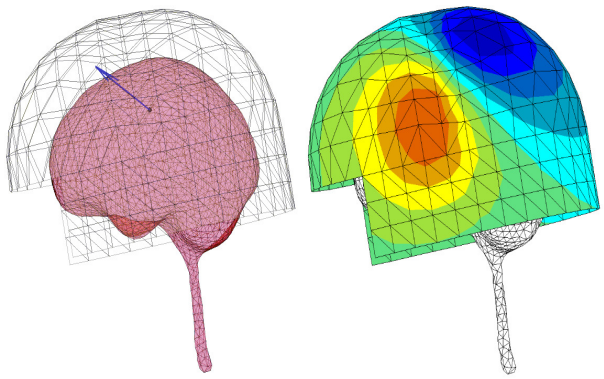

Figure 6: An example of a representation of the BEM head model and a dipole source model. An accurate mesh (number of nodes 1995) describing the outer surface of the brain was created with the SOVITA program, from [15]

With the knowledge of the source models and the head geometry, whether being the (i) or (ii) case, we can measure the lead field matrix directly. Thus, with the information about the source space S and the lead field G, we can easily calculate the EEG measurement; thus finalizing the forward problem.

Although, for all this to work, we need to know the propagation of the current within the head and the head conductivity, thus some physical properties of the head. It can be modeled by a Poisson's differential equation, with Neumann and Dirichlet boundary conditions, and the knowledge of the quasi-staticity phenomena. We won't get further into the physical nature of the problem, but for a curious reader, more information can be found here [10, 23]. Thus, once we have the EEG measurements from the Forward problem, we then can address the inverse model.

Inverse problem. As we already mentioned, this problem is ill-posed, presented as

$$S = G^{-1}M$$

So, with the known measurements M, the inverse problem refers to finding the source matrix S. It has infinite number of solutions, therefore we need to apply some additional constraints and regularization. As in the forward model, there are two principal groups of models solving the inverse problem, (i) Equivalent dipole methods and (ii) Linear distributed methods. More about the inverse problem can be found in [9]
The (i) group is characterized by

- overdetermined system
- searches for parameters of a number of dipoles
- nonlinear optimization techniques
- may converge to local minima
- known methods: non-linear least squares, beamforming, MUSIC, simulated annealing, Genetic algorithms, etc.

The (ii) group is characterized by

- under determined system
- searches for activation in given locations
- linear optimization techniques
- needs additional constraints
- known methods: Bayesian methods, MNE, LORETA, LAURA, etc.

Our model of source localization. The model we are using is the realistic head model, a mesh with a dense source configuration, found with the BEM numerical solver. We will discuss more about our case of the Forward Problem in the section 5.1 Synthetic Data and Leadfield. As for the Inverse problem, we revealed a possibility of reducing it's computational complexity by using the coefficient matrix within the m-JADL method. Note that with the original JADL method, such a relation wouldn't be possible as it is using only one electrode in

multiple trials, thus loosing the information about the source distribution. We will also discuss the general relation of the Source Localization with the multi-channel or multi-dimensional Dictionary Learning approach in 4.2 Dictionary Learning and Source Localization.

In the next chapter we will present the relation of the Dictionary Learning methods and Source Localization ones. In order to do so, we must explain the Statistical techniques used for achieving the relation between these two fields.

Chapter 3

As said earlier, the main purpose of the work in this chapter is continuation of the previous work done at Inria research center [13, 22]. In this chapter we describe our work. It consists first in finding the relationship between the two different approaches in analysis of the EEG signals obtained in measurements: Dictionary learning and Source localization. Next we analyse correlation between any two trials, from different signal sources, but taking into account all channels. For that purposes we suggest two algorithms.

4.1 Preliminaries

Average signal values over all time slots can be considered as random variables. Therefore, for completeness, we start this chapter with definitions of the basic terms from Probability and Statistics that will be used later. The connection between Probability and Linear algebra is briefly presented. The technique from Linear algebra, known as Singular value decomposition (SVD) is outlined as well. It will be used in our study presented in this chapter.

4.1.1 Correlation coefficient.

Let us consider two discrete random variables (r.v. for short) X and Y with the same size n and the random variable of their product XY:

$$X = \begin{pmatrix} x_1 & x_2 & \dots x_n \\ p_1 & p_2 & \dots p_n \end{pmatrix};\ Y = \begin{pmatrix} y_1 & y_2 & \dots y_n \\ q_1 & q_2 & \dots q_n \end{pmatrix};\ XY = \begin{pmatrix} x_1y_1 & \dots x_ny_n \\ r_1 & \dots r_n \end{pmatrix},$$

where p_j, q_j and r_j are corresponding probabilities. Let further $E(X)$ and $Var(X)$ denote mathematical expectation and variance (or dispersion) of r.v. X respectively:

$$E(X) = \sum_{i=1}^{n} x_i p_i;\ Var(X) = E(X - E(X))^2 = E(X^2) - (E(X))^2.$$

Covariance and correlation between two random variables X and Y are then defined as

$$Cov(X, Y) = E[(X - E(X)) \cdot (Y - E(Y))] = E(XY) - E(X)E(Y).$$

$$Cor(X,Y) = \frac{Cov(X,Y)}{\sqrt{V}ar(X) \cdot \sqrt{V}ar(Y)}$$

From above equations it is clear that $Cov(X,X) = Var(X)$.

