# Adaptive machine learning methods for event related potential-based brain computer interfaces

Nathalie Gayraud

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

# THÈSE DE DOCTORAT

# Méthodes adaptatives d'apprentissage pour des interfaces cerveau-ordinateur basées sur les potentiels évoqués


**Nathalie Thérèse Hélène GAYRAUD**

INRIA Sophia Antipolis – Méditerranée, Équipe-Projet Athéna


**Présentée en vue de l'obtention du grade de docteur en** STIC **d'**Université Côte d'Azur

**Dirigée par** : Maureen Clerc

**Soutenue le**: 11/12/2018

**Devant le jury, composé de** :
Maureen Clerc, Inria Sophia-Antipolis
Marco Congedo, GIPSA Lab, Grenoble
Nicolas Courty, Université Bretagne Sud
Moritz Grosse-Wentrup, University of Munich
Alain Rakotomamonjy, Université de Rouen
Michael Tangermann, University of Freiburg

Athena

Inria

# Méthodes adaptatives d'apprentissage pour des interfaces cerveau-ordinateur basées sur les potentiels évoqués

# Adaptive Machine Learning Methods for Event Related Potential-Based Brain Computer Interfaces

Jury :

Rapporteurs :
Marco Congedo, GIPSA Lab, Grenoble
Michael Tangermann, University of Freiburg

Examinateurs :
Nicolas Courty, Université Bretagne Sud
Moritz Grosse-Wentrup, University of Munich
Alain Rakotomamonjy, Université de Rouen

# RÉSUMÉ

Les interfaces cerveau machine (BCI pour Brain Computer Interfaces) non invasives permettent à leur utilisateur de contrôler une machine par la pensée. Ce dernier doit porter un dispositif d'acquisition de signaux électroencéphalographiques (EEG), lesquels sont par la suite traités et transformés en commandes. Cependant, les signaux EEG sont dotés d'un rapport signal sur bruit très faible ; à ceci s'ajoute l'importante variabilité que l'on trouve tant à travers les sessions d'utilisation qu'à travers les utilisateurs. Par conséquent, la calibration du BCI, pendant laquelle l'utilisateur est amené à effectuer une tâche prédéfinie, doit souvent précéder son utilisation. Le sujet de cette thèse est l'étude des sources de cette variabilité, dans le but d'explorer, concevoir, et implémenter des méthodes d'autocalibration. Nous nous intéressons en particulier aux interfaces cerveau machine qui utilisent des potentiels évoqués comme marqueur neurophysiologique (ERP-based BCI), que nous introduisons dans la première partie.

La deuxième partie de cette thèse porte sur l'analyse des sources de variabilité que l'on rencontre dans ce type d'interface cerveau machine. Nous effectuons une étude bibliographique de la variabilité des potentiels évoqués, en nous intéressant particulièrement au potentiel tardif connu sous le nom de P300. Nous réalisons aussi une analyse de la variabilité du signal EEG sur deux bases de données qui proviennent d'expériences de BCI avec potentiel évoqué. Après avoir évalué les sources de variabilité de manière quantitative, nous nous penchons sur les méthodes adaptatives d'apprentissage automatique, notamment, les méthodes d'apprentissage par transfert. Ces méthodes permettent aux algorithmes de classification de généraliser malgré la variabilité, et donc de ne pas avoir besoin d'être calibrés avant chaque utilisation du BCI. Nous analysons sur trois méthodes en particulier pour décrire à quel type de variabilité elles sont le mieux adaptées : la géométrie riemannienne, le transport optimal, et l'apprentissage ensembliste. Puis

nous proposons un modèle du signal EEG généré pendant l'utilisation d'un BCI qui tient compte de la variabilité. Les paramètres résultants de nos analyses nous permettent de calibrer ce modèle et à simuler une base de données, qui nous sert à évaluer la performance de ces méthodes d'apprentissage par transfert. Les résultats de cette analyse démontrent que ces méthodes sont adaptées à certains types de variabilité ; cependant, aucune de ces méthodes ne s'affranchit de toutes les sources de variabilités présentes dans les données EEG.

La troisième partie de cette thèse porte sur l'application de ces méthodes à des données expérimentales et à la conception de méthodes dérivées de celles-ci. Nous proposons une méthode de classification basée sur le transport optimal dont nous évaluons la performance. Ensuite, nous introduisons un marqueur de séparabilité qui nous permet d'évaluer un ensembles de vecteurs de caractéristiques, particulièrement dans le cadre de la géométrie riemanniene. Ce dernier nous permet de concevoir une méthode qui réunit géométrie riemannienne, transport optimal et apprentissage ensembliste. Nos résultats témoignent que la combinaison de plusieurs méthodes d'apprentissage par transfert nous permet d'obtenir un classifieur qui s'affranchit des différentes sources de variabilité du signal EEG de manière efficace. Enfin, nous proposons une méthode de calibration non supervisée pour le cas particulier d'un BCI spécifique : le clavier virtuel P300 . La thèse se conclut par une discussion générale, ainsi que nos contributions additionnelles.

## Mots-clés

Interfaces cerveau machine, Apprentissage automatique, Traitement du signal

# ABSTRACT

Non-invasive Brain Computer Interfaces (BCIs) allow a user to control a machine using only their brain activity. The BCI system acquires electroencephalographic (EEG) signals using an EEG acquisition device. The signals are subsequently processed and transformed into commands. However, EEG is characterized by a low signal-to-noise ratio and an important variability both across sessions and across users. Typically, the BCI system is calibrated before each use, in a process during which the user has to perform a predefined task. This thesis studies of the sources of this variability, with the aim of exploring, designing, and implementing zero-calibration methods. In particular, we are interested in Event Related Potential-based BCI (ERP-based BCI), which we introduce in the first part.

The second part of this thesis deals with the analysis of the sources of variability encountered in ERP-based BCI. We review the variability of the event related potentials, focusing mostly on a late component known as the P300. We also perform an analysis on two databases containing EEG signals that were generated during ERP-based BCI experiments. This allows us to quantify the sources of EEG signal variability. Our solution to tackle this variability is to focus on adaptive machine learning methods, such as transfer learning , which allow classification algorithms to generalize. We focus on three methods in particular and describe which type of variability they are the most suited for: Riemannian geometry, optimal transport, and ensemble learning. Then, we propose a model of the EEG signal generated during the use of an ERP-based BCI that takes variability into account. The parameters resulting from our analyses allow us to calibrate this model in a set of simulations, which we use to evaluate the performance of the aforementioned transfer learning methods. The results of this analysis demonstrate that these methods generalize under certain types of variability; however, none of these methods can cope with all the sources of variability that are present in the EEG

signal.

The third part of this thesis deals with the application of these methods to experimental data and the design of methods derived from their combination. We first propose a classification method based on optimal transport which we evaluate in terms of classification performance. Then, we introduce a separability marker that can be applied to evaluate training sets, especially under the framework of Riemannian geometry. We use the separability marker to design a method that combines Riemannian geometry, optimal transport and ensemble learning. Our results demonstrate that the combination of several transfer learning methods produces a classifier that efficiently handles multiple sources of EEG signal variability. Finally, we propose an unsupervised calibration method for a specific BCI: the P300 Speller. The thesis concludes with a general discussion, as well as our other contributions.

## Keywords

Brain Computer Interfaces, Machine Learning, Signal Processing

# ACKNOWLEDGEMENTS

To my grandfather Giannis
Στον παππού μου το Γιάννη

Three years and three months have passed, and the journey came to an end. And what a journey it was. Full of all kinds of lessons, enriching and wholesome, but also hard and full of storms and trials. A true odyssey. The time has come for me to thank all those who accompanied me in this journey and helped me complete it, one way or another. Forgive me in advance if I have forgotten someone, I have so many things to be grateful for!

Maureen, I would like to thank you first, for giving me this amazing opportunity. Without you none of this would've been possible. You stood by me all these years, providing wise council. I would like to especially thank you for your immeasurable support during the time I was writing my thesis. You made easier one of the hardest moments in a PhD student's life and I cannot thank you enough for that. Rachid, I cannot thank you enough either, for you have put together a team of remarkable people in every way, something only a remarkable person can do. Your warmth and kindess matched by Efi's made so much easier for me to not feel homesick.

Guillermo, we started this together and we finished it together. No words can accurately describe my gratitude to you, and anyway we've said them all for the past four years. I enjoyed a truly perfect friendship with you, which led to a beautiful collaboration too, so I have been doubly blessed. Let me add simply this: thank you for being my christian moral compass all this time ;) Marco, I am so glad I met you my friend, so glad for all those beautiful memories of emptying wine bottles together while philosophising about life, or partying with you and Guillermo. Without you and Guillermo, my life would've been bland, deprived of that beautiful spirit of yours.

Of course, I was able to go through all that because I had the best officemates EVER. Lavinia, your friendship has been so precious to me, we helped each other through hard moments, but your kindness gave me the will to overcome all of them. Along with Guille, Marco, Luna, Samira and Fatmanur, it is not at all an exaggeration to say that I have been blessed to have met you all and spent these moments with you. One of the things that made me the happiest was finding a group of friends to share amazing times with. It also makes me sad now because we are about to take different trajectories in life. But that is life, right? So, thank you for all these moments, all that you did for me, the small and the big things: the gifts, the staying with me at the hospital, the apéros, the dinners, the trips, the friendship.

Federica, thank you for the amazing collaboration, it was so fulfilling and interesting. I hope it will be the first of many. I will definitely miss your sweet sweet person and will always remember the fun we've had in and out of the office, with you, Isa and Antonia. Isa, Antonia, you truly are amazing people, I love you girls! Kostya, you have been the best colleague ever. I look forward to many more Dota games with you, Samuel, Marco and Guille, so that I may keep my sanity as I did during the last days of my PhD. Last days that would've been much much harder if I hadn't met my gaming partner and wonderful friend Chris in an unlikely turn of events. Thank you so much for all that motivation you gave me when I felt like giving up.

Athenians, past and present, visiting and permanent, you are all amazing people and skilled scientists. Let me mention those I haven't yet, and thank you all for every moment: Theo, Ragini, Demian, Marco, Rutger, Brahim, Kai, Nathanael, Amandine, Mauro, Patryk, Abib, Matteo, Ivana, Sara, Romain, It's been an honor and a pleasure. Jelena, I mention you especially because you were my first true friend when I came back to France. Thank you for putting up with me all this years and for relentlessly motivating me. I'd like to also thank each and every person implicated in the BCI-Lift project, to which I owe this PhD and numerous collaborations. Sebastien, j'ai vraiment apprecié nos collaborations, et j'espère bien qu'il en aura encore dans le futur.

I would like to especially mention a beautiful soul that I met and became friends with during these three years. Agnes, nos chemins vont se séparer, mais je n'oublierais jamais ta gentillesse et je prie pour que tu en sois récompensée,

parce que tu ne mérites que le meilleur. Δήμητρα, Έλενα, Χρήστο, Χριστίνα, Κωνσταντίνε και λοιποί έλληνες της Côte d'Azur, σας ευχαριστώ πολύ για όλα όσα κάνατε για μένα, και πάνω από όλα για τη φιλία σας. Περάσαμε πολλά μαζί αλλά τα καταφέραμε και θα συνεχίσουμε έτσι. Je remercie aussi Eli, Philippe, Fanny, Jean Paul, les pasteurs Thibaud et Patrick, et tous mes frères et soeurs en Christ de l'église d'Antibes και της Ελλάδας. Vos prières m'ont amené jusqu'ici et je prie que Dieu puisse vous bénir tous. Un grand merci pour tout ce que vous avez fait pour moi, tant spirituellement que par vos actes.

Κορίνα, Φωτεινή, Μαρία, Έφη, είστε η οικογένεια που διάλεξα και ξέρω ότι δε χρειάζεται να σας πω πολλά. (Σαν) Αδερφές μου όλα αυτά τα χρόνια, είμαστε πλάι η μία στην άλλη (ακόμα και νοητά, δεν έχει σημασία) στα δύσκολα και στα εύκολα μαζί, κάτι που δε θα ήθελα ποτέ να τελειώσει, μία φιλία που δε θέλω να την καταστρέψει ο χρόνος και η απόσταση και θα κάνω τα πάντα για αυτό. Σας ευχαριστώ που με αγαπάτε όπως είμαι και που με διορθώνεται όταν δεν είμαι αυτό που θα έπρεπε να είμαι. Μαρίνα μου, γλυκούλα μου, τι να πρωτοαναφέρω, τις συνεργασίες μας, το ότι ήμασταν μαζί όταν έμαθα τα νέα ότι θα έρθω Γαλλία, ή το ότι μετά από τόσα χρόνια η σχέση μας δεν έχει αλλάξει καθόλου··· Σάγαπώ πολύ και σε ευχαριστώ για όλα.

Τέλος, ένα μεγάλο σας αγαπώ και σας ευχαριστώ θα πω στην οικογένειά μου. D'abbord à mes parents, Pascal et Calypso, qui ont tant donné pour moi. Á mon frere Anthony qui est le meilleur frères du monde. Je suis tellement heureuse d'avoir pu accomplir tout cela, που σας έκανα περήφανους. Πως να ανταποδώσω την αγάπη σας· Το αφήνω στα χέρια του Θεού αυτό γιατί θα μου ήταν αδύνατο. A mon papi et ma mamie et toute ma famille en france. Στην γιαγιά μου, και στον παππού μου που τόσο θα χαιρόταν αν με έβλεπε, και που για το λόγο αυτό του αφιερώνω τη διατριβή αυτή. Σε όλη μου την οικογένεια στην ελλάδα.

To God, I give grace in the name of Jesus, who blessed me with everything.

Ευχαριστώ. Merci. Thank you.

# TABLE OF CONTENTS

# GENERAL INTRODUCTION

*Mankind has long been on a quest to understand the mind, source of human thought. This pursuit has led us into one of the most intricate organs of the human body: the brain. In ancient Greece, Hippocrates was the first to proclaim that the brain was the seat of intelligence, the source of our thoughts and sensations. Fast forward to the end of the 19th century, when Spanish anatomist Santiago Ramón y Cajal introduces the neuron to the scientific world. This discovery made him one of the pioneers of neuroscience and marked the beginnings of the neuroscience field. In 1924, for the first time, Hans Berger produces a human electroencephalographic recording. Forty nine years years later, Jacques Vidal publishes a study on direct brain-computer communication [Vidal, 1973].*
*The Brain Computer Interface field is born.*

## Context

Brain Computer Interfaces (BCIs) are conceived with an aim to provide an alternate means of communication to people with severe motor disabilities [Wolpaw et al., 2002]. A BCI system reads and deciphers electroencephalographic activity. Electroencephalography (EEG) measures the scalp electric potentials produced by electrical activity in neural cell assemblies [Baillet et al., 2001]. The discovery of significant correlations between spatiotemporal variations of the EEG signal and specific mental tasks has made it possible to use EEG to decipher a person's intentions [Wolpaw et al., 2002; Cabestaing and Derambure, 2016]. Neurophysiological markers such as event related synchronization/desynchronization (ERD/ERS) allow to identify imagined movements. Sensory evoked potentials (SEP) and event

related potentials (ERP) elicit distinct responses to stimuli, which we can extract and convert into commands.

In this thesis, we study a particular type of BCI, namely ERP-based BCI. ERP-based BCI are non-invasive, EEG-based, reactive BCI. Non-invasive, because the signal is acquired through sensors that are placed on the scalp. EEG-based, because these sensors record EEG activity. Reactive, because the user controls the BCI by choosing to pay attention (or to not pay attention) to a stimulus [Cabestaing and Derambure, 2016].

The stimuli of an ERP-based can be visual, auditory, or tactile. Therefore, the communication provided by ERP-based BCI does not depend on a particular sensory input. This makes them an attractive framework for people who suffer from serious motor disorders that lead to a locked-in syndrome, such as Amyotrophic Lateral Sclerosis (ALS). This was precisely the motivation of Farwell and Donchin, who in 1988 introduced the first ERP-based BCI application [Farwell and Donchin, 1988]. Today, ERP-based BCI are used for spelling [Blankertz et al., 2011; Guy et al., 2018], moving on-screen objects [Iturrate et al., 2015] and even gaming [Barachant and Congedo, 2014].