Independence between two vectors is very important relation in probability theory. It requires that the expected value of $X \cdot Y$ is equal to product of each expected value: $E(XY) = E(X) \cdot E(Y)$. Therefore, if two random variables are independent, their covariance and variance are equal to zero.

From the definition of covariance above, we also got a key connection between linear algebra and probability theory:

Property 1. If X, Y are two random variables of zero mean ($E(X) = E(Y) = 0$), then

- (i) the covariance $Cov(XY) = E(XY)$ is the dot product of X and Y;
- (ii) The variance $Var(X) = E(X^2)$ is the squared length of X ($\sum_i x_i^2$).

Another property, that is also easy to prove, makes this connection even more clear:

Property 2. Vectors in R^n can be seen as random variables on the probability space $\{1, 2, ..., n\}$.

Thus, random variable X, Y and XY may be presented as

$$X = \begin{pmatrix} 1 & 2 & \dots n \\ x_1 & x_2 & \dots x_n \end{pmatrix};\ Y = \begin{pmatrix} 1 & 2 & \dots n \\ y_1 & y_2 & \dots y_n \end{pmatrix};\ XY = \begin{pmatrix} 1 & \dots n \\ x_1 y_1 & \dots x_n y_n \end{pmatrix},$$

i.e., as vectors that belong to R^n.

Correlation between two vectors is in fact their covariance, with the condition that it should be between -1 and 1.

We now need to switch to statistics to define the correlation between two vectors X and Y. Assume that values x_i and y_i are given. Then the correlation coefficient ρ is

$$\rho(X,Y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{(\sum_{i=1}^n (x_i - \bar{x}))} \cdot \sqrt{(\sum_{i=1}^n (y_i - \bar{y}))}},$$

where function that estimate expected value is mean $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$. Thus, correlation is the cosine of the angle between the two vectors. Positive correlation

means an acute angle, negative correlation means an obtuse angle. Uncorrelated, or independent, means orthogonal.

4.1.2 Singular value decomposition.

Suppose $A \in R^{m\times n}$ with $rank(A) = r$. The singular value decomposition (SVD) of A is to choose orthogonal basis $\{v_1, \ldots, v_r\}$ of row space of A, and choose orthogonal basis $\{u_1, \ldots, u_r\}$ of column space of A so that

$$Av_i = w_i u_i,\ w_1 \geq w_2 \geq \cdots \geq w_r > 0.$$

Values w_i are called *singular values*. In matrix form, the equations $Av_i = wu_i$ become $AV = UW$, where W is diagonal with singular values on diagonal. Therefore, since V is orthogonal ($VV^T = E$), after multiplying from the right the latest matrix equality by V^T, the matrix A can be written as

$$A = UWV^T,$$

Note that both $U \in R^{m\times r}$ and $V \in R^{n\times r}$ have orthonormal columns.

Interpretation of $A = UWV^T$.

Let us consider the relation $y = Ax$. It transforms the unit circle to an ellipse. By SVD we decompose the 'action' of A into the following three simple steps: rotation, scaling and rotation:

- rotate (or reflection) by V^T
- scale along the axes
- rotate by U.

Low rank approximation. Sometimes dimensions of A are too large. The basic idea for the low rank approximation of A is that

- $\bar{A}_1 = w_1 u_1 v_1^T$ gives the best rank 1 approximation to A and
- $\bar{A}_r = \sum_{j=1}^{r} w_j u_j v_j^T$ is the best approximation error to A.

where approximation error is $Err = A - \bar{A}_r$. The approximation is best in the sense that it reaches the minimum of the following objective function:

$$(\min) Err = \sum_i \sum_j |\varepsilon_{ij}|^2.$$

SVD theorem.

The singular value decomposition of $A \in R^{m \times n}, rank(A) = r$, has orthogonal matrices U and V so that

$$AV = UW \Leftrightarrow A = UWV^T = U_1 W_1 V_1^T,$$

where $A = [U_1\ U_2]\ W\ [V_1\ V_2]^T =$

$$\underbrace{\begin{bmatrix}\mathbf{u}_1 & \dots & \mathbf{u}_r\end{bmatrix}}_{\text{Col } A} \quad \underbrace{\begin{bmatrix}\mathbf{u}_{r+1} & \dots & \mathbf{u}_m\end{bmatrix}}_{\text{Nul } A^T} \begin{bmatrix} w_1 & \dots & 0 & \dots & 0 \\ \dots & & & & \\ 0 & \dots & w_r & \dots & 0 \\ 0 & \dots & 0 & \dots & 0 \\ \dots & & & & \\ 0 & \dots & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^T \\ \dots \\ \mathbf{v}_r^T \\ \mathbf{v}_{r+1}^T \\ \dots \\ \mathbf{v}_n^T \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}a\\b\\c\\d\end{matrix}}\right\} \text{Row } A \\ \left.\vphantom{\begin{matrix}a\\b\\c\end{matrix}}\right\} \text{Nul } A \end{matrix}$$

and $U_1 \in R^{m \times r}, W_1 \in R^{r \times r}, V_1 \in R^{n \times r}$ and $w_1 \geq w_2 \geq \dots \geq w_r > 0$.

Calculation of U, W and U. Calculation is based on finding the eigenvalues of $A^T A$. Since $rank(A) = rank(A^T A) = r$, $A^T A$ has r positive eigenvalues $w_1^2, \dots, w_r^2$. Therefore, singular values $w_1, \dots, w_r$ are found. Then, from the equation

$$A^T A v_i = w_i^2 v_i,$$

we can find orthonormal vectors $v_1, \dots, v_r$, and thus, V_1 is defined. Matrix U_1 can be obtained as $u_i = \frac{Av_i}{w_i}, i = 1, \dots, r$, i.e., $Av_i = w_i u_i$.

4.2 Dictionary learning and source localization

As mentioned earlier, our goal is to check if two approaches in analysing M/EEG signals do have some correlation: dictionary learning and source localization. Let us recall some notation that we use in our correlation Algorithm.

- G - leadfied matrix with dimensions $G \in R^{C \times U}$, where C is the number of channels and U the number of sources;

- $A_k, k = 1, \ldots, K$ - coefficients obtained in dictionary learning, $A_k \in R^{M \times K}$, where M is the number of repetitions (trials) and K the number of atoms;
- $A = [A_1 \ldots, A_K]$

A natural assumption is that an atom in the dictionary may correspond to a source in Source localization, i.e., we would like to check if $K = U$ based on series of experiments. To check if $U = K$, we develop an algorithm that we call `LeadfieldCorrelation`. In fact, we first assume that $U = K$ and then verify if there is strong correlation between columns of G and A_k. If the correlation exists, we can conclude that indeed, each active source in the brain corresponds to an atom.

Since matrices G and A_k do not have the same dimensions, we use singular value decomposition (SVD) approach. Namely, matrices A_k are decomposed: $A_k = U_k W_k V_k^T$. The j^{th} column of V_k that corresponds to the largest singular value w_j is taken to represent atom k. This is based on the property mentioned earlier, that the best rank one approximation to A_k is $\bar{A}_k = w_j u_j v_j^T$. Thus, the j^{th} column is placed as k^{th} column of the new matrix Y.

`LeadfieldCorrelation` pseudocode is presented bellow.

Algorithm 1: Correlation between leadfield matrix G and A

Function `LeadfieldCorrelation` (K, Z)
Read M, K and C (number of trials, atoms and channels resp.)
Read leadfield matrix G, $G \in R^{C \times K}$.
Let $k = 1$
while $k \leq K$ **do**
    Read matrix $A_k, A_k \in R^{M \times C}$;
    $U, W, V, r \longleftarrow$ `SVD`(A_k, M, C)
    Let $j_{max} = arg\max_j\{w_1, \ldots, w_r\}$
    Let $y(i, k) = v(i, j_{max}), i = 1, \ldots, C$
    Let $k = k + 1$
$Z \longleftarrow$ `Correlation`(C, K, G, Y)
return Z.

In `LeadfieldCorrelation` pseudo-code, all input data are read within it. Formal parameters (K, Z) present output, i.e., a correlation matrix of dimension $K \times K$. Subroutines that perform SVD and correlation are denoted with $\texttt{SVD}(A_k, M, C)$ and $\texttt{Correlation}(C, K, G, Y)$ respectively. Input variables of those procedures are listed as formal variables, why output variables are on the left hand side. For example, statement 6: $U, W, V, r \longleftarrow \texttt{SVD}(A_k, M, C)$ claims that input variables for SVD are A_k, M and C, while outputs are U, W, V and rank r.

In the Computational results section we will demonstrate and analyse results obtained by this algorithm on synthetic data.

We would also like to mention an important discovery. Let us recall the Source Localization model we used. We generated a gain matrix by using the information of the source distribution model obtained from a BEM head model, as explained in section 3.5. We synthetically created the EEG measurements (see 5.1) for a subset of the brain, assuming that only that part would be activated in a chosen time window. Using a number of channels influenced by a number of active sources, we created a synthetic gain and, solved the forward problem by calculating the EEG with these synthetic data. Then, by implementing the m-JADL onto that synthetic EEG, we obtained the Dictionary and the Coefficient matrix. The coefficient matrix represents the activation and influence of each atom onto the EEG measurement, as the gain matrix represents the same for the sources. As we showed that we can assume that $atoms = sources$, we can say that the Coefficient matrix might be used for solving the inverse problem. In other words, when we get such a correlation, we know which sources were surely active, thus we could inspect their location, as follows. We need to take into account the full gain matrix, as it happens in the source localization situation, to discover the location of active sources from the whole source configuration. As we used only a subset of the gain matrix for creating the synthetic EEG, we obtained a reduced coefficient matrix, representing only the active sources. By correlating this coefficient matrix with the full gain matrix, we might discover that the correlation reveals those columns/sources of the full gain which are the most influential in the EEG. This means that we will find active sources by correlating each column of the coefficient matrix with the full

leadfield, thus solving the inverse problem. The same could be applied for real data, meaning we already have the EEG and Gain as input, we first execute the m-JADL and then correlate the coefficients with the used leadfield and find the locations of active sources. Note that this is only in theory, as many details and practical issues were omitted. Although, some experiments were done to prove this hypothesis and has shown some promising results. Unfortunately for now we wont present them in this paper, due to the lack of time.