## Objective

ERP-based BCI systems operate by recording EEG activity, from which features are extracted and classified. Upon classification, the user receives feedback. A feedback example for a visual ERP-speller is the display of the selected letter on the screen. If the classification result is correct, the feedback will correspond to the user's intention. Therefore, the performance of the classifier is highly important for a BCI. However, a major drawback of EEG-based applications is the low Signal-to-Noise Ratio (SNR) of the EEG signal. As a result, advanced signal processing and machine learning techniques need to be employed to enhance the classification accuracy. Moreover, as the variability of the EEG signal is very high, BCI sessions usually include a preliminary system calibration. Calibration describes a process during which the user has to perform a specific task without feedback. This process can be lengthy, and is generally dull and tiresome for the user [Clerc et al., 2016]. The goal of this thesis is to propose adaptive machine learning methods which take explicitly into account the various types of EEG variability and the low SNR of the

signal.

## Contributions

In this thesis, we focus on the performance of various adaptive machine learning in terms of EEG signal variability. The contributions of our work are detailed below.

**A Model of EEG variability** We perform a detailed study of EEG signal variability. Our first contribution is a review of the existing literature on ERP variability, coupled with an analysis on two experimental datasets. We propose a model of the EEG signal which includes the various types of variability that arise from this study.

**Transfer Learning Methods Against Variability** Using our EEG signal model, we generate simulated EEG signals and use them to evaluate three transfer learning frameworks: Riemannian geometry; optimal transport; and ensemble learning. In particular, we propose to interpret how each one deals with variability and we compare them between each other. In the final chapters, we propose classification methods that combine these three frameworks.

**Optimal Transport Applied to ERP-based BCI** We introduce a transfer learning framework based on optimal transport theory. We propose two methods that make use of this framework in the classification pipeline. The first one acts on the feature extraction step while the second acts on the classification step.

**Separability in the Riemannian manifold of Symmetric Positive Definite Matrices** We provide a theoretical analysis of the Riemannian manifold of Symmetric Positive Definite matrices based on high dimensional geometry and statistics. This analysis leads to a marker of separability that can be applied to binary classification problems. We use this separability marker to assess the classification results in an ensemble classifier.

**Unsupervised Classification** We introduce an unsupervised classification method for a specific ERP-based application, the P300 speller. Our method takes into ac-

count the structure of ths particular BCI system and of the vectors in the feature space. We present preliminary results that provide a proof of concept of our method.

## Structure

The thesis is structured in the following way:

**Chapter 1** In the first chapter, we introduce ERP-based BCI. We provide a brief history and detail the BCI system and its components. Finally, we focus our attention on calibration and introduce our proposed solution, which is the use of adaptive machine learning methods.

**Chapter 2** The second chapter starts a review of the bibliography on ERP component variability. Then, we perform an analysis of the EEG variability of two experimental datasets. The first dataset includes EEG recordings from healthy users and the second contains EEG recordings from ALS patients.

**Chapter 3** We begin chapter 3 with a short review of transfer learning and its previous applications to ERP-based BCI. Then, we present three transfer learning frameworks. The first one is the Riemannian geometry framework, in which features are covariance matrices of the segmented EEG signal. The second one is based on optimal transport theory, whose aim is to compute an optimal transport plan that moves probability masses while minimizing a cost. The third is ensemble learning, which trains multiple classifiers and aggregates their decisions to provide a single classification result.

**Chapter 4** In the fourth chapter, we introduce our EEG signal model, which takes into account our experimental analysis of variability (chapter 2). We use this model to simulate a BCI experiment and generate EEG signals. These simulated signals are used to study how each transfer learning method in chapter 3 deals with each type of variability analyzed in chapter 2. We present the results of our study and discuss them.

**Chapter 5** Chapter 5 provides a detailed description of the optimal transport framework. We introduce the problem and provide the theoretical background of discrete regularized optimal transport. We propose two methods that use optimal transport. The first one applies it in the feature space as a domain adaptation tool. The second one applies optimal transport to derive a new classification method. Then, we combine these methods to ensemble learning. We perform experiments on experimental data to assess the performance of the two methods and of the ensemble learning framework. We conclude this chapter with a discussion.

**Chapter 6** In the sixth chapter, we conduct a geometrical analysis of the Riemannian space of symmetric positive definite matrices. First, we recall some known geometrical and statistical properties on high-dimensional spaces and provide the theoretical framework of our analysis. Then, we introduce the Separabiliy Marker, a marker of class separability for binary classification problems under the Riemannian geometry framework. Then, we propose an ensemble learning method that make use of the Separability Marker and combines the transfer learning methods of chapter 3. We discuss our results and conclude this chapter.

**Chapter 7** In chapter 7, we propose a proof of concept for a novel unsupervised method applied to the P300-Speller. We introduce our method and provide preliminary results that prove its feasibility. We discuss those results and conclude the chapter.

Finally, we conclude this thesis with a general discussion on our results and future perspectives.

# NOTATIONS

| | |
|---|---|
| $\alpha$ | Parameter of the pink noise process . . . . . 26 |
| $\bar{\Sigma}$ | Riemannian mean . . . . . 80 |
| $\bar{\delta}^N$ | Expected value of $\delta_i^N$ . . . . . 82 |
| $\bar{\delta}^T$ | Expected value of $\delta_i^T$ . . . . . 82 |
| $\bar{s}_{n_p}$ | Average ERP peak amplitude . . . . . 44 |
| $\Delta$ | Set of time sample differences denoting peak latency variability . . . . . 44 |
| $\delta^N$ | Probability distribution of the distance of a nontarget feature vector to the nontarget class center . . . . . 81 |
| $\delta^T$ | Probability distribution of the distance of a target feature vector to the target class center . . . . . 81 |
| $\delta_C$ | Average distance between a feature and its class center . . . . . 79 |
| $\delta_I$ | Average distance between two same-class features . . . . . 79 |
| $\delta_i$ | Dirac distribution at location $i$ . . . . . 35 |
| $\mathrm{d}_E$ | Euclidean distance . . . . . 80 |
| $\mathrm{d}_R$ | Riemannian distance . . . . . 32 |
| $\gamma_0$ | Transport plan, optimal transport solution . . . . . 35 |
| $\hat{X}_N^l$ | Proxy narget average . . . . . 95 |
| $\hat{X}_T^l$ | Proxy target average . . . . . 95 |

# PART I

# BACKGROUND

# CHAPTER 1
# ERP-BASED BRAIN COMPUTER INTERFACES

This chapter provides a background on ERP-based BCI. We first introduce the ERP-based BCI system and provide a short history of its conception and evolution. We then proceed to detail the system and provide state-of-the art references for each component. We detail the issue of BCI calibration and briefly mention some of the existing solutions. We conclude by presenting our approach towards zero-calibration BCI, which is the the main objective of this thesis.

## 1.1 Introduction

An ERP-Based BCI is a system composed of several components. During each session, these components interact with each other and with the user in a closed loop. In general, a BCI session refers to the continuous use of a BCI during which the user does not remove the EEG acquisition device. Each one of these components can be seen as pipeline which comprises different subcomponents. A set of functions and parameters are linked to each subcomponent. In a state-of-the art system for a visual P300-Speller, we identify four main components: (i) interface (ii) acquisition (iii) information extraction (iv) system update. These are outlined in figure 1.1. Note that this particular taxonomy can be generalized to different ERP-based BCI as well.

This thesis focuses on the sources of EEG variability. In particular, we are interested to detail how this variability affects the system performance and what parameters can be modified or adapted to resolve the issues that arise because of that variability. The design of an adaptive ERP-Based BCI necessitates a solid understanding of how each system component is designed and how it interacts with the other components [Mladenovic et al., 2018]. To this end, this chapter is

organized in the following way. First, we provide a brief history of ERP-Based BCI. We the detail the system components presented in figure 1.1 one by one. Finally, we expose and discuss the pre-BCI use calibration issue.

## 1.2 Brief History of ERP-based BCI

The first BCI to use ERPs was detailed in 1988 by Farwell and Donchin [Farwell and Donchin, 1988]. Its objective is to allow the user to spell words by means of an on-screen grid-like keyboard, whereby rows or columns are flashing. The user has to attend the screen and concentrate on the character they wish to spell, disregarding the rest of the characters on the keyboard. This task can be viewed as a covert discrimination task between two types of occurrences: either the row or column flashing contains the desired character, or not. This BCI paradigm relies on the elicitation of a well-studied ERP component, known as the P3b, the late positive complex, or simply, the P300.

Donchin et al. proved the feasibility of this ERP-based speller [Donchin et al., 2000], which has since been studied and used extensively under the name *P300-Speller*. Recent studies focus on achieving better performances by either optimizing the signal processing and classification framework [Blankertz, 2004; Guger et al., 2009b; Rivet et al., 2009; Blankertz et al., 2011; Kindermans et al., 2012a], or by modifying the decision process and the way items are presented on-screen. [Townsend et al., 2010; Thomas et al., 2014; Mattout et al., 2015]. The P300 component has also been used in a BCI gaming application, presented by Congedo et al. [Congedo et al., 2011], in which the user is playing a P300-based BCI version of the arcade game “alien invaders”.

Other ERP components have been used in BCI besides the P300. The N400, a negative component which is related to recognition of meaningful stimuli, has been used in a P300-Speller-like BCI by Kaufmann et al. in [Kaufmann et al., 2011]. The authors propose to replace flashing keys with familiar faces, which elicit the N400 component and facilitate trial classification. The Error Potential, a component elicited not by a conventional stimulus but by the recognition of a mistake, has also been used in a number of BCI paradigms. Mattout et al. use the Error Potential as a spelling correction tool in a P300-Speller [Mattout et al., 2015]. In [Iturrate et al., 2015], Iturratte et al. design a BCI that uses the Error Potential to

control an on-screen item.

In the remainder of this thesis, we are mainly considering the P300-Speller paradigm. In the following section, we detail the four main components of a P300-Speller system, depicted in figure 1.1.

## 1.3 The BCI System

### 1.3.1 The Interface

The interface is the point where the user and the system immediately interact. The interface component receives constant updates from the system update component. According to that information, it generates stimuli or provides feedback to the user. The user generates EEG signal which is subsequently recorded by the acquisition component. Hence, the interface encloses everything that is related to the task and the user. Task-related parameters include the choice of the paradigm and the strategy that should be employed by the user to achieve the desired result. These are often selected by the experimenter. In a P300-Speller, the screen displays a keyboard on which groups of characters are flashing. The user is asked to count incrementally every time he sees a flash on the character he wishes to spell. The interface holds parameters such as the groups of characters that flash and the interval between two flashes.

In ERP-based BCI, user-specific mental states and characteristics, such as arousal levels, mood and mental workload directly affect the temporal pattern of the ERP [Polich and Kok, 1995; Polich, 2009; Jeunet et al., 2016; Mladenovic et al., 2018]. Moreover, they are subject to a high amount of variability across different users [Lotte and Jeunet, 2015; Jeunet et al., 2016]. This has a direct impact on the rest of the system. In particular, the information extraction component needs to be adjusted accordingly to deal with the resulting variability in the EEG signal.

### 1.3.2 EEG Acquisition and Processing

The acquisition component is responsible for recording the EEG signal, preprocessing it and transmitting it to the information extraction component. Parameter choices typically concern the equipment used (EEG acquisition device, amplifier),

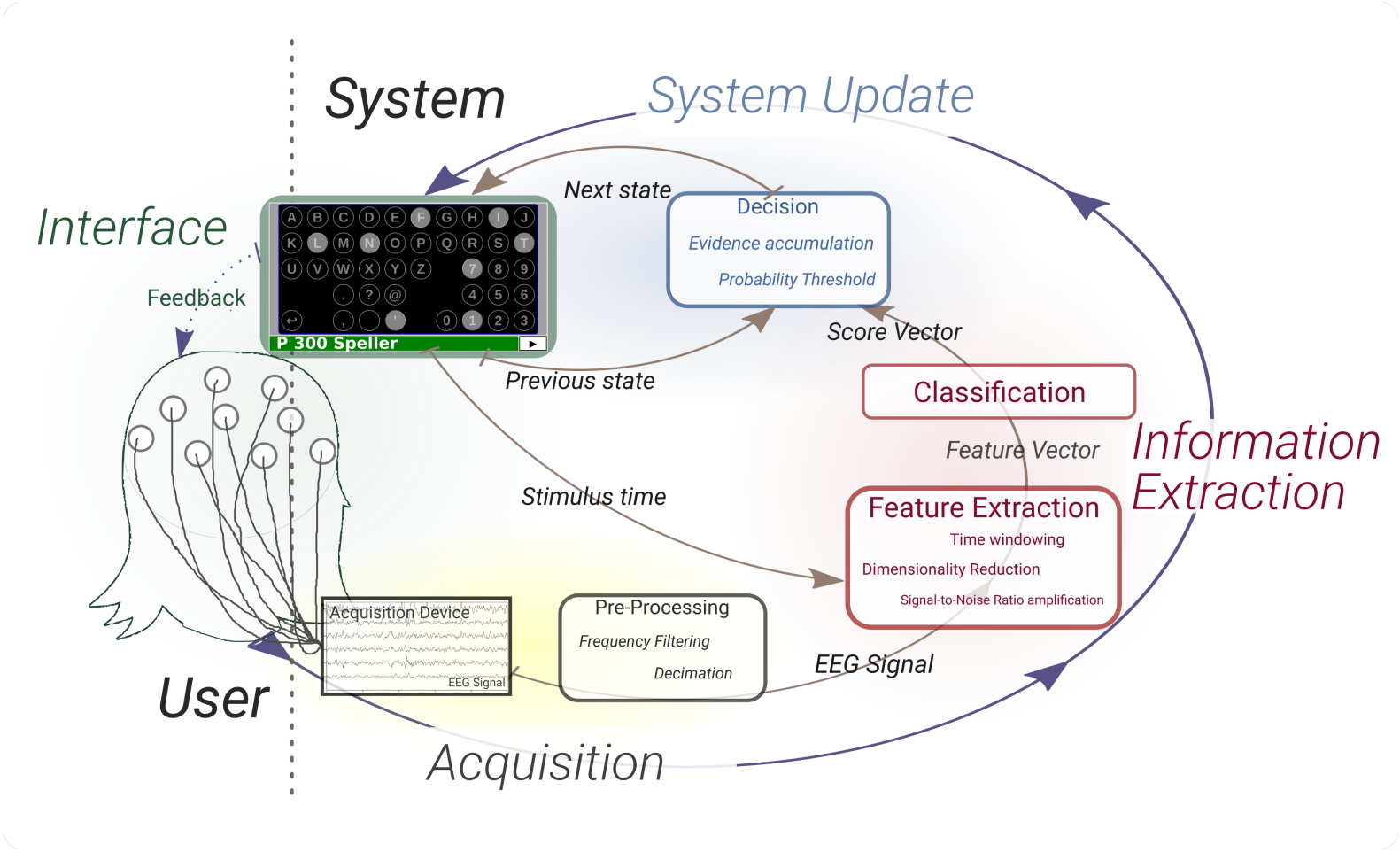



Figure 1.1: The four system components of a state-of-the-art P300-Speller, divided into subcomponents. The data goes through the interface component, the acquisition component, the information extraction component, the system update component and back to the interface component in a closed loop. However, additional interactions can take place between components that are not in the main loop. For example, the interface can directly communicate stimulus times to the information extraction component.

the choice of electrodes, and their positioning. During a non-invasive EEG-based BCI experiment, the EEG signal is acquired on the scalp through electrodes that record the electrophysiological activity that is generated from cortical neuronal activity and transmitted through the skull and scalp [Niedermeyer et al., 2005]. EEG recordings have a very low amplitude, which is on the order of $\mu$V. Therefore the EEG measurements need to be processed by an amplifier. Typically, digital amplifiers are employed for this task which provide one discrete signal per sensor. Most amplifiers offer a sampling frequency that can range between 100 and 1000Hz.