4.3 Correlation among trials

Another challenging question is if there is a correlation among different trials in measuring EEG signals from different channels? The simplest way to do that is without using SVD, i.e., we find correlation between any two row vectors of A_k, $k = 1, \ldots, K$. The final correlation matrix Z has dimension $M \times M$. Algorithm has the following steps.

Algorithm 2: Correlation between trials in A

Function `TrialCorrelation` (M, Z)

1 **Read** M, K and C (number of trials, atoms and channels resp.)
2 **Let** $k = 1$
3 **while** $k \leq K$ **do**
4 **Read** matrix $A_k, A_k \in R^{M \times C}$;
5 $Z_k \longleftarrow$ `Correlation`(M, C, A_k, A_k)
6 **Let** $k = k + 1$
7 **return** Z_k.

Again, results with `TrialCorrelation` (M, Z) will be demonstrated in Computational results section.

Chapter 4

This chapter contains computational results obtained by algorithms from the previous chapter. We first give brief description how synthetic (random) data-sets are generated. They are in fact taken from [22]. Then we analyse the correlation questions on synthetic (with and without noise) data sets.

5.1 Input data

In this sections we explain how the input data for testing different Dictionary learning techniques are obtained.

Synthetic Data. Synthetic data used in our work are those already generated in [22]. For the sake of completeness, the way how they are obtained will be briefly explained.

The generation of the synthetic data are based on the following observations. N input signals in N seconds may be obtained by generating just N amplitudes of N peaks of the signal. Then a learned atom could be presented as one of the peaks; the latencies would be calculated as the difference of the time point where the peak occurred and the time point of the learned atom and the coefficients would be the differences among the amplitude of the peaks and amplitude of learned atom. However, the real problem is much more complex than this simplified example: the wave-shapes we are searching for (the peaks in the previous example) are spread in different signals over the trials and channels; this more than one wave-shape may exist in the signal.

To check the quality of the dictionary learning method, we also need to generate some initial data.

(i) *Choose parameters* N, K, M and C that denote number of time intervals (seconds), atoms, trials and channels respectively;

(i) *Generate a dictionary* $D \in R^{N \times K}$ of K atoms. For that purposes usually are used normalized vectors (atoms) of various types, as spike, Gaussian peak or oscillatory;

(ii) *Generate a set of coefficients* at random for the multidimensional data set $\bar{a}_{ijc} \in R^{K \times M \times C}, i = 1, .., K; j = 1, .., M; c = 1, .., C;$

(iii) *Generate a set of jitters* or random latencies $\bar{\delta}_{ijc} \in Z^{K\times M\times C}, i = 1, .., K; j = 1, .., M; c = 1, .., C;$

(iv) *Generate synthetic sources* using the following formula

$$S_c^{(j)} = \sum_{i=1}^{K} \bar{a}_{ijc}\ \bar{\delta}_{ijc}\ d_i^{(j)}, \forall j = 1, .., M,\ \forall c = 1, .., C \tag{22}$$

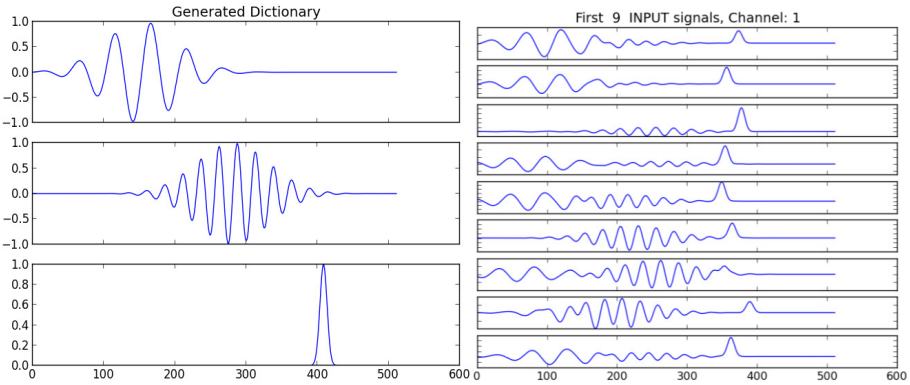

Figure 7: An example of a synthetic normalized dictionary and some generated signals. A dictionary with three atoms (left). The dictionary in this figure is plotted row wise for visualization reason. The first nine generated signals from the first channel (right), from [22]

In plain words, we activated the sources by multiplying them with the previously generated atoms.

Leadfield and synthetic data. The connection between a dictionary learning model and the forward and inverse models

$$M = G \cdot S$$

is discussed in & 4.2. We also need to be able to get the location of sources from the synthetic data set. For those purposes the lead field matrix G may be seen as a linear combination of synthetically generated atoms. However, the data used for getting G and S for solving the forward problem are taken from real data, based on real EEG measurements, as in [12].

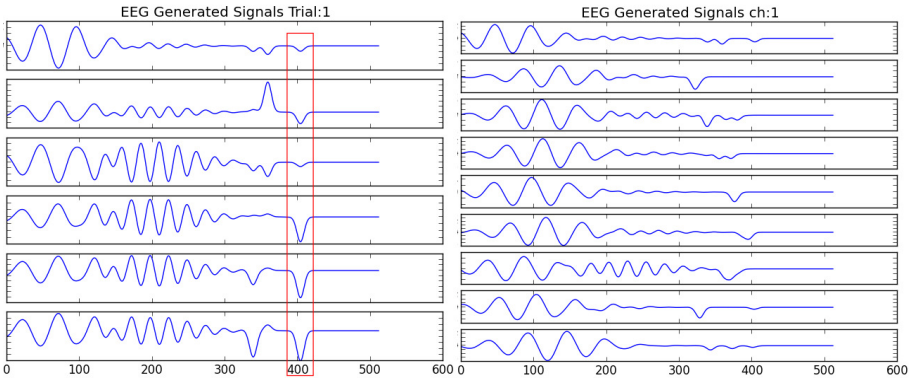

Figure 8: An example of the generated EEG measurements. The measurements corresponding to different channels for a single trial appear "synchronously" (left). Notice that the negative peaks occur at the same time point (around sample 400). The measurements that corresponds to different trials for a single channel have different jitters, from [22]

Two different ways of using those real data, together with the synthetic data already generated are explored [22]:

- A *simple leadfield data set*: the number of sources U is given; each group contains a single source from the source configuration; each group receives a different pattern of activity;
- A *complex leadfield data set*: the number of sources U together with the number of their closest neighbors V (with respect to the source configuration) are given; each group receives the different pattern of activities; each of the selected sources within a group receives 'jittered' version of the group activity.

It is easy to see that the simple leadfield data set is a special case of the complex one. Thus, in [22], the synthetic multi-dimensional EEG measurement that corresponds to a single trial that takes into account source localization problem, may be presented as

$$M = G\bar{S}\bar{a}^T D^T, \tag{23}$$

where elements of $\bar{S}$ and $\bar{a}^T$ belong to the set $\{0, 1\}$. The purpose of vector $\bar{S} \in R^U$ is to select the desired source columns of the gain matrix G. On

the other hand, the 0-1 vector $\bar{a}$ selects corresponding rows of the extended dictionary. Dictionary matrix D is an extended dictionary (previously denoted as D^S, see (20)). Therefore, solving the equation (23) (forward problem) for each trial, gives the multi-dimensional measurement matrix $M \in R^{N \times M \times C}$. In the figures bellow, we can see an example of 3 sources being activated.

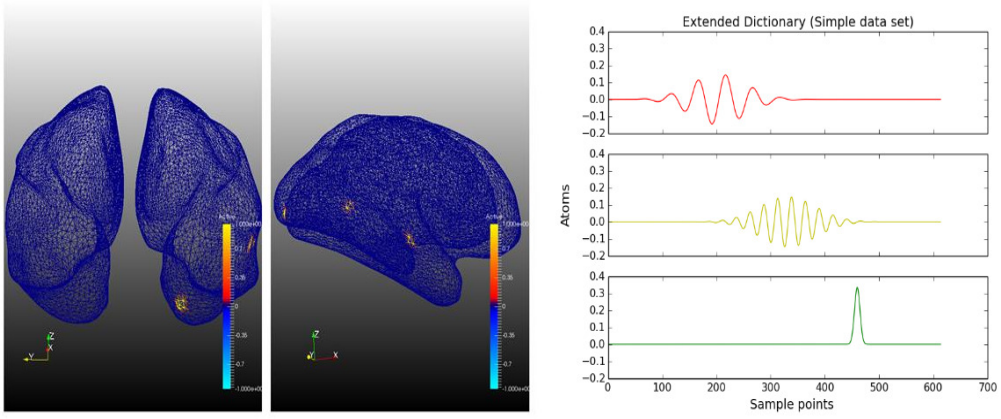

Figure 9: Left: an example of 3 selected source groups to be active, with the frontal and side view. Created with Vtk and visualized with Paraview. Right, from [22]: atoms used to activate these sources.

Adding noise into the synthetic data. As it was mentioned several times earlier, M/EEG data measured by electrodes are very noisy. That fact should be taken into account during the generation of realistic synthetic data. Noise is usually generated using random variable with Gauss (or normal) distribution. Noise is obtained by using a white additive Gaussian distribution. It can be introduced in two ways:

(i) *Noise at the channel level.* Its source are electronic devices. Noise can be generated for all the channels or to just some of selected ones.

(ii) *Noise at the source level.* Its source is psychological. It can appear by persons who are tested after some period of time. Thus, the random noise can be added when atoms are generated in solving the forward problem (23).

5.2 Computational results

Algorithms from chapter 3 are coded in Fujicu Fortran 95 and run on personal computer with 250 MhZ.

5.2.1 Algorithm 1.

We start with correlation between the leadfield matrix and dictionary learning coefficients, as explained in Algorithm 1.