The EEG signal has the advantage of having a high temporal resolution in contrast to its poor spatial resolution, which is restricted by the number of electrodes. The number of electrodes varies with respect to the paradigm: it can range from a few electrodes up to 128 electrodes. (Note that BCI experimenters avoid using more than 64 electrodes, since the process of placing them on the scalp is lengthy and tiresome for the user.) In addition, EEG measurements have a very low Signal-to-Noise Ratio. Therefore, before extracting features of interest,the EEG signal goes through a pre-processing pipeline. Since the physiological markers of interest in ERP-based BCI live in low frequency components, the raw signal is bandpass filtered so that only those frequencies of interest are kept in the signal. Often, the signal goes through an additional downsampling step. Note that some amplifiers provide built-in signal pre-processing methods.

### 1.3.3 Extracting Information from the EEG signal

The information extraction components is responsible for extracting relevant information from the pre-processed EEG signal. It is often represented as a pipeline of two subcomponents which perform feature extraction and classify the resulting feature vector. For example, visual P300-Spellers rely on the detection of the P300 component, which should be elicited every time the user's character of interest flashes. The pre-processed EEG signal is thereby given as input to the feature extraction subcomponent, along with information concerning the timing of the flashes. Its aim is to output one feature vector per flash in a way that ensures high inter-class separability and Signal-to-Noise Ratio (SNR). Each feature vector is subsequently transmitted to the classification subcomponent, which decides whether the stimulus associated to a feature vector was a target stimulus or a

nontarget stimulus and transmits the resulting vector of probabilities to the system update component. Target stimuli correspond to groups of characters that contain the character that the user wishes to spell, as opposed to nontarget.

Due to the high amount of variability in the signal and to the low SNR, advanced signal processing, feature extraction, and classification methods have been employed and incorporated within BCI systems. In ERP-based BCI, the pre-processed EEG signal $X$ is segmented into trials of a specific duration, starting from stimulus offset. The number of time samples should be chosen so that the relevant information is contained within the time segment. Each trial can be represented as a matrix $X_i \in \mathbb{R}^{I_c \times I_w}$, where $I_c$ denotes the number of electrodes and $I_w$ the number of time samples. These trials can be converted directly into *spatiotemporal features* $\mathbf{x}^i \in \mathbb{R}^{I_c \cdot I_w}$, where vector $\mathbf{x}^i$ contains the rows of matrix $X_i$.

Alternatively, the trials are projected onto a lower dimensional subspace through some spatial projection matrix $V \in \mathbb{R}^{I_f \times I_c}$, where $I_f$ denotes the number of components, i.e. the new spatial dimensionality of the signal. These spatial projection matrices are often referred to as spatial filters, since they modify the spatial dimension of the signal. The new subspace often has a property that enhances some relevant characteristic, such as class separability [Lotte and Guan, 2011] or SNR [Rivet et al., 2009]. These projected trials are also converted into spatiotemporal features $\mathbf{x}^i \in \mathbb{R}^{I_f \cdot I_w}$.

The resulting feature vectors are given to the classifier component. The classification output usually takes the form of a vector $\mathbf{p}^i = (p_1^i, p_2^i, \cdots, p_{I_l}^i)$, where $p_l^i$ denotes the probability that the $i^{th}$ feature vector belongs to class $l$ and $I_l$ denotes the number of classes. State-of-the-art classifiers for ERP-based BCI include linear classifiers such as Linear Discriminant Analysis (LDA) [Panicker et al., 2010; Blankertz et al., 2011; Gayraud et al., 2017] and Support Vector Machines (SVM) [Rakotomamonjy and Guigue, 2008]. For an extensive review of BCI classification methods, we refer the reader to [Lotte et al., 2007, 2018]

### 1.3.4 System Update

The system update component holds the parameters related to the decision process. In particular, given the previous state of the system and a new probability vector, it decides on the new state of the system and transmits the decision to the

interface. In a P300-Speller, the system states include whether the system should continue flashing groups of characters, or if a character has been found and can be displayed on the screen. For example, the P300-Speller system designed by Thomas et al. [Thomas et al., 2014] accumulates evidence for each character in the keyboard and uses an early stopping criterion to decide when the evidence accumulation has converged and a character can be selected. The same criterion is applied in the work of Mattout et al., who also incorporate an error detection strategy in which the information process module has the additional task of detecting an Error Potential at the end of each sequence of flashes [Mattout et al., 2015]. If an error has been detected, the system update corrects the character by choosing the second most probable character.

## 1.4 System Calibration

In a process that precedes BCI use, the system goes through a calibration which aims to tune its parameters. These parameters typically concern the spatial filter coefficients and classification weights within the information extraction component, as described in section 1.3.3. Each BCI paradigm has its own calibration strategy. For example, during the calibration of a P300-Speller, the system tries to extract a template of the ERP that should be generated by the user when the character he desires to spell is flashing. Therefore, the user is asked to focus on specific characters. This allows the BCI to gather labeled data and train the system pipeline. The process may last between 5 and 10 minutes, depending for example on the quality of the acquired signals or the classifier scores [Lotte and Congedo, 2016b].

While the reported performances of BCI applications have been satisfactory, state-of-the art feature extraction and classification methods have been unable to generalize across different sessions and different subjects [Clerc et al., 2016]. This means that the system must be calibrated before each session. Regarding the classification process in particular, the high amount of variability in the EEG signal combined to the low SNR lead to changes in the feature domain. The broad use of BCIs greatly depends on discarding the need for calibration sessions. BCI users should not have to undergo the tedious calibration process each time they want to use a BCI.

A solution to this problem is the application of transfer learning. This machine

learning approach is increasing in popularity in the BCI domain [Barachant et al., 2010; Kindermans et al., 2014; Barachant and Congedo, 2014; Gayraud et al., 2017]. Transfer learning approaches allow for the combination of acquired knowledge, which can take the form of multiple training set or multiple classifiers. Hence, it can enable us to efficiently use information from previously acquired data to calibrate the BCI, without having to ask the user to perform an additional calibration session. Ideally, a global solution would take the form of a transfer learning method that deals with the all of the sources and types of variability in BCI simultaneously. To do that, it is imperative that we understand the variability in the EEG signal and how various transfer learning methods deal with its sources.

## 1.5 Conclusion

BCI are intricate systems, so the choices made for each component are directly affected by other parts of the system, and in turn affect other component choices as well. It is therefore imperative to understand how these interactions take place. For example, choosing to detect a neurophysiological marker such as the P300 component implies that the user has to perform a task that necessitates attention; that the system will present some kind of stimulus to elicit the P300; and that it will have to search for the temporal pattern that characterizes this particular component.

Calibrating the system is not a trivial task, on account of both the system complexity and the variability of EEG signals and ERP components. In spite of this variability, we aim for BCI that do not need calibration before usage. Throughout the rest of this thesis, we focus on how the variability that stems from the interfaces affects system performance and in which way the information extraction component can adapt to it. In order to render our results interpretable, we will be assuming that the acquisition and system update parameters are fixed. These parameters will be detailed when necessary. In particular, we analyze the sources of EEG variability and investigate selected adaptive machine learning methods that adjust to this variability in terms of classification performance. Finally, we assess whether this approach suffices to design zero-calibration ERP-based BCI.

# PART II

# VARIABILITY IN ERP-BASED BCI

# CHAPTER 2
# SOURCES OF VARIABILITY IN ERP-BASED BCIS

In the previous chapter, we described the ERP-based BCI system and discussed the interactions between system components. In particular, we are interested in how the EEG signal variability affects system performance. Our goal is to quantify variability. In particular, we are interested in retrieving parameters that allow us to determine which adaptive machine learning methods are robust to that variability. In this chapter we analyze the sources of variability in ERP-based BCI. First, we review the existing bibliography on the variability of ERP components and propose categorizations of ERP variability. Then we perform an analysis on two experimental datasets. This analysis provides us with a way to parameterize EEG variability and gives us insight on the relationship between the sources of variability and their effect on the EEG signal.

## 2.1 Introduction

ERPs are comprised of a group of components presumed to be involved in human information processing, reflecting factors such as stimulus registration, attention and evaluation [Michalewski et al., 1986]. The sources of ERP variability have been the center of extensive research [Michalewski et al., 1986; Polich and Kok, 1995; Polich, 2009]. Regarding the EEG signal generated during an ERP-based BCI session, we distinguish between three primary sources.

The first is the inherent variability of the ERP components. ERP variability is typically measured in terms of peak amplitude, peak latency, and scalp topography, i.e., the amplitude change over EEG electrodes [Polich, 2009]. While the variability of sensory-related components present in an ERP has been found to be fairly low,

the same cannot be said of the P300 [Michalewski et al., 1986; Jung et al., 2001; Dalebout and Robey, 1997].

The second source of variability is noise. This noise contains physiological artifacts, such as blinks and muscle movement, technical artifacts, but also background brain activity that is unrelated to the task [Clerc et al., 2016]. EEG signals have a very low SNR ratio, which makes ERP extraction a difficult task

The third source of variability is scalp topography, which we have already mentioned as a source of ERP variability. We put it here in a larger context which includes all factors that contribute to scalp topography variability. EEG recordings are prone to spatial variability related to the location of the sources of activity in the brain, the (dipole) orientation of the sources of activity, the location of the electrode on the scalp, and the conductivities of the intermediate layers [Bledowski, 2004; Papageorgakis, 2017]. ERPs that arise from stimulus discrimination tasks comprise other components that are related to sensory processing, namely, the sensory evoked N1, P1, N2 and P2 components. Similar to the P300, they are named after their peak latency, that is, the peak negative or positive polarity observed at a specific time after stimulus onset. The contribution of these sensory components depends on the paradigm, whether for instance the presented stimuli are auditory, or visual [Michalewski et al., 1986; Saavedra and Bougrain, 2012].

The aforementioned sources of variability can occur either across different sessions, or within the same session. For the most part, variability analyses across sessions extract ERPs by averaging multiple trials. Therefore, they study the variability of the average ERP. On the other hand, single trial analyses, otherwise known as trial-to-trial analyses, reveal the variability within the same session, which is often referred to as across-trial variability. The cross-session approach provides more information on the correlation of the sources of variability (e.g. task relevance, attention) to their quantitative effects on the signal (e.g. average peak amplitude variability). Trial-to-trial analyses provide a more detailed view on these effects and allow us to have a better insight on the noise that is present in the signal, since no averaging is performed.

In the following sections, we review the different variability sources for various ERP components, such as the P300, the Novelty P3 and the N1, P1, N2 and P2 components. Then, we perform a trial-to-trial and a cross-session variability analysis on two experimental datasets.

## 2.2 Variability of ERP Components

### 2.2.1 The P300 Component

The P300 component, otherwise known as P3b or late positive component, was first reported in 1965 by Sutton et al. [Sutton et al., 1965]. It has been observed to arise during auditory and visual stimulus discrimination tasks. It is characterized by a recorded positive amplitude peak around 300 milliseconds after stimulus onset, which is most prominent on the middle parietal, central and frontal electrodes (Pz, Cz, Fz). Further research showed that it is elicited most strongly under the "oddball" paradigm, in which a frequently presented stimulus is interweaved by a less frequent one. Usually, the user is asked to take notice of the latter [McCarthy and Donchin, 1976; Donchin et al., 1978; Pritchard, 1981]. The P300 component is associated with attention and memory operations [Polich, 2009].

P300 variability is affected by a number of physiological and environmental sources. Table 2.1 summarizes the sources of variability and the affected component characteristics, namely peak amplitude, peak latency and scalp topography. Isreal et al. [Isreal et al., 1980] demonstrate the effect of introducing a second task at the same time as target-nontarget stimulus discrimination. The subjects are asked to discriminate between auditory stimuli while at the same time performing a tracking task. In this task, they were asked to correct the position of a moving cursor using a joystick with their right hand. The correct position was in the center of a screen. They show that, while the introduction of the second task decreases the peak amplitude of the P300, increasing the tracking task difficulty does not affect the waveform. A review of the sources of P300 variability was provided by Polich et al. in 1995 [Polich and Kok, 1995]. In this research, the sources of P300 variability are grouped into natural factors, which include circadian and ultradian rhythms, seasonal variations, and menstrual cycle; and environmentally induced factors, such as exercise, fatigue levels, sleep deprivation, and drug intake.

In particular, Katayama et al. [Katayama and Polich, 1999] assess the variability of the P300 (termed P3b) between auditory and visual paradigms in a 3-stimulus paradigm. Their findings demonstrate a clear difference between peak amplitude and peak latency for the two modalities. Visual stimuli generate higher peak amplitude and latency values that those generated by auditory stimuli. Nevertheless,

Table 2.1: Sources of variability and their effects on peak amplitude; peak latency; and scalp topography of the P300 component.

| **Variability Source** | **Modulates Peak Amplitude** | **Modulates Peak Latency** | **Modulates Scalp Topography** |
|---|---|---|---|
| Ultradian Rhythm (90 min) | Yes, 12-27$\mu$V [Polich and Kok, 1995] | Yes, 320-385 ms [Polich and Kok, 1995] | - |
| Circadian Rhythm (Indirect) | No [Polich and Kok, 1995] | 220-380 ms, correlated with body temperature and heart rate [Polich and Kok, 1995] | - |
| Food intake | Yes, 8-18 $\mu$V [Polich and Kok, 1995] | No [Polich and Kok, 1995] | - |
| Target-to-Target Interval, Target Stimulus probability | Yes, 6-15$\mu$V (auditory), 10-25$\mu$V (visual) [Gonsalvez et al., 2007; Polich, 2009] | - | - |
| Task difficulty | No [Isreal et al., 1980] | No [Isreal et al., 1980] | - |
| Second task | Yes, depending on the task [Isreal et al., 1980] | No [Isreal et al., 1980] | - |
| Auditory vs Visual Stimulus | Yes [Katayama and Polich, 1999; Yagi et al., 1999; Gonsalvez et al., 2007; Polich, 2009] | Yes [Yagi et al., 1999; Katayama and Polich, 1999] | No [Katayama and Polich, 1999] |
| Task type | Yes [Polich, 2009] | Yes [Polich, 2009] | Yes [Polich, 2009] |
| Exercise | Yes, 6-15$\mu$V (auditory), 10-18$\mu$V (visual) [Yagi et al., 1999; Polich and Kok, 1995] | Yes, 340-380 ms (auditory), 380-420 ms (visual) [Yagi et al., 1999; Polich and Kok, 1995] | - |
| Age | Yes, <30$\mu$V [Dinteren et al., 2014; Walhovd and Fjell, 2002] | Yes, 250-500 ms [Dinteren et al., 2014; Walhovd and Fjell, 2002; Polich, 2009] | - |
| Cognitive performances | Yes [Polich and Kok, 1995] | Yes [Polich and Kok, 1995] | - |
| Drug intake (Caffeine, Nicotine, Alcohol) | Affected by Caffein and Alcohol intake [Polich and Kok, 1995] | Yes [Polich and Kok, 1995] | - |
| Fatigue | Yes [Polich and Kok, 1995] | Yes [Polich and Kok, 1995] | - |
| Sleep deprivation | Yes,<18$\mu$V [Polich and Kok, 1995] | Yes [Polich and Kok, 1995] | - |
| Sensor Position | - | - | Yes [Clerc et al., 2016] |

the scalp topography appears unaffected. These findings are in accordance with the results of Yagi et al. [Yagi et al., 1999], who also show that physical exercise affects both P300 peak amplitude and latency.