Example 1. Let us first consider an example with synthetic data set without noise that has the following parameter values: $N = 3000, M = 300, K = U = 3$ and $C = 6$. First, the dimension of time ($t = 1, \ldots, N$) is removed from consideration by finding the average values over t, for each trial $j = 1, \ldots, 300$, and for each channel $c = 1, \ldots, 6$. At the input of Algorithm 1 from chapter 3.2, we in fact have matrices $A_k \in R^{300\times 6}$, for $k = 1, 2, 3$. From obvious reasons we do not present here A_k matrices. However, the generated (synthetic) 6×3 leadfield matrix used is

$$G = \begin{bmatrix} 26.10494941 & -2.91802128 & -15.17124887 \\ -12.42936481 & -9.95802063 & 11.22054694 \\ 4.7740596 & 27.7527064 & -10.96806411 \\ -11.62695673 & -22.10600332 & -6.03912566 \\ -10.21628105 & -24.94081715 & -15.1523816 \\ -10.03478176 & -20.35329612 & -17.30765927 \end{bmatrix}$$

The following values are obtained from Algorithm 1. First of all, for all k, the $rank(A_k) = 3$. The diagonal 6×6 matrix $W^{(1)}$, that correspond to the first atom (channel) looks as follows.

$$W^{(1)} = \begin{bmatrix} 482.1786 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5.4515 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5249 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Therefore, there are 3 singular values and diagonal matrix $W_1^{(1)}$ becomes

$$W_1^{(1)} = \begin{bmatrix} 482.1786 & 0 & 0 \\ 0 & 5.4515 & 0 \\ 0 & 0 & 0.5249 \end{bmatrix}$$

The largest singular value 482.1786 is in the first column ($j_{max} = 1$). According to Algorithm 1 (and based on SVD properties explained earlier in chapter 3), we got

$$y(1, j) = 482.1786 \cdot v_1^T = (2.7833, 9.8729, -27.3949, 21.7361, 24.4749, 19.9446)^T.$$

The ranks of A_2 and A_3 matrices are also 3, and we get

$$W_1^{(2)} = \begin{bmatrix} 350.0590 & 0 & 0 \\ 0 & 0.4295 & 0 \\ 0 & 0 & 0.1625 \end{bmatrix}$$

and

$$W_1^{(3)} = \begin{bmatrix} 327.7443 & 0 & 0 \\ 0 & 3.7919 & 0 \\ 0 & 0 & 0.1712 \end{bmatrix}.$$

In the same way as we got the first column of mtrix Y, we get its second and third columns. The final matrix Y that should be correlated with the given leadfield matrix G is then

$$Y = \begin{bmatrix} 2.7833 & 26.3891 & -15.3998 \\ 9.8729 & -12.5658 & 11.4067 \\ -27.3949 & 4.8289 & -11.1745 \\ 21.7361 & -11.7553 & -6.1135 \\ 24.4749 & -10.3293 & -15.3682 \\ 19.9446 & -10.1453 & -17.5633 \end{bmatrix}.$$

We now need to find correlation between any two columns of G and Y. The final correlation matrix $Corr$ has obviously dimensions 3×3.

$$Corr(G, Y) = \begin{bmatrix} -0.5179 & 1.0000 & -0.3801 \\ -1.0000 & 0.5167 & 0.0212 \\ -0.0199 & -0.3794 & 1.0000 \end{bmatrix}.$$

It appears that for each column of ledfield matrix G there exists column of Y such that these two columns correlate (the corresponding correlation coefficients are equal to 1 or -1). In this example, it is confirmed hypothesis that the number of sources correspond to the number of atoms, $U = K$. Figure 11 presents the correlation of columns of G and Y.

Example 2. We will use the same set of synthetic data and the same gain matrix G as in Example 1, but this time we will add the noise at the channel level. The following result is obtained by Algorithm 1. All three matrices A_1, A_2 and A_3 have the full rank 6. To save the space, we do not present all details (hey can be seen at the web page of the candidate). We just present the final matrices Y and $Corr(G, Y)$:

$$Y = \begin{bmatrix} 2.7990 & 26.3004 & -15.4082 \\ 9.8404 & -12.5671 & 11.4725 \\ -27.3895 & 4.7738 & -11.1806 \\ 21.6683 & -11.7749 & -6.0263 \\ 24.4973 & -10.3090 & -15.3844 \\ 19.9437 & -10.2123 & -17.6349 \end{bmatrix}.$$

We can conclude that noise did not influence much columns in Y. Indeed, coefficients y_{ij} in this example with noise slightly differ from those obtained in the previous noiseless example.

$$Corr(G, Y) = \begin{bmatrix} -0.5173 & 1.0000 & -0.3799 \\ -1.0000 & 0.5165 & 0.0207 \\ -0.0209 & -0.3789 & 1.0000 \end{bmatrix}.$$

The same conclusion may be drawn after comparing correlation matrices in last two examples. Indeed, the values of 1 and -1 remain at the same position.

5.2.2 Algorithm 2.

Here we present results obtained with Algorithm 2 from previous section. It directly correlates all rows of each matrix A_k.