Age is another source of variability, which is demonstrated to affect both peak amplitude and latency [Walhovd and Fjell, 2002; Dinteren et al., 2014]. According to the findings of Dinteren et al., [Dinteren et al., 2014], who performed a meta analysis on 75 studies, peak amplitude increases until late adolescence and gradually decreases after that, while peak latency until early adulthood and increases thereafter. Finally, in an review on ERP components, Polich et al [Polich, 2009] note the effect of Target-to-Target interval on peak amplitude and peak latency (in accordance with Gonsalves et al. [Gonsalvez et al., 2007]) as well as the importance of arousal and attention. They also note that ERP components are genetically transmitted: P300 components are similar among members of the same family.

The above researches largely focus on the variability of the average ERP across different experiments. However, as noted by Makeig et al. [Makeig et al., 2004], the ERP average can differ significantly from the single trials it is derived from. Further research has investigated trial-to-trial variability within the same experiment to assess the degree of change in the peak amplitude and latency of single trials. In their work, Michalewski et al. study ERP components in an auditory paradigm, in order to determine the effects of latency variation on the ERP grand average [Michalewski et al., 1986]. They reported peak latency varies between 177 and 363 ms at the Pz electrode, while the peak amplitude varies between approximately 10 and 25 $\mu$V, which shows the magnitude of trial-to-trial variability. Jung et al., in results obtained in a visual 2-stimulus experiment, reveals an important amount of peak latency variability across trials in the same session, which appears to be correlated to Response Time [Jung et al., 2001]. These results are corroborated by Gramfort et al. [Gramfort et al., 2010].

Physiological and environmental sources of ERP variability contribute to different types of EEG variability. EEG variability can be investigated according to two different taxonomies. First, whether it is the same individual using the BCI or not, in which case we define intra-individual variability and inter-individual variability [Clerc et al., 2016]. Then, variability can also be studied across different BCI sessions, or within the same session, in which case we talk about inter-session and intra-session variability. While inter-individual variability by definition refers

to different sessions, intra-individual variability can occur either across different sessions, or within the same session. On table 2.2, we propose an classification of the sources of P300 variability presented in table 2.1 according to both taxonomies. Note that, unsurprisingly, inter-individual variability gathers the largest amount of sources. Regarding intra-session variability, we assume that changes in the amount of tasks, in the task difficulty or the stimulus probability can occur within the same session.

Table 2.2: A categorization of the various sources of ERP variability according to two different taxonomies: (i) whether they occur across different sessions or within the same session; and (ii) whether they occur across different subjects or for the same subject.

<table>
<tr><td colspan="2">Inter Session</td><td>Intra Session</td></tr>
<tr><td>Inter Subject</td><td colspan="2">Intra Subject</td></tr>
<tr><td>Biological Factors</td><td colspan="2"></td></tr>
<tr><td>Cognitive Skills</td><td colspan="2"></td></tr>
<tr><td colspan="2">Food Intake, Age, Exercise</td><td></td></tr>
<tr><td colspan="2">Sleep Deprivation, Drug Intake</td><td></td></tr>
<tr><td colspan="2">Circadian Rhythm</td><td></td></tr>
<tr><td colspan="3">Arousal Levels, Fatigue, Sensor Position, Stress, Ultradian Rhythm</td></tr>
<tr><td colspan="3">Second Task, Stimulus Probability, Task Difficulty</td></tr>
</table>

### 2.2.2 The P3a Component

The P3a Component or “no-go” component [Polich, 2009] is elicited by distractor targets, in contrast to the P300 which is related to information processing operations. Regarding P3a variability, Polich et al [Polich and Kok, 1995] note that the P3a is more sensitive to inter-individual variability than the P300, but also point out the fact that latency variability, otherwise known as latency jitter, can affect the observations related to peak amplitude. The sources of the P3a component are found to be more central/frontal than those of the P300, suggesting that the two components have a distinct topography [Polich, 2009]. Bledowski et al locate P3a generators in the precentral sulcus and anterior insula [Bledowski, 2004].

Note that the P3a should not be confused with the “novelty” P3 component. This

particular ERP component, which is more prominent in the frontal lobe, is elicited by novelty targets. Its peak latency is similar to the P3a, but its peak amplitude decreases over time due to habituation [Polich, 2009].

### 2.2.3 Sensory Evoked Potentials

Sensory evoked potentials are low-amplitude positive and negative peaks that are generated within 200 ms after stimulus onset. These components however do not vary to the same degree as the P300 component. Michalewski et al. study the variability of the auditory N1, P1 and P2 components, which is indeed found to be less important than the P300 component [Michalewski et al., 1986]. These results are also corroborated by the findings of [Jung et al., 2001] and [Makeig et al., 2004]. In the same study, Michalewski et al. investigate the intertemporal relationships between the aforementioned components. The temporal correlation between them is mostly found to be low, indicating that the processes that generated them are independent.

## 2.3 Analysis on Experimental Datasets

### 2.3.1 Dataset description

We study the EEG signal and ERP variability in two experimental datasets that contain EEG signal recorded during P300-Speller calibration sessions. In these sessions, the screen displayed a keyboard on which groups of letters were flashing. More specifically, for each letter, the user was asked to focus on a particular letter, while counting incrementally the number of times it flashed. The flashing strategy consists of groups of letters flashing in specific patterns, as described in the work of Thomas et al. [Thomas et al., 2014]. The flash ratio of target letters (that the subject was asked to focused on), versus nontarget letters was set equal to 1/5. The interval between consecutive flashes was set to 300ms. The interval between two consecutive letters was set to 2s. No feedback was presented during calibration, i.e. the user did not receive any information from the system on their performance. In both experimental datasets, a Refa-8 amplifier (ANT) was used for the recording. We analyze the EEG signals of 12 electrodes (Fz, C3, Cz, C4, P7, P3, Pz, P4, P8,

O1, Oz, O2), downsampled at 64Hz and filtered with a 4th order Butterworth filter between 1 and 20Hz.

Dataset A includes EEG signals from four healthy subjects, which were recorded during P300-Speller sessions conducted in the premises of Inria Sophia-Antipolis Méditerranée. Each subject participated in three free-spelling sessions, each preceded by a calibration session. Here, we only include the calibration sessions. During the calibration sessions, the subjects were asked to spell the word "CALIBRATION". The number of repetitions (flashes) per target letter was set to 6.

The second dataset used in our experiments, dataset B, consists of calibration sessions that were conducted by 20 adult patients suffering from Amyotrophic Lateral Sclerosis. Each subject participated in three free-spelling sessions, each one preceded by a calibration session. The experiment took place in the premises of the Nice University hospital, and had been approved by the local ethics committee CPP Sud Méditerranée [Guy et al., 2018]. During the calibration sessions, the subjects were asked to spell 10 random letters. The number of repetitions (flashes) per target letter was set to 20.

### 2.3.2 Analysis

**ERP component extraction and visualization** We conduct a trial-to-trial analysis on the recordings of dataset A and B. The EEG signal of each session is segmented into trials lasting 0.6 seconds, starting from stimulus onset. In the trial-to-trial analyses mentioned in section 2.2, the authors used the recordings of single electrodes, typically Pz, Cz and Fz. This allowed them to perform trial-to-trial analysis within a single session and measure the peak amplitude and latency variability across trials. However, a trial-to-trial variability analysis across sessions that uses single electrode measurements does not take into account scalp topography variability.

In order to study the variability of peak amplitude, peak latency and scalp topography simultaneously, we chose to extract the ERP component using the Xdawn algorithm. Xdawn is a state-of-the art feature extraction method for ERP-based BCI, described by Rivet et al. [Rivet et al., 2011]. Let $X \in \mathbb{R}^{I_c \times I_t}$ denote the EEG signal acquired over $I_c$ electrodes, where $I_t$ denotes the total amount of time samples. The algorithm's objective is to produce a projection matrix

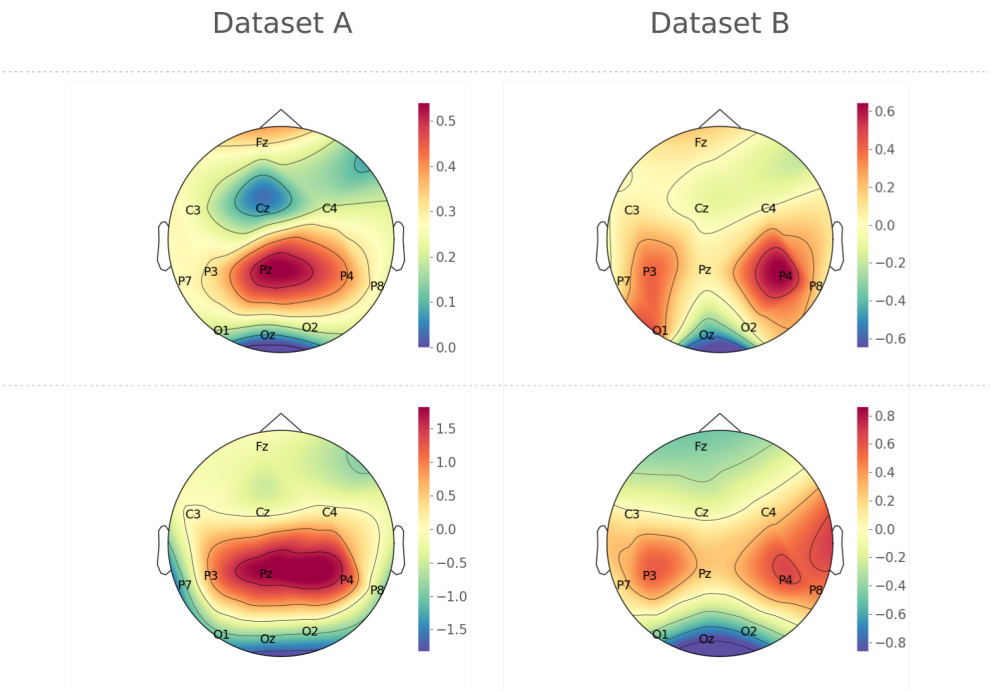



Figure 2.1: Trial-to-trial analysis for datasets A and B. (a) Grand average of all target trials in the dataset. (b) ERP image of the target trials (extracted using the first Xdawn filter), sorted by peak latency. (c) Coefficients of the first Xdawn filter. (d) Scalp topography resulting from the inverse of the first Xdawn filter.

$V \in \mathbb{R}^{I_f \times I_c}$ that projects the signal $X$ onto a subspace of dimension $I_f$, where the ERP component variance is maximized while the signal variance is minimized. Let $A_t \in \mathbb{R}^{I_c \times I_w}$ be the archetype of the scalp topography of a P300 component over time, where $I_w$ is the number of samples in a specific time window that immediately follows stimulus onset. This response can be calculated as the average over all target trials (trials whose onset stimulus is a target stimulus and are thus assumed to contain the P300 component) or using the least squares method proposed in [Rivet et al., 2009, 2011]. The projection matrix $V$ is composed of concatenated vectors $v_i$ that are the first $I_f$ maximizers of the the following Rayleigh quotient,

$$\frac{v^\intercal \Sigma_{A_t} v}{v^\intercal \Sigma_X v}$$

where $\Sigma_{A_t} = \frac{1}{I_w} A_t A_t^\intercal$ and $\Sigma_X = \frac{1}{I_t} X X^\intercal$ are the covariance matrices of the

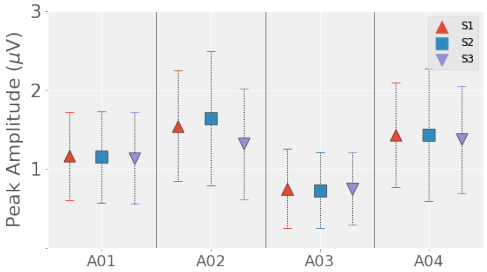
archetype response $A_t$ and the signal $X$ respectively. Therefore, $V$ is the matrix whose $I_f$ rows are the $I_f$ eigenvectors associated to the $I_f$ largest eigenvalues of the generalized eigenvalue problem $\Sigma_{A_t} v = \lambda \Sigma_X v$.

Using the Xdawn algorithm has the advantage of allowing us to extract a single component $I_f = 1$. Since the subspace maximizes the variance of $A_t$, we can safely assume that this component contains the ERP. At the same time, matrix $V^{-}1$ provides us with a topography that indicates the sites where the P300 is most prominent. We compute one projection matrix for each dataset, by concatenating all signals and computing an archetype response $A_t$ using the least squares method described in [Rivet et al., 2011]. Then, we project each signal onto the subspace generated by the first component and segment it into one-dimensional trials $x_i \in \mathbb{R}^{I_w}$ that start from stimulus onset and last approximately 0.6 seconds, resulting into $I_w = 38$ time samples.

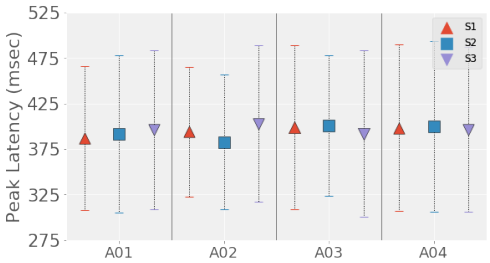
In [Jung et al., 2001; Makeig et al., 2004; Gramfort et al., 2010; Delorme et al., 2015], the authors use a a visualization tool called an ERP Image for trial-to-trial analysis. An ERP Image represents stacked trials of the same length. Typically, the trials are ordered according to some meaningful measure such as response time or peak amplitude.

We create one ERP image per dataset by ordering the trials according to peak latency. For each trial, we compute the peak amplitude by searching for the highest value starting from 220 ms (14th time sample) until the end. The time sample which holds that value denotes the peak latency. On figure 2.1 we can see the ERP images of each dataset, the average target response, filter coefficients induced by the the first eigenvector of $V$ and the scalp topography computed using $V^{-1}$. By looking at these ERP figures of each dataset, we can already distinguish the variability of both the peak amplitude and the peak latency of the P300 component. We can also observe that the sensory components, namely, the N100, P100 and N200 present very little latency variability. Dataset A presents a lower trial-to-trial peak latency variability: the average peak latency variability is equal to $\approx 380$ms $\pm 60$ms. On the other hand, dataset B has a much higher trial-to-trial peak latency variability, equal to $\approx 420$ms $\pm 130$ms. In addition, dataset B also has a higher average peak amplitude than dataset A, equal to $\approx 6\mu$V against $\approx 2\mu$V. Finally, looking at the spatial filter coefficients, we can observe significant scalp topography differences between the two datasets. The most prominent electrodes for dataset A

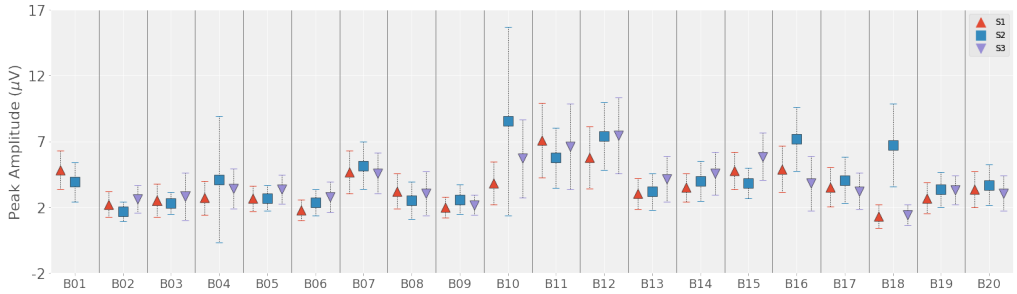
are electrodes Fz, Pz and P4, while for dataset B these are Fz, P3 and P4.



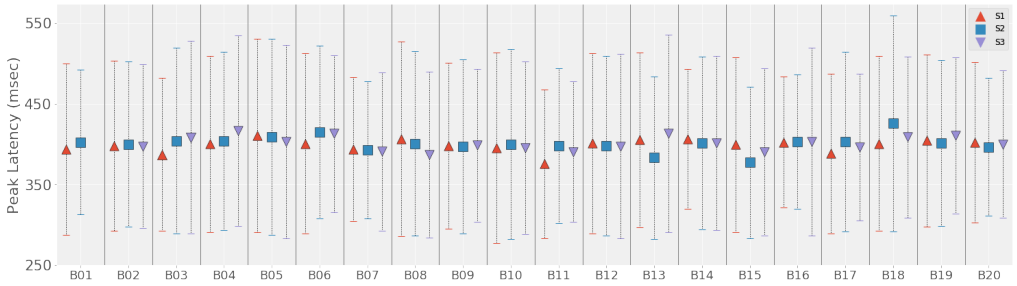

(a) Average peak amplitudes, dataset A.




(b) Average peak latencies, dataset A.




(c) Average peak amplitudes, dataset B.




(d) Average peak latencies, dataset B.

Figure 2.2: A plot of the average and standard deviation of the peak amplitudes and peak latencies for each session and each dataset. This allows us to assess the inter-session and intra-session variability of these two components at the same time. Axx and Bxx denote the subjects of datasets A and B respectively, while Sx denotes the session index of a particular subject.

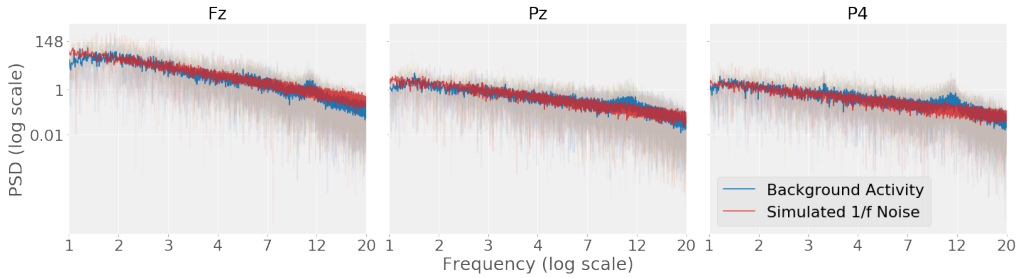
**Variability of the Peak Amplitude and Latency** The ERP images give us a very useful global view of the trial-to-trial amplitude and latency variability throughout the entire dataset. We can at the same time verify that sensory components do not vary in amplitude and latency compared to P300. However, they do not provide any information on the across session and across subject average peak amplitude and latency variability. Therefore, we group the trials by session and perform a cross-subject and cross-session variability analysis. In particular, we compute the average and standard deviation of the single-trial peak amplitudes and latencies, for each subject and each session, in both datasets.

On figures 2.2a and 2.2c, we display the results for peak amplitude variability. We can see that for dataset B in particular, the average peak amplitude varies significantly, taking values between 1,5 and 9 $\mu$V. The standard deviations about average peak amplitudes for each session indicate that trial-to-trial peak amplitude variability is different across sessions and across subjects as well. For example, subject B18 has a low average peak amplitude and peak amplitude variability in the first and third session, but in the second session, both these values are high.

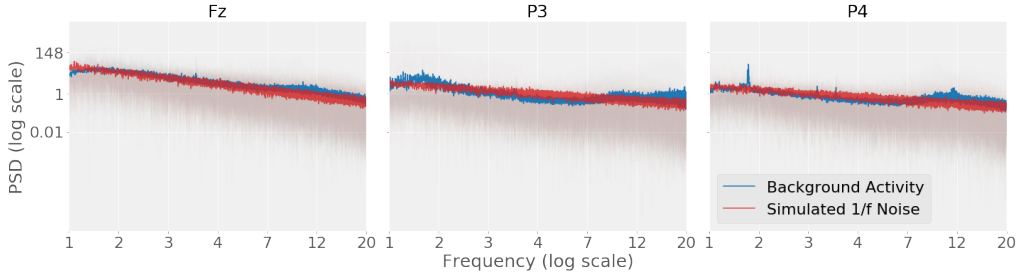
Figures 2.2b, 2.2d show the same analysis for peak latency. The average peak latency for both datasets takes values between 360 and 400 millisecond. The standard deviations reflect the inter-session peak latency variability, taking values that range from $\pm$100ms to $\pm$250ms. Note that these values, for both peak amplitude and peak latency, are in accordance with the literature (Table 2.1).

**Noise** In an EEG recording, the signal that contains the ERPs also encloses on-going activity that is not time-locked with the stimulus, as well as artifacts and additive noise. Background EEG activity has been known to contribute to the peak amplitude and latency variability of the ERP [Polich, 1997]. EEG activity has been observed to possess a $1/f$ frequency spectrum. Such processes, also known as pink noise, have been often observed to arise in biological systems. In [Ward, 2002], the authors study the frequency spectrum of ERPs generated in an auditory paradigm. Their findings show that the frequency spectra of both the ERPs and the background EEG signal is $1/f$.

We perform a spectral analysis on the noise $N$ of each session in each dataset to explore the cross dataset variability and find out whether $N$ possesses a $1/f$ spectrum. For each session, we use the least squares method described in [Rivet




(a) Dataset A. Fz: $\alpha = 1$, Pz: $\alpha = 0.8$, P4: $\alpha = 0.7$




(b) Dataset B. Fz: $\alpha = 0.8$, P3: $\alpha = 0.5$, P4: $\alpha = 0.5$

Figure 2.3: Average power spectra of the noise in the signal of the most prominent electrodes of each dataset, across sessions. A simulated $1/f^{\alpha}$ process is also displayed for comparison. Parameter $\alpha$ varies across electrodes and across datasets.

et al., 2011] to estimate the target and nontarget responses $A_t$ and $A_n$ respectively. Then, we use the following model to compute $N$ :

$$X = A_t D_t^{\intercal} + A_n D_n^{\intercal} + N \tag{2.1}$$

$D_t$ and $D_n$ are Toeplitz matrices that allows us to model the ERP distribution in time. Their first column $\mathbf{d}$ is constructed so that the only nonzero elements are $\mathbf{d}_{j_\kappa} = 1$, where $j_\kappa$ are the time samples that correspond to the onsets of $I^T$ target stimuli, $\kappa \in \{1, ..., I^T\}$.

We analyze the power spectral density (PSD) of the signals recorded on three sensors, Pz, Cz and Fz for each dataset. On figure 2.3, we display the average PSD (blue) and the average PSD of a set that contains realizations of $1/f^{\alpha}$ processes (red). For each dataset, the number of pink noise simulations is set to be equal to the number of sessions in the dataset. Initially we observe that the noise on all electrodes in both datasets matches the $1/f^{\alpha}$ process pattern. We then empirically

adapt the slope of the noise using parameter $\alpha$ to match the average PSD of each electrode. These values are displayed on 2.3. Note that they vary both across electrodes and across datasets.

## 2.4 Conclusion

EEG signal variability can be the product of variability in the neural sources of activity, such as peak amplitude and latency variability of the ERP components, variability in the location of sources of ERP and background activity in the brain or intensity of the background activity. Variability can also occur at a sensor level, which might be induced by artifacts such as electrode malfunctions, or sensor placement. These sources of EEG signal variability are observed across different subjects, across different sessions or across different trials within the same session. Their impact on ERP-based BCI is, among others, the degradation of the generalization capacities of existing classification methods [Lotte et al., 2007; Clerc et al., 2016]. EEG signal variability can be quantified through average peak amplitude and peak amplitude variability, average peak latency and peak latency variability, noise energy and signal-to-noise ratio. In the next chapter, we discuss a selected number of adaptive machine learning methods in search for a solution that is tailored to the findings of our variability analysis.

# CHAPTER 3
# TRANSFER LEARNING METHODS

In chapter 2, we analyzed, classified and quantified the sources of EEG variability during the use of an ERP-based BCI. The present chapter presents three transfer learning approaches that have theoretically or experimentally proved efficient against variability. We start by giving a basic an introduction to transfer leaning methods and review some of the existing literature in ERP-based BCI. We then describe each method and provide examples that reflect their efficiency against different types of EEG variability.

## 3.1 Introduction

EEG variability is among the major factors that cause ERP-Based BCI to necessitate calibration before each use. Calibration allows us to acquire the needed training data and rebuild the feature extraction and classification models. As we saw in chapter 1, section 1.4, it is a process that can be tiresome for the user. Our objective is to reduce the need and effort to collect this training data. One possible solution to this issue is transferring the knowledge of existing datasets. This can be achieved through transfer learning.

Transfer learning is a relatively recent branch of machine learning that focuses on the case when the training and the testing sets in a classification problem present differences that hinder the classification task. Transfer learning methods have been the center of recent BCI research [Lotte et al., 2018]. Most of these approaches have shown promising results. Nevertheless, their generalization capacities, while broader than conventional machine learning methods, still encounter some limitations. Having analyzed the sources of variability in chapter 2, we aim to understand these limitations in the light of our findings.

In this chapter, we formally introduce transfer learning and present a short review of existing transfer leaning methods in ERP-based BCI. We proceed to select three frameworks that have theoretically or experimentally proved efficient against specific types of variability. In particular,

1. The Riemannian Geometry framework, which is invariant to affine transformation and could prove effective against scalp topography variability.

2. Optimal transport, which has proven effective against drifts in the feature space and therefore could be robust to peak latency and amplitude variability.

3. Ensemble learning methods, who perform well when the training dataset is noisy.

## 3.2 Background

### 3.2.1 Definition and Notations

Formally, a domain is defined as $\mathcal{D} = \{\mathcal{X}, \mathrm{P}(\mathbf{X})\}$, where $\mathbf{X} = \{x_i\}_{i=1}^{I_n} \subset \mathcal{X}$ denotes a sample of $I_n$ feature vectors, $\mathcal{X}$ is the feature space and $\mathrm{P}(\mathbf{X})$ is the marginal probability distribution of $\mathbf{X}$. A classification task can be defined as a pair $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$ or $\mathcal{T} = \{\mathcal{Y}, \mathrm{P}(\mathbf{Y}|\mathbf{X})\}$, where $\mathbf{Y} = \{y\}_{i=1}^{I_n} \subset \mathcal{Y}$ is the set of labels that correspond to the sample $\mathbf{X}$; $\mathcal{Y}$ denotes the label space; $f : \mathcal{X} \mapsto \mathcal{Y}$ the labeling function which is learned from the pair $\{\mathbf{X}, \mathbf{Y}\}$; and $\mathrm{P}(\mathbf{Y}|\mathbf{X})$ is the conditional probability distribution of the labels.

We typically define two pairs of domain-task: $(\mathcal{D}_s, \mathcal{T}_s)$ and $(\mathcal{D}_t, \mathcal{T}_t)$ , where $s$ and $t$ denote the *source* and *target* respectively. To avoid confusion with the terms "source" and "target" that respectively denote neural sources (or sources of variability) and target stimuli, we will use an emphasized font when we refer to the transfer learning-related terms *source* and *target*. Usually, the *source* labels are known while the *target* labels are unknown. Traditional Machine learning approaches assume that $\mathcal{D}_s = \mathcal{D}_t$ and $\mathcal{T}_s = \mathcal{T}_t$. Transfer learning addresses the following issues:

1. $\mathcal{D}_s \neq \mathcal{D}_t$. This means that either the feature spaces are different, or that the

probability distributions of the *source* and *target* samples are different. The latter is also known as **covariate shift** [Shimodaira, 2000].

2. $\mathcal{T}_s \neq \mathcal{T}_t$. This describes the case when there is a mismatch between the labels due for example to unbalanced class labeling between the *source* and *target*, or when the conditional probability distributions of the labels have changed.

Since transfer learning deals with a family of issues, it offers a family of solutions. Weiss et al. [Weiss et al., 2016] define four general categories with respect to the type of information transferred, each one related to one or more of the issues listed above:

1. Transfer learning through features.
2. Transfer learning through instances.
3. Transfer learning through shared parameters.
4. Transfer learning based on defined relationships between *target* and *source*.

In the following section, we present some of the transfer learning methods that have been applied to BCI.

### 3.2.2 Transfer Learning in ERP-based BCI

In ERP-based BCI, all of the scenarios addressed by transfer learning can occur between *source* and *target* datasets. For example, if the number of electrodes is different between two sessions, we have $\mathcal{X}_s \neq \mathcal{X}_t$. Mistakes from the side of the user lead to differences in the labels between the *target* and the source domain. The non-stationarity of the signal, the effects of ERP variability and the changes in the background brain activity or additive noise all result in covariate shift [Clerc et al., 2016]. Moreover, the *target* dataset is not available immediately. On the contrary, if we consider that the online use of a BCI generates the *target* dataset, it becomes available trial by trial. This results in a large amount of imbalance between *source* and *target* datasets. Out of the solutions proposed by transfer learning in the taxonomy of Weiss et al. [Weiss et al., 2016], the following three have been applied to ERP-based BCI: (i) transfer learning through features, (ii) transfer learning through instances and (iii) transfer learning through shared parameters.

**Transfer learning trough features** can be divided into two approaches. In the first one, one seeks to find a feature subspace where the *source* and *target* domains match. A promising approach that falls into that particular category is the Riemannian Geometry framework. Riemannian Geometry based algorithms were introduced in 2010 by Barachant et al. to classify features in Motor Imagery based BCI [Barachant et al., 2010]. This approach proposes to use covariance matrices as features, which are invariant to affine transformations when manipulated on the Riemannian manifold of symmetric positive definite matrices. This framework has been applied to ERP-based BCI by Congedo et al. and by Barachant et al. in [Congedo et al., 2013; Barachant and Congedo, 2014], where a special form of the covariance matrix is used as a feature. The second family of transfer learning through features focuses on reweighting the features of the *source* domain so that it matches the *target* domain, or vice versa. An example would be the application of a noise reduction spatial filter, trained on the *source* dataset, over *target* data. This approach was employed by Gayraud et al. in [Gayraud et al., 2017], where noise reduction filters are learned over one P300-Speller session and applied on another.

**Transfer learning through instances** mostly apply to the covariance shift problem. Such solutions work by either reweighting the *source* dataset so that it matches the *target* dataset, or by reweighting the *target* dataset so that it matches the *source*. In comparison to transfer learning through features, the weights are particular to each feature vector, instead of each feature. Such solutions have been proposed in the works of [Gayraud et al., 2017; Zanini et al., 2018]. In these works, the authors compute transportation matrices to relocate a *target* dataset so that it matches a *source* dataset.

**Combined use of learned parameters** involves parameters such as classifier weights or distribution priors. Kindermans et al. propose a method in which they combine classification priors over multiple *sources* to train a classifier on the *target* dataset in [Kindermans et al., 2012a] and [Kindermans et al., 2014]. This category also includes ensemble learning methods which have been used in ERP-based BCI to boost classification performance [Rakotomamonjy and Guigue, 2008].

In the following section, we focus on three transfer learning frameworks that have been applied to ERP-based BCI: (i) Riemannian geometry (ii) Optimal trans-

port and (iii) Ensemble Learning. Each comes from a different family of transfer learning solutions and addresses a different domain adaptation issue. In chapter 2, we identified four main sources of ERP-based BCI variability: 1. peak amplitude variability, 2. peak latency variability, 3. scalp topography variability and 4. background noise variability. We describe each transfer learning method and discuss their strengths and limitations in dealing with EEG variability.

## 3.3 Tackling ERP-based BCI Variability

### 3.3.1 Invariant Features using Riemannian Geometry

In BCI, Riemannian Geometry was introduced by Barachant et al. [Barachant et al., 2010] as a transfer learning framework for motor imagery (MI) based BCI. Let $X_i \in \mathbb{R}^{I_c \times I_w}$ be a trial, where $I_c$ denotes the number of electrodes and $I_w$ the number of time samples. In MI-based BCI classification problems, the discriminative information lies in the signal variance and scalp topography. Features such as the log-variance of each electrode in trial $X_i$ can be used to identify a specific activity and turn it into a command [Lotte and Guan, 2011]. These features are however not invariant to affine transformations of the signal. Barachant et al. proposed a feature which is invariant to affine transformation, while containing the same discriminative information as the electrode log-variance [Barachant et al., 2010]. This feature is the spatial covariance matrix $\Sigma_i = \frac{1}{I_w} X_i X_i^\intercal$ of trial $X_i$.