Example 3. We use again the same instance from Example 1. Since we directly correlate rows of A_k, we do not need leadfield matrix G. The final correlation

matrix $Corr(A_k, A_k)$, for all $k = 1, 2, 3$ has dimension 300×300. It appeared that most values were 1 or -1. This is specially the case for the first source (atom), i.e., for $k = 1$. Correlation matrices for two extreme cases are presented in figures 10 and 12. As a plotting tool, we used gnuplot.

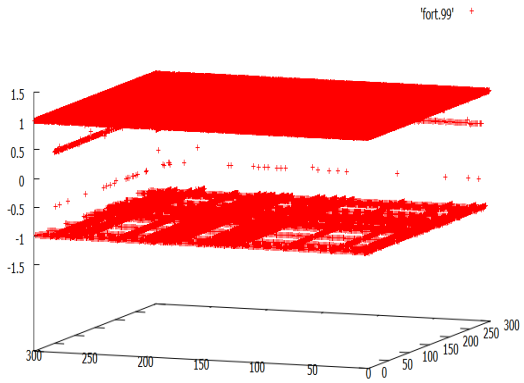

Figure 10: Direct correlation of 300 trials on example with $C = 6$ channels, noiseless data and $k = 1$.

In Figure 10 the correlation coefficients among 300 vectors is presented where we got the strongest correlation. It is mostly 1 or -1. Indeed, in less than 1 % cases coefficients were different than 1 or -1.

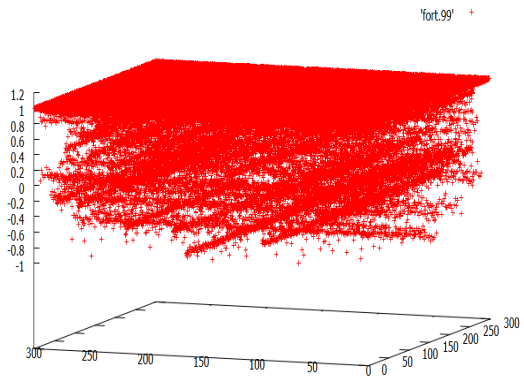

Figure 11: Direct correlation of 300 trials on example with $C = 6$ channels, noisy data and $k = 3$.

The weakest correlation is obtained in the noisy data case and with the third source (atom). It appears that the negative correlation almost never occurred.

Chapter 5

In this final chapter we outline possible contribution of this Intership, some conclusions, and then some possible suggestions for the future work.

6.1 Contributions

The following contributions may be attributed to this Intership:

- An original review on the Dictionary learning topic is given. It is based on reading and analysing large literature from different fields, since the topic is already multi-disciplinary.
- New interpretation of some steps of dictionary learning and JADL algorithms is provided. It could lead to improvement of the existing methods. For example, the new discrete optimization formulation of sparse coding step of JADL is proposed. In the original method that problem is solved by local search heuristic. However, better heuristic, based on some meta-heuristic paradigms (e.g. Genetic algorithm) can be applied.
- It is clearly shown the connection between Dictionary learning methods and Alternate heuristics, commonly used technique in Optimization and Operations research. The connection between the two research fields is always welcome, bringing benefits to both.
- This work also contains new algorithms to compare and correlate results obtained by different mathematical models for EEG measurements. For that purposes, some classical numerical analysis and probability theory methods are used and coded: Singular value decomposition method and Correlation.
- After the information gained from the Correlation and SVD, the connection between Dictionary Learning and the Source Localization problems is quite evident. It is suggested how the complexity of the source localization problem could be reduced by introducing the solution of the Dictionary Learning as a constraint.

6.2 Conclusions

The first chapter aims to introduce the reader into the topic. The second chapter provides an original view to the topic, trying to find the connection between the dictionary learning approach and the well known heuristics from Optimization and Numerical analysis areas. Namely, Alternate heuristic is implemented in a large number of numerical problems such as clustering (k-means algorithm), location (Cooper's method), Zaidel method for solving simultaneous system, etc. At the same time, it is shown that Alternate heuristic is a basis for JADL method as well. In third chapter we propose a new way of comparing two different approaches from the field, Dictionary Learning and Source Localizaion. It appears that the hypothesis claiming that each atom corresponds to one source is confirmed for the synthetic data with and without noise. It also appears that trials are strongly correlated among each other. Therefore, there is no need to perform many of them and increase unnecessary the size of the problem.

Future work may include the following ideas:

- Develop a new heuristic for finding both dictionary and coefficients that will relay on some metaheuristic (as Evolutionary algorithms, Iterated Local search etc). Classical alternate heuristic may be used as a part of the final method, i.e., as a local search (as Memetic algorithm that uses local search heuristic as mutation operator).

- Improve JADL by solving both steps with new global optimization solution techniques.

- There are a lot of space to analyse huge amount of EEG data obtained. Some clustering, data mining or Big data techniques could obviously be applied as well.

- Reducing the complexity of the Source Localization problem with the coefficient matrix of Dictionary Learning

References

[1] Audet C, Hansen P, Jaumard B, Savard G (1999). A symmetrical linear maxmin approach to disjoint bilinear programming. *Mathematical Programming A* 85, 573–592.

[2] Bénar CG, Papadopoulo T, Torrésani B, Clerc M (2009). Consensus matching pursuit for multi-trial EEG signals. *J of Neuroscience Methods* 180 (1), 161 – 170.

[3] Candès EJ, Demanet L (2004). The Curvelet Representation of Wave Propagators is Optimally Sparse (2004), *Comm. Pure Appl. Math.* 58, 1472-1528.

[4] Candès EJ and Romberg J (2006). Sparsity and incoherence in compressive sampling. Inverse Problems, 23 969-985.

[5] Cooper L. (1964) Heuristic methods for location-allocation problems. *SIAM Rev.* 6, 37–53.

[6] Dawson GD (1954). A summation technique for the detection of small evoked potentials. *Electroencephalography and clinical neurophysiology* 6, 65–84.

[7] Efron B, Hastie T, Johnstone L and Tibshirani R (2004). Least angle regression. *The Annals of statistics*, 32(2):407–499.

[8] Elad M and Aharon M (2006). Image denoising via sparse and redundant representations over learned dictionaries. *Image Processing, IEEE Transactions* 15(12), 3736–3745.

[9] Grech R, Cassar T, Muscat J, Camilleri K, Fabri S, Zervakis M, Xanthopoulos P, Sakkalis V, and Vanrumste B (2008). Review on solving the inverse problem in EEG source analysis *J Neuroeng Rehabil* doi: 10.1186/1743-0003-5-25

[10] Hallez H, Vanrumste B, Grech R, Muscat J, Clercq W, Vergult A, D'Asseler Y, Camilleri K, Fabri S, Van Huffel S and Lemahieu I (2007). Review on

solving the forward problem in EEG source analysis. *Journal of NeuroEngineering and Rehabilitation*, doi:10.1186/1743-0003-4-46

[11] Haverly CA (1978). Studies of the behavior of recursion for the pooling problem, *ACM SIGMAP Bulletin* 25, 19–28.

[12] Henson R, Wakeman D, Phillips C and Tilman L (2014). Effective connectivity between ofa and ifa during face perception.

[13] Hitziger S, Clerc M, Gramford A, Saillet S, Bénar C and Papadopulo T (2013). Jitter-adaptive dictionary learning application to multi-trial neuroelectric signals. *arXiv preprint arXiv:1301.3611*

[14] Hyvärinen A and Oja E (1997). A fast fixed-point algorithm for independent component analysis. *Neural Comput* 9: 1483–1492.

[15] J. Lötjönen (2003): Construction of patient-specific surface models from MR images: application to bio-electromagnetism, *Computer Methods and Programs in Biomedicine*; 72:167-178

[16] Knuth KH, Shah AS, Truccolo WA, Ding M, Bressler SL and Schroeder CE (2006). Differentially variable component analysis: Identifying multiple evoked components using trial-to-trial variability. *Journal of Neurophysiology* 95 (5): 3257–3276.

[17] MacQueen JB (1967). Some methods for classification and analysis of multivariate observations, *Proceeding of the 5th Berkeley Symposium on Mathematical Statistics and Probability* 2, 281-297.

[18] Mak JN, Wolpaw JR (2009). Clinical Applications of Brain-Computer Interfaces: Current State and Future Prospects. *IEEE Rev Biomed Eng* 2, 187-199.

[19] Mairal J, Bach F, Ponce J and Sapiro G (2010). Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.* 11:19–60. ISSN 1532-4435.

[20] Mallat S and Zhang Z (1993). Matching pursuits with time-frequency dictionaries. Signal Processing, *IEEE Transactions on,??* 41(12): 3397–3415.

[21] Mallat S (1999). A wavelet tour of signal processing. Academic press, London.

[22] Papageorgakis C, Papadopoulo T (2014). *Dictionary learning for multidimensional data*, Intership report, Inria, University of Nice Sophia Antipolis.

[23] Plonsey R, Heppner DB (1967). Considerations of quasistationarity in electrophysiological systems. *Bulletin of Mathematical Biophysics*, 29(4):657-664.

[24] S ornmo L and Laguna P (2005). *Bioelectrical signal processing in cardiac and neurological applications* (chapter 2), Academic Press, London.

[25] Schimpf P, Ramon C and Haueisen J (2002). Dipole Models for the EEG and MEG, *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. 49, NO. 5,*

[26] Tibshirani R (1996). Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B.* 58 (1) 267-288.

[27] Truccolo Z, Knuth KH, Shah A, Bressler SL, Schroeder CE, and Ding M (2003). Estimation of single-trial multicomponent erps: Differentially variable component analysis (dvca). *Biological cybernetics* 89 (6): 426–438.

[28] Vidal JJ (1973). Toward Direct Brain-Computer Communication. *Annual Review of Biophysics and Bioengineering* 2, 157–80.

[29] Wolpaw JW, Birbaumer N, McFarland DJ, Pfurtscheller G, Vaughan TM (2002). Brain–computer interfaces for communication and control. *Clinical Neurophysiology* 113, 767–791.

[30] Woody C (1967). Characterization of an adaptive filter for the analysis of variable latency neuroelectric signals. *Medical and Biological Engineering and Computing* 5(6): 539–554.