Covariance matrices that have non-zero eigenvalues live on a Riemannian manifold which contains the set of $n \times n$ Symmetric Positive Definite (SPD) matrices $P_n(\mathbb{R})$. Forstner et al. [Förstner and Moonen, 2003] proposed to endow the SPD manifold with the following metric:

$$\|\Sigma\| = \|log(\Sigma)\|_F^2 = \sum_{i=1}^{n} log^2 \lambda_i \,, \tag{3.1}$$

where $\lambda_i$ are the eigenvalues of $\Sigma$. This metric is also known as the Affine Invariant metric. Under the Affine Invariant metric, the structure of the SPD manifold becomes highly regular, bearing much resemblance to a curved vector space. The manifold transforms from a high dimensional cone into a regular and complete manifold of non-positive curvature [Pennec et al., 2006; Pennec, 2009].

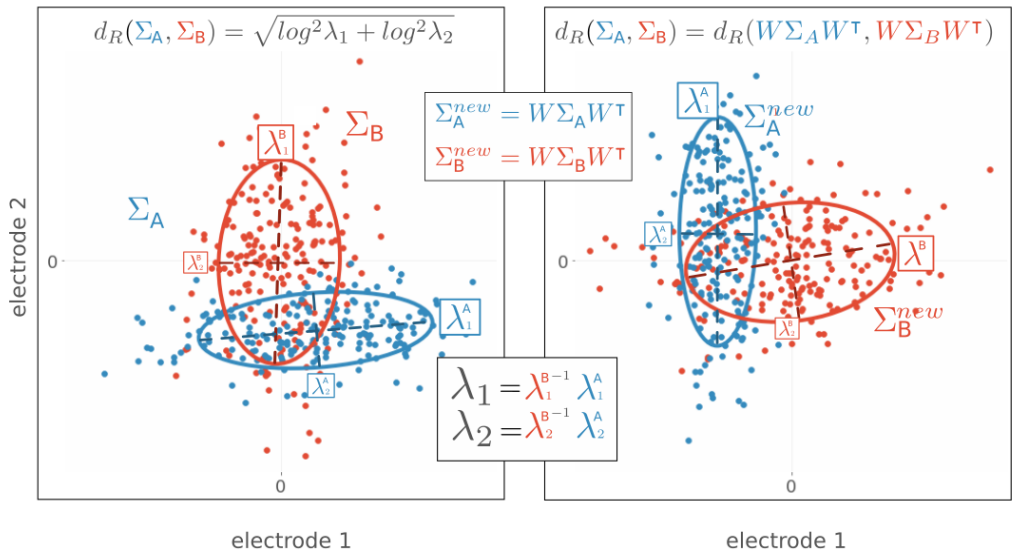



Figure 3.1: An illustrative example of the invariance property of the Riemannian distance for the special case when their covariance matrices commute, i.e. $\Sigma_A\Sigma_B = \Sigma_B\Sigma_A$. Two 2-dimensional signals are plotted, along with their covariances and eigenvalues. When both signals undergo an affine transformation, the ratio of the eigenvalues of their covariance matrices does not change, so the Riemannian distance of the two covariance matrices is the same.

Hence, when covariance matrices are used as features instead of the electrode log-variance,the feature space becomes the SPD manifold, equipped with the following affine invariant distance. Given any matrix $W \in GL_C$ in the General Linear group,

$$\mathrm{d}_R(W\Sigma_A W^\intercal, W\Sigma_B W^\intercal) = \mathrm{d}_R(\Sigma_A, \Sigma_B) = \sqrt{\sum_{i=1}^{C} log^2\lambda_i} \tag{3.2}$$

where $\lambda_i$ are the eigenvalues of $\Sigma_A^{-1}\Sigma_B$. On figure 3.1, we can see an simple illustrative example with two simulated signals $S_A \in \mathbb{R}^{I_c \times I_t}$ and $S_B \in \mathbb{R}^{I_c \times I_t}$ where $I_c = 2$ and $I_t = 200$. This example illustrates the special case when their respective covariance matrices $\Sigma_A$ and $\Sigma_B$ commute, hence $\lambda_i = \lambda_i^A/\lambda_i^B$. Each time sample $s(t) \in \mathbb{R}^{I_c}$ is a random variable $s_A(t) \sim \mathcal{N}(0, \Sigma_A)$ and $s_B(t) \sim \mathcal{N}(0, \Sigma_A)$. The left panel plots the time samples of $S_A$ and $S_B$ for the two simulated electrodes,

as well as the covariance matrices $\Sigma_A$ and $\Sigma_B$. On the right panel, we have applied a transformation on both signals, such that the new signals are $S_A^{new} = WS_AW^\intercal$, $S_B^{new} = WS_BW^\intercal$. Since this transformation does not affect the values $\lambda_1^A/\lambda_1^B$ , $\lambda_2^A/\lambda_2^B$ the distance between $\Sigma_A^{new}$ and $\Sigma_B^{new}$ will remain unchanged as well. This invariance property extends to non-commutative sample covariance matrices as well [Förstner and Moonen, 2003].

The spatial covariance matrix has been emerging as a feature for the classification of mental tasks [Congedo et al., 2013]. Riemannian geometry has become an attractive framework for feature extraction and classification in BCI [Barachant et al., 2013; Gayraud et al., 2016]. In [Barachant and Congedo, 2014], the authors consider that the affine invariant property is what allows for the obtained classification results, under the assumption that cross-session and cross-subject variability can be described in terms of linear transformations. In [Congedo et al., 2015], Congedo et al. demonstrate the significance of affine invariance for BCI classification problems.

Riemannian Geometry has produced two families of classification methods. One where the classification is only based on Riemannian distances, including algorithms such as the Minimum Distance to Riemannian Mean; and a second one which is based on projecting the covariance matrices onto the tangent space of the Manifold. Both of these methods rely on estimating an average covariance matrix, which can be computed on the manifold with the help of Fréchet definition of the mean $\bar{\Sigma} = \arg\min_\Sigma \sum_{i=1}^{I_n} \mathrm{d}_R(\Sigma, \Sigma_i)$, where $I_n$ denotes the number of covariance matrices. Since the SPD manifold has a non-positive curvature, this mean is unique and can be estimated using Newton's gradient descent algorithm [Pennec et al., 2006].

In ERP-based BCI, the difference between the target and nontarget responses lies mainly in their temporal pattern, not in the spatial distribution of the variance. State-of-the art classification methods typically use temporal or spatiotemporal features [Blankertz et al., 2011]. Like the electrode log-variance, these features are not invariant to affine transformations either. To combine the invariance property of the Riemannian framework and the discriminative information lying in the temporal patterns of ERPs, Congedo et al. introduce a special form of covariance matrix [Congedo et al., 2013]. Let $\bar{A}_t = \sum_{i=1}^{I^T} X_i$ and $\bar{A}_n = \sum_{i=1}^{I^N} X_i$ be the estimated average of all trials in the target and nontarget class respectively. $I^T$ and $I^N$ denote

the number of trials in each class. The *extended* trial and the *extended* covariance matrix are then defined as:

$$\tilde{X}_i = \begin{bmatrix} \bar{A}_t \\ \bar{A}_n \\ X_i \end{bmatrix}, \qquad \tilde{\Sigma}_i = \frac{1}{3I_w}\tilde{X}_i\tilde{X}_i^{\intercal}, \qquad \tilde{\Sigma}_i \in \mathbb{R}^{3I_w \times 3I_w} \tag{3.3}$$

Using this feature allows to take advantage of the invariance property in the Riemannian framework, while considering spatial and temporal discriminative information at the same time. If the extended covariance matrix is separated into blocks, the spatial information remains enclosed in the lower right block, which is in fact the covariance matrix $\Sigma_i$. The middle right and top right blocks enclose the temporal correlation of trials to each averaged response.

Within the context of transfer learning, this method allows us to use precomputed average responses in the creation of the extended covariance matrix, while the invariance property of the Riemannian distance renders it immune to linear transformations. These averages can hence be computed from the *source* domain(s). Nevertheless, this feature is sensitive to jitter, i.e. peak latency variability. In their results, Barachant et al. show that the mean performance in terms of Area Under the ROC Curve(AUC) of a Riemannian classifier trained on a session and tested with a different session is equal to 82%. Upon performing experiments with simulated jitter, this performance degrades significantly when the jitter exceeds 50ms [Barachant and Congedo, 2014].

### 3.3.2 Solving Covariate Shift with Optimal Transport

Transport theory studies a problem known as the Monge-Kantorovic transportation problem [Santambrogio, 2015]. This problem can be intuitively understood as the search for the optimal way to transport mass between two probability distributions. The optimization criterion is the minimization of a transportation cost; typically, the cost function represents some metric between the random variables of each distribution. This problem is also known as the optimal transport problem.

Optimal transport has been rising in machine learning as a transport learning approach to solve covariate shift [Courty et al., 2017]. In these approaches, the authors use the discrete optimal transport solution to transform the features of a

labeled *source* domain so that they match those of a *target* domain. Thereby, a classifier can be trained over the labeled *source* features, and used to classify the unlabeled *target* features. A similar approach is used in Gayraud et al. [Gayraud et al., 2017] in a P300 Speller classification problem. The author transport the spatiotemporal features of new, unlabeled data, onto the labeled features of an existing dataset, where a classifier has already been trained. This classifier is then used to label the transported data.

We proceed to describe the discrete optimal transport framework in the particular case of a ERP based-BCI binary classification problem. Let $\mathbf{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{I_n}$ be the set of data acquired during a BCI session. We assume that the data has already undergone preprocessing and we have extracted $d$ relevant features. We thus have a set of $I_n$ extracted feature vectors $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{I_n} \subset \mathbb{R}^d$ coupled with the corresponding labels $\mathbf{Y} = \{y\}_{i=1}^{I_n}$. Furthermore, let $\mathrm{P}(\mathbf{X}) \in \mathcal{P}(\mathcal{X})$ denote the probability distribution from which the samples in $\mathbf{X}$ are drawn, where $\mathcal{X} \subset \mathbb{R}^d$ is a measurable space of dimension $d$ and $\mathcal{P}(\mathcal{X})$ the set of all probability measures over $\mathcal{X}$.

Assume that we have a *source* session and a *target* session, whose respective datasets we denote by $\mathbf{S}^s$ and $\mathbf{S}^t$. We also suppose that we know the labels of the *source* dataset and seek to recover the *target* dataset labels, which are completely unknown. In addition, we assume that the *source* and *target* domains $\mathcal{D}_s = \{\mathcal{X}_s, \mathrm{P}(\mathbf{X}^s)\}$ and $\mathcal{D}_t = \{\mathcal{X}_t, \mathrm{P}(\mathbf{X}^t)\}$ have been subject to covariate shift, i.e. $\mathrm{P}(\mathbf{X}^s) \neq \mathrm{P}(\mathbf{X}^t)$. In [Gayraud et al., 2017], the authors use Optimal Transportation (OT) theory to recover a transport plan between the two probability distributions. Using this transport plan, we can map the *target* domain onto the *source* domain. Then a classifier trained on the *source* dataset can be used to recover the labels of the *target* dataset.

Since we only have a fixed number of samples from each set, the discrete adaptation of our problem boils down to matching empirical measures $\mu_s$, $\mu_t$ of $\mathrm{P}(\mathbf{X}_s)$ and $\mathrm{P}(\mathbf{X}_t)$. In particular, we use the two corresponding empirical distributions $\mu_s = \sum_{i=1}^{I^s} p_i^s \delta_{x_i^s}$ and $\mu_t = \sum_{j=1}^{I^t} p_j^t \delta_{x_j^t}$, where $p_i^s$ and $p_j^t$ are the probability masses associated the *source* and *target* samples respectively, $\delta_x$ denotes the Dirac distribution at location $x$ and $I^s$ and $I^t$ denote the cardinality of the *source* and *target* distributions respectively. Let $\mathbf{p}^s$ and $\mathbf{p}^t$ be the probability vectors of the *source* and *target* datasets and let $\mathbf{1}_I$ denote an $I$-dimensional vector of ones. We compute

the transport plan $\gamma_0$ such that, if $\mathcal{B} = \left\{\gamma \in (\mathbb{R}^+)^{I^s \times I^t} \mid \gamma \mathbf{1}_{I^t} = \mathbf{p}^s, \gamma^\intercal \mathbf{1}_{I^s} = \mathbf{p}^t\right\}$, the transport plan $\gamma_0 \in \mathcal{B}$ is the output of the following minimization problem.

$$\gamma_0 = \arg\max_{\gamma \in \mathcal{B}} \ \langle \gamma, C \rangle_F + \lambda \sum_{i,j} \gamma(i,j) \log \gamma(i,j) + \eta \sum_j \sum_c \|\gamma(\mathcal{I}_c, j)\|_2 \quad (3.4)$$

Matrix $C_{i,j}$ represents the cost of moving probability mass from location $x_j^t$ to location $x_i^s$. In the case of ERP-based BCI, we saw on chapter 1, section 1.3.3 that each feature vector consist of the concatenated rows of the corresponding trial. Using the squared Euclidean distance $\|x_i^s - x_j^t\|_2^2$ is therefore an adequate solution.

The first regularization term allows us to solve this optimization problem using the time-efficient Sinkhorn-Knopp algorithm [Cuturi, 2013]. Since we are performing supervised classification, the second regularization term induces a group-sparse penalty on the columns of $\gamma_0$ ensuring that new samples will give mass only to existing samples of the same class [Courty et al., 2017]. The term $\mathcal{I}_c$ encloses the indices of the rows that correspond to the existing samples of class $c$.

Finally, we compute the new location of the *target* data with barycentric mapping $\hat{\mathbf{X}}^t = \text{diag}(\gamma_0^\top \mathbf{1}_{N_s})^{-1} \gamma_0^\top \mathbf{X}^s$, where $\hat{\mathbf{X}}^t$ and $\mathbf{X}^s$ are matrices whose rows are the feature vectors of the transported *target* dataset and of the *source* respectively. Each *target* feature vector will therefore be transported to the barycenter of those *source* feature vectors it was matched with in $\gamma_0$. A more detailed description of optimal transport is found in chapter 5.

Optimal transport offers a new solution to covariate shift, which often occurs in ERP-based BCI data. We present a simple example of using OT on two simulated datasets. These datasets have been generated in a way that mimics a case where two ERP responses have different peak latencies. For each dataset, we simulate two unbalanced classes. Class 1 has 40 feature vectors and class 2 has 210 feature vectors. These feature vectors consist of five time samples, each one generated according to a normal distribution with a different mean. Class 1 represents the target class and class 2 the nontarget class. For the *source* dataset, i.e. set A, the peak latency is at the second sample, while for the *target* set B it is at the third. We can see the average feature vector of each set and each class on Figure 3.2a.

We compute the probability vectors of each dataset and transport plan $\gamma_0$ which can be seen in Figure 3.2b. The probability vectors are ordered according to the

class in which each feature vector belongs to, starting from class 1. Note that, the transport plan $\gamma_0$ is in fact a joint probability matrix. We can see that the feature

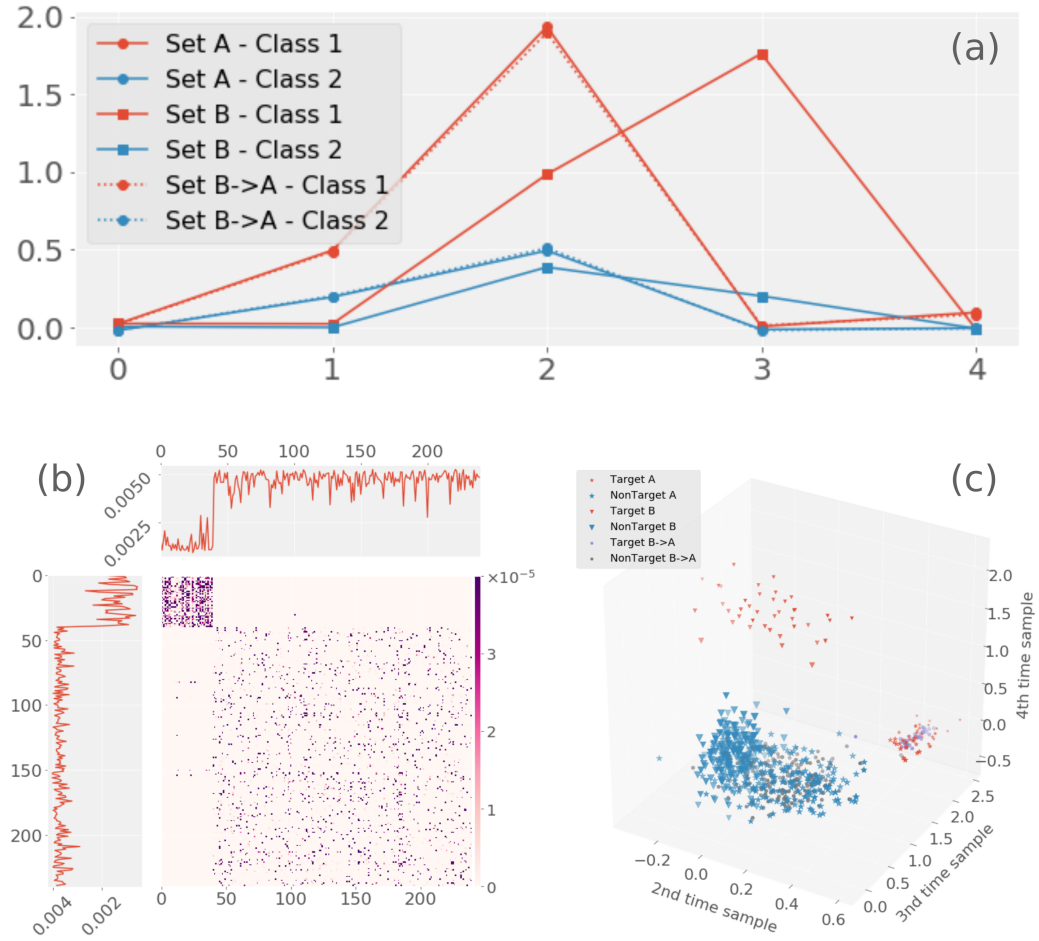



Figure 3.2: Example of an optimal transport on a simulated dataset. (a) we generate two sets of data, each one containing two classes. Class 1 simulates an ERP. The two sets of data have a different peak latency. After transporting the temporal feature vectors of set B onto those of set A, the average peak latency of the two sets is the same. (b) The computed transport plan $\gamma$, along with the marginal probability distributions of the feature vectors of each set, estimated using the Kernel Density Estimation method. (c) A plot of the subspace generated by the second, third and fourth time samples. This subspace corresponds to the feature subspace. After the transport, the feature vectors of set B match the feature vectors of set A, and a classifier trained on set A can now be efficiently used to label them.

vectors of class 1 in the *source* dataset (vertical) are most often associated with the class 1 feature vectors in the *target* dataset (horizontal). On Figure 3.2a we show the average of the *target* dataset, set B, after it has been transported with barycentric mapping onto dataset A. Since the most discriminative features are the time samples 1,2 and 3, we also show a 3d plot where we have plotted on figure 3.2c these particular features for each of the three sets: set A; set B; and the transported set B. We can clearly see the effects of latency variability on the feature space, which causes a shift on the target class (class 1). The mapping computed via OT efficiently maps back the drifted features onto the original location.

Optimal transport is an appealing solution to the covariate shift problem. Nevertheless, there are some limitations to this approach. Assume that the *target* dataset has shifted in such a way that opposite classes are closer to each other. In that case, OT will map the target class of the *target* dataset onto the nontarget class of the *source* dataset, and vice versa. In addition, choosing the regularization parameters is important, especially concerning the entropic regularization term of equation 3.4. High values of $\lambda$ result in denser solutions for $\gamma_0$. Therefore, the barycentric mapping tends to transport features onto the average of the entire dataset.

### 3.3.3 Ensemble Learning: Bagging Classification

If we consider that each session's particular brain activity is noise, classifiers trained over data that comes from a single sessions can be seen as overfitted. Moreover, the same can be said about any feature extraction method that relies on a calibration dataset. Ensemble learning classifiers have been employed several times in BCI to alleviate the effects of overfitting.

One of the ensemble learning methods that effectively avoids overfitting is bootstrap aggregating, or bagging. Bagging uses the technique of bootstrapping to draw samples from a training set with replacement, train one classifier per sample, and use the voting method to predict the outcome. A typical bagging scheme is represented on figure 3.3. For a large number of bootstrap samples, each sample should have $1 - \frac{1}{e} \approx 63.2\%$ of the unique samples in the original set, where $e$ is the base of the natural logarithm, the rest being duplicates [Aslam et al., 2007].

In his introduction of the bagging method, Breiman shows that bagging improves the accuracy of classifiers that do not generalize well [Breiman, 1996]. It is therefore

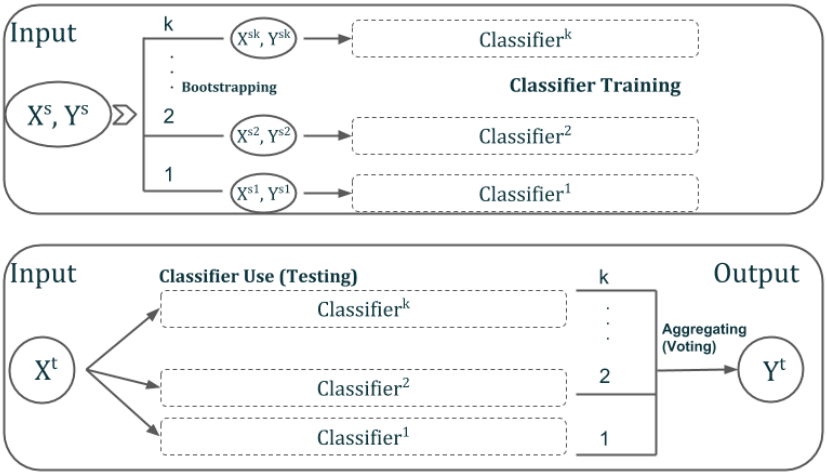



Figure 3.3: A common bagging scheme, representing the training and testing parts of the classification process. During training, k bootstrap samples are created and a respective number of classifiers is trained over each sampel. These classifiers each produce one label for a feature vector of the testing set. The final label is decided after a majority vote.

unsurprising that this method performs well as a meta-classification algorithm in BCI [Sun et al., 2007; Blankertz et al., 2005]. This can be seen for example in the works of Rakotomamonjy et al., where the authors use bagging and train 17 SVM classifiers [Rakotomamonjy and Guigue, 2008]. This method resulted in the best performance of the BCI Competition III. This particular technique has been used in the kaggle BCI Challenge @ NER 2015 by the best performing team [Barachant, 2015; Perrin et al., 2012].

Few researches in BCI have used bagging to train classifiers using a multi-subject or multi-session training set. In [Gayraud et al., 2017], we use bagging on P300-Speller datasets to train LDA classifiers paired with optimal transport. We demonstrate that bagging improves the classification performance. Fazli et al. use a different ensemble learning method, that is, the Adaboost ensemble learning method, in a Motor Imagery-based BCI [Fazli et al., 2009]. Their offline results show that this approach can lead to cross-subject classifiers that perform well.

## 3.4 Conclusion

In the previous chapter, we performed a variability analysis and retrieved parameters and associated values. In this chapter, we have presented three transfer learning methods that we believe can each deal with a variability factor. In the following chapter, will put these methods to the test against each variability parameter, and evaluate their performance and generalization capacities. Our hypothesis is that Riemannian Geometry is effective against scalp topography variability, which will be described as an affine transformation over the source activity of the brain. Optimal transport is a compelling solution to covariate shift, which can be caused by peak amplitude and peak latency variability. Finally, bootstrap aggregating should boost the generalization capacity of a state-of-the-art classifier in the presence of noise.

# CHAPTER 4
# EVALUATION ON SIMULATED EXPERIMENTS

In this chapter, we present a model which allows us to study the sources of variability that were presented in chapter 2 using specific parameters. Then, we simulate a *source* dataset and a number of *target* datasets whereby we modulate the values of these parameters one by one in order to evaluate the transfer leaning methods presented in the previous chapters. At the end of the chapter, we present our results and discuss them.

## 4.1 Introduction

Transfer learning methods have gained popularity within the BCI community since they have been performing well in cross-session and cross-subject classification tasks. However, each family of method has limitations. Transfer learning through features assumes the existence of a common invariant subspace, or a transformation that is common across *target* or *source* domains, which might not be the case. Transfer learning through instances necessitates part of the *target* dataset to be available. In addition, its performances are better when the conditional probability distributions of the *source* and *target* datasets are similar. Transfer leaning through shared parameters depends on the quality of the training data and parameters, and on their degree of similarity to the *target* dataset.

In the previous section, we described three different transfer learning methods, on account of their capacity to generalize, namely, Riemannian geometry, Optimal Transport and Bootstrap Aggregating. Our objective is to quantify the limitations of each method with respect to specific parameters of EEG variability. Having an accurate and realistic model of the EEG signal during the use of an ERP-based BCI is paramount to understanding the effects of variability. Such a model should

allow us to efficiently evaluate how each transfer learning method manages each variability source by allowing us to separately modulate their parameters.

We propose a model that, taking into account both the neural source activity and the recorded signal at the EEG sensors, which are connected through the EEG forward problem [Baillet et al., 2001], incorporates parameters that modulate variability. This model enables simple but realistic simulations of ERP-based BCI experiments. Then, we use this model to produce a number of simulated datasets. We simulate a *source* dataset with fixed parameters and produce *target* datasets by modulating EEG variability parameters. In each set, we only modulate a single EEG parameter and perform classification experiments to evaluate the generalization capacity of the selected transfer learning methods, in terms of classification performance.

## 4.2 Modeling EEG recordings

### 4.2.1 Source Analysis in EEG

To create an EEG model that accounts for the source activity of the brain, we need to understand the relationship between the source space and the sensor space. This matter is studied by the field of EEG source analysis. EEG source analysis boils down to two closely related problems: the inverse problem and the forward problem. The inverse problem in EEG source analysis aims to recover the sources of brain activity given a conductive model of the head and an EEG signal [Baillet et al., 2001]. This approach uses a linear model to describe the contribution of the sources of brain activity to the measured scalp potential. Given a scalp electric potential $x_i(t)$ measured at the i-th EEG sensor at time $t$, the contribution of $I_s$ sources can be modeled as:

$$x_i(t) = \mathbf{g}_i^\intercal \mathbf{s}(t), \quad \mathbf{g}_i, \mathbf{s}(t) \in \mathbb{R}^{I_s}, \tag{4.1}$$

where $\mathbf{g}_i$ is a vector representing the contribution of each source in the vector $\mathbf{s}(t)$ at time $t$. For $I_c$ EEG sensors and $I_t$ time samples, the matricial form of the model becomes

$$X = GS, \quad X \in \mathbb{R}^{I_c \times I_t}, G \in \mathbb{R}^{I_c \times I_s}, S \in \mathbb{R}^{I_s \times I_t} \tag{4.2}$$

$G$ is known as the gain matrix or the forward model [Baillet et al., 2001]. The EEG forward problem is therefore concerned with the computation of $G$, which necessitates a conductivity model of the human head. Typically, we model the head as several tissues with different conductivities; for example, brain, skull, scalp. The forward problem can be formulated as a quasistatic approximation of Maxwell's equations [Hämäläinen et al., 1993]. In general, this equation does not have an analytic solution for realistic head models. Numerical methods are needed to solve it, such as the Finite Elements Method (FEM) or the Boundary Elements Method (BEM). The FEM is based on volumic discretization, while BEM only needs surface meshes between different tissue [Wolters et al., 2004; Sarvas, 1987; Kybic et al., 2005].

### 4.2.2 Existing EEG Signal Models and Beyond

Several models have been proposed for the recorded EEG activity during the use of an ERP-based BCI [Blankertz et al., 2011; Rivet et al., 2009, 2011]. The most common model is based on the forward EEG model of equation (4.2) plus a term $N$ that encloses any on-going activity that is not time-locked to the evoked activity, as well as artifacts and additive noise [Blankertz et al., 2011].

$$X = GS + N, \quad G \in \mathbb{R}^{I_c \times I_s}, S \in \mathbb{R}^{I_s \times I_t}, N \in \mathbb{R}^{I_c \times I_t}, \tag{4.3}$$

In their works, Rivet et al. [Rivet et al., 2009] propose a linear model for EEG measurements that arise from the use of a P300-Speller. This model is based on the knowledge that target stimuli elicit a P300 component. Let $A_t \in \mathbb{R}^{I_c \times I_w}$ be the archetype of the scalp distribution of a P300 component over time, where $I_w$ is the number of samples in a specific time window that immediately follows stimulus onset. The resulting EEG signal can be modeled as

$$X = A_t D_t^\intercal + N, \quad D_t \in \mathbb{R}^{I_t \times I_w}, N \in \mathbb{R}^{I_c \times I_t}, \tag{4.4}$$

This model uses what we will from now on refer to as a *distribution matrix* $D_t$ to model the distribution of the archetype target response $A_t$ in time. In particular, this $\mathbb{R}^{I_t \times I_w}$ matrix is a Toeplitz matrix whose first column $\mathbf{d}$ is constructed so that $\mathbf{d}_{j_\kappa} = 1$, where $j_\kappa, \kappa \in \{1, ..., I^T\}$ are the time samples that correspond to the onsets of $I^T$ target stimuli. The authors incorporate a second term in [Rivet

et al., 2011] that describes the template activity produced by the nontarget stimuli. Equation (4.4) thus becomes

$$X = A_t D_t^\intercal + A_n D_n^\intercal + N, \quad D_t, D_n \in \mathbb{R}^{I_t \times I_w}, N \in \mathbb{R}^{I_c \times I_t}, \tag{4.5}$$

The same concept is used in the generation of matrix $D_n$, which distributes the nontarget response in time. Souloumiac et al. propose a similar model whose second term models the response to every stimulus, instead of only the nontarget stimuli [Souloumiac and Rivet, 2013]. In the same research, Souloumiac et al. modify the distribution matrix $D_t$ to account for the P300 latency variability.

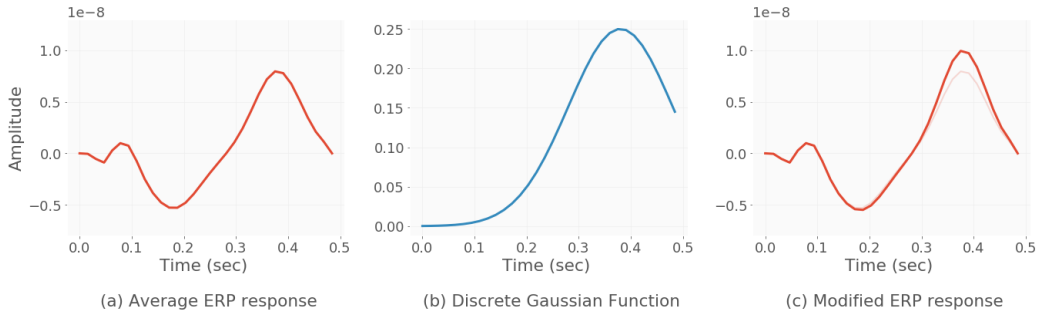
Using distribution matrices permits us to model the diffusion of any response that is time-locked to a stimulus. Hence, if we combine these approaches, we can model the recorded EEG signal during an ERP-based BCI experiment as the diffusion of the target response, the nontarget response and the sensory response $A_f \in \mathbb{R}^{I_c \times I_w}$ to the stimulus as:

$$X = A_t D_t^\intercal + A_n D_n^\intercal + A_f (D_t^\intercal + D_n^\intercal) + N \tag{4.6}$$

### 4.2.3 Modeling Trial-to-Trial Variability

**Peak Latency** Let $\mathbf{d}$ denote the first column of $D_t$, constructed so that $\mathbf{d}_{j_\kappa} = 1$. $\mathbf{J} = \{j_\kappa\}_{\kappa=1}^{I_t}$, $j_k \in \{1, \cdots, I^T\}$, is the set of time samples that correspond to the onsets of $I^T$ target stimuli. Peak latency variability can be modeled by modifying each $j_\kappa$ by a value $\delta^\kappa \in \mathbb{Z}$, for each target response. Then, given a set $\Delta = \{\delta^\kappa\}_{\kappa=1}^{I_t}$ and a distribution matrix $D_t$, set $\mathbf{J}$ becomes $\tilde{\mathbf{J}} = \{\tilde{j}_\kappa\}_{\kappa=1}^{I_t}$, where $\tilde{j}_\kappa = j_\kappa + \delta^\kappa$ and a new matrix $\tilde{D}_t$ can be constructed so that $\tilde{\mathbf{d}}_1(\tilde{j}_\kappa) = 1$.

**Peak Amplitude** Peak amplitude variability can also be modeled through the distribution matrix $D_t$. For simplicity, let us consider that the archetype target ERP response in the brain can be modeled as a single discrete signal $\mathbf{s} \in \mathbb{R}^{I_w}$ of length $I_w$. Let $\mathbf{s}_n$ denote the signal amplitude at time sample $n$, $n \in \{1, \cdots, I_w\}$, and $n_p$ the time sample that corresponds to the peak latency. Let $\mathbf{s}^\kappa \in \mathbb{R}^{I_w}$ denote the ERP response to the $\kappa^{th}$ target stimulus. If we denote the average ERP peak amplitude as $\bar{s}_{n_p}$, then we can write the peak amplitude of the $\kappa^{th}$ response to the




(a) Average ERP response (b) Discrete Gaussian Function (c) Modified ERP response

Figure 4.1: An example of introducing peak amplitude variability to an archetype ERP. (a) The archetype ERP $s$(b) The Gaussian function $g(n)$ (c) The result of increasing the peak amplitude of $s$ by applying $s^n = s^n(1 + g(n))$.

target stimulus as a function of a value $a^\kappa \in \mathbb{R}$ that modulates the average peak amplitude multiplied by $\bar{s}_{n_p}$ so that $s^\kappa_{n_p} = a^\kappa \bar{s}_{n_p}$.

Let $\mathbf{d}_\kappa = (d_{\tilde{j}_\kappa,1}, d_{\tilde{j}_\kappa+1,2}, \cdots, d_{\tilde{j}_\kappa+I_w-1,I_w})$ denote the vector that contains the non-zero elements of matrix $\tilde{D}_t$ that correspond to the $\kappa^{th}$ target stimulus. These coefficients, which are all equal to 1, will be by construction multiplied to the average ERP response $\mathbf{s}$, giving $\mathbf{s}^\kappa = (d_{\tilde{j}_\kappa,1}s_1, d_{\tilde{j}_\kappa+1,2}s_2, \cdots, d_{\tilde{j}_\kappa+I_w-1,I_w}s_{I_w})$. We add to each coefficient $d_{\tilde{j}_\kappa,\mathbf{s}_n}$ the corresponding coefficient the following discrete Gaussian function: $g_\kappa(n) = (a^\kappa - 1)e^{-\frac{n-p_n}{\sigma}}$, where $a^\kappa$ denotes the peak amplitude of the $\kappa^{th}$ target trial and $\sigma$ the kernel bandwidth. By choosing an adequate value for $\sigma$, such as, the bandwidth of the average ERP response $s$, this technique allows us to obtain the desired result: The peak amplitude of the response to the $\kappa^{th}$ stimulus is rendered equal to $s^\kappa_{n_p} = a^\kappa \bar{\mathbf{s}}_{n_p}$, while the signal $s^\kappa$ follows the peak amplitude change in a smooth manner.

An example of this process is presented on figure 4.1. Figure 4.1a displays the average source ERP response $s$, with $I_w = 32$ time samples. The peak latency of $s$ is equal to 0.38 ms, therefore $n_p = 24$, and the peak amplitude $s^{n_p} = 8nA \cdot m$. Suppose that the $\kappa^{th}$ trial has peak amplitude $s^\kappa_{n_p} = 10nA \cdot m$. In this case, $a_k = 0.25$, and the corresponding function $g_\kappa(n)$ is displayed on figure 4.1b. Multiplying each coefficient $s^n$ with $1 + g(n)$ produces the response of figure 4.1c.

### 4.2.4 An EEG Model for ERP-Based BCI

We can now formally introduce our model of the EEG signal $X$ that is generated during an ERP-Based BCI experiment.

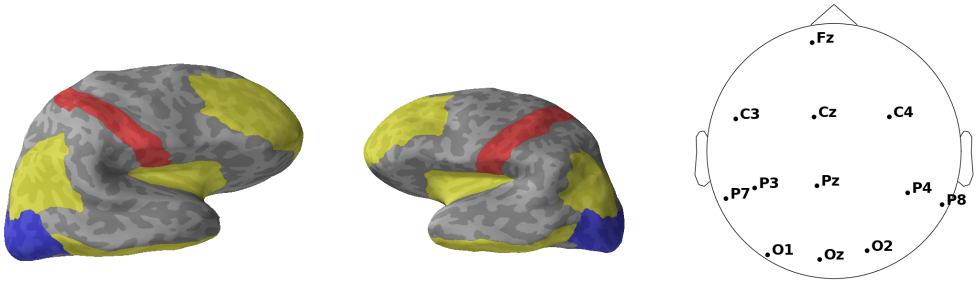
$$X = G(S_t\mathbf{D}_t^\intercal + S_nD_n^\intercal + S_f(D_t + D_n)^\intercal + N_b) + N_a \tag{4.7}$$

$G \in \mathbb{R}^{I_c \times I_t}$ denotes the gain matrix; $S_t \in \mathbb{R}^{I_s \times I_w}$, $S_n \in \mathbb{R}^{I_s \times I_w}$ are the simulated target and nontarget archetype responses in the source space respectively; $S_f \in \mathbb{R}^{I_s \times I_w}$ corresponds to the archetype evoked response to the flashes; $N_b \in \mathbb{R}^{I_s \times I_t}$ is the background activity in the brain; and $N_a \in \mathbb{R}^{I_c \times I_w}$ the noise which is uncorrelated to the activity in the sources. With respect to equation 4.4, $A_t = G(S_t + S_f)$, $A_n = G(S_n + S_f)$ and $N = GN_b + N_a$. We can see that there is a clear correspondence between what we measure in the sensor space and what we can simulate in the source space.

Through matrix $\mathbf{D}_t$, our model takes into account both inter-session and intra-session variability. It allows for the simulation of the following different sources of variability:

1. **Peak latency and amplitude variability**. Peak latency and amplitude variability can be modeled both across different sessions, and within the same session. Given two signals $X^a$ and $X^b$ that correspond to two different sessions, we can produce two different ERP responses $S_t^a$ and $S_t^b$ with different peak latencies and peak amplitudes. Within the same session, we can modify the diffusion matrix $D_t$ by producing a set $\Delta$ as a set of random variables $\delta^\kappa$ whose expected value is $E[\delta^\kappa] = 0$, which implies that $E[\tilde{j}_\kappa] = j_\kappa$. Given a distribution matrix $\tilde{D}_t$ and a set of coefficients $\mathbf{A} = \left\{a^\kappa\right\}_{\kappa=1}^{I_t}$, we can construct a new matrix $\mathbf{D}_t$ that models trial-to-trial amplitude variability as well.

2. **Background activity and additive noise.** EEG recordings during ERP-Based BCI experiments contain background brain activity and additive noise, which is enclosed in a single noise term $N$. The noise term $N$ can be decomposed into two different terms, namely $N = GN_b + N_a$, where $N_b \in \mathbb{R}^{I_s \times I_t}$ is the background activity in the brain; and $N_a \in \mathbb{R}^{I_c \times I_t}$. Both terms $N_a$ and $N_b$ can be modified accordingly to generate inter-session noise variability. the noise which is uncorrelated to the activity in the sources.




(a) Lateral view of the brain and the selected regions.

(b) Medial view of the brain and the selected regions.

(c) Sensor positions on the scalp.

Figure 4.2: Source regions and and electrode positions of the simulated datasets.

3. **Scalp topography variability.** Scalp topography variability can be modeled across sessions though matrix G, which by construction can either model a change in the source space or a change in the sensor space, such as the a change of sensor placement. Given two forward models $G^a$ and $G^b$, we can therefore produce two signals $X^a$ and $X^b$ with different scalp topography.

## 4.3 Simulation of a P300 Speller Experiment

### 4.3.1 Neural Source Simulation and Experiment Parameters

Using the Sample dataset provided by MNE [Gramfort et al., 2014] and the MNE-python toolbox [Gramfort et al., 2013], we compute a forward model $G$ with 12 sensors and 34 sources, using the head model of figure 4.2. We select the target and nontarget sources in the brain according to the findings of Bledowski et al. [Bledowski, 2004]. The authors of this work identify the following six bilateral pairs of source regions of responses to target stimuli: the prefrontal cortex; precentral sulcus; inferior parietal lobe; posterior parietal cortex; inferior temporal cortex; and anterior insula. We place an equivalent amount of sources of activity within these regions at random locations, as seen in figures 4.2a, 4.2b. Since the activity on the inferior temporal cortex and the inferior parietal lobe was found to have a higher amplitude, we add two sources on each hemisphere for each one of these regions. These 16 sources simulate the ERPs of figure 4.3a for the target and nontarget

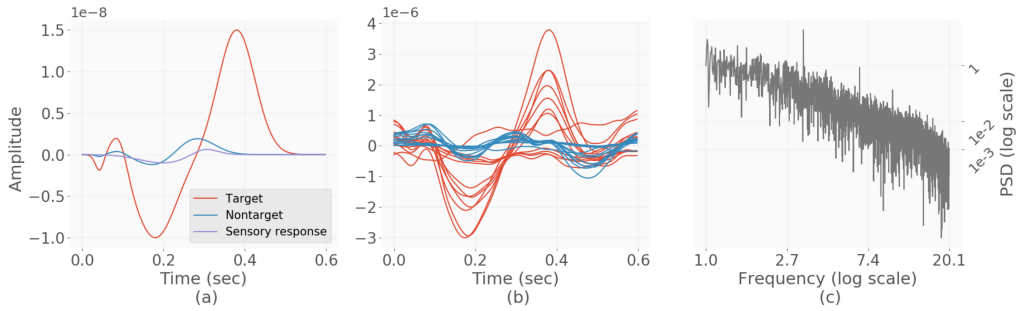



Figure 4.3: Simulated *source* signals. (a) The archetype target, nontarget and sensory responses. (b) Average across all target trials (red) and nontarget trials(blue) of the resulting simulated EEG signal, for all electrodes. (c) Spectrogram of $G(D_t + D_n)S_f + GN_a + N$. The peak that appears about 3 Hz corresponds to the sensory response to the stimulations. It does not appear in figure 2.3, since the estimated average target and nontarget responses in that figure enclose the sensory response (see chapter 2, section 2.3.2).

classes.

We also add 2 sources of activity in the right occipital lobe and another 2 sources in the left occipital lobe that produce the source activity of figure 4.3. The activity in these sources is time locked to every simulated stimulus and represents the evoked potential in response to the flashes. Then, we add 16 additional sources of background activity for $N_b$ which we simulate as pink noise. Note that $N_b$ is not time locked to the simulated stimuli. We choose to add these sources near the parietal lobe (Red region on Figure 4.2), according to the works of [Bledowski, 2004], in which the fMRI activation maps show an increase of the blood oxygenation level-dependent signal. Finally, we add an additional source of noise $N_a$ which is uncorrelated to the sources, in the form of white noise.

All our simulated P300 speller experiments use the same parameters as the experimental dataset (chapter 2, section 2.3.1). The flash interval is equal to 300 ms and the target/nontarget ratio is equal to 1/5, i.e. in a sequence of 6 events, 5 are nontarget and one is target. The sampling frequency is equal to 64 Hz and the total duration of the simulated experiment is equal to 5 minutes.

### 4.3.2 Simulating Variability

Our objective is to evaluate how each one of the transfer learning methods described in chapter 3 (section 3.3) performs against different sources of variability. To that end, we use our EEG model described in equation (4.7) and the parameters described in the previous section to simulate one *source* domain $\mathcal{D}_s$ and several *target* domains $\mathcal{D}_t$. Each *target* domain is identical to the *source* domain, except for one of the following parameters: 1. average peak amplitude and trial-to-trial peak amplitude variability; 2. average peak latency and trial-to-trial peak latency variability; 3. the spectral energy density of the background activity signal; and 4. the SNR resulting from the additive noise power.

In chapter 2, section 2.3 we compute the average and standard deviation of both peak amplitude and peak latency for every session in datasets A and B. We find that the average peak amplitude varies between 1.5 and 9 $\mu$V, while the average peak latency varies between 360 and 400 ms. In accordance to those findings, we simulate our *source* dataset using the signals[1] $\mathbf{s}^t, \mathbf{s}^n$ and $\mathbf{s}^f$ of figure 4.3a. The average peak amplitude of $\mathbf{s}^t$ is set equal to $s^t_{n_p} = 8nA \cdot m$, which for the given forward model $G$ results in a measured average peak amplitude that is equal to $2\mu V$ in the sensor space (Figure 4.3b). The average peak latency is set equal to 380 ms, which for a sampling frequency equal to 64 Hz gives $n_p = 24$. Trial-to-rial variability is modeled through the diffusion matrix $\mathbf{D}_t$. We chose the sets $\mathbf{A}$ and $\Delta$ such that each element $a^\kappa \in \mathbf{A}$ is a random normal variable $a^\kappa \in \mathcal{N}(0, \sigma_{amp})$ and each element $\delta^\kappa \in \Delta$ is a random normal variable $\delta^\kappa \in \mathcal{N}(0, \sigma_{lat})$. For the *source* dataset, $\sigma_{amp} = 0$ and $\sigma_{lat} = 0$, which implies that the *source* dataset does not present trial-to-trial amplitude and latency variability.

The background activity is modeled as a $1/f^\alpha$ process in the following way: First, we model a white noise process $n$, where each time sample is drawn from a Gaussian distribution, $n(t) \sim \mathcal{N}(0, \sigma_b)$. We proceed to apply a $1/f^\alpha$ frequency filter on the signal $n(t)$ in order to generate a pink noise process $\mathbf{n}^b$. For simplicity, we set $\alpha = 1$. For the *source* domain, the value of $\sigma_b$ was chosen empirically to match the average background activity with the lowest spectral energy density among the sessions of the experimental datasets (chapter 2, section 2.3). The resulting signal $\mathbf{n}^b$, seen on figure4.3c, has a spectral energy density equal to

[1] The code that generate these signals is available in the MNE-python toolbox

$\sim 10^{-6}$ for $\sigma_b = 10^{-9}$.

The rows of matrix $S_t$ that correspond to the selected target source in the brain are set equal to $\mathbf{s}^t$, while the remaining rows are equal to zero. Matrices $S_n$, $S_f$ and $N_b$ are generated from $\mathbf{s}^n$, $\mathbf{s}^f$ and $\mathbf{n}^b$ in the same way. Finally, we simulate the additive noise as a white noise process, i.e. a multivariate signal $N_a$ whose time samples are drawn from a multivariate Gaussian distribution $n_a(t) \sim \mathcal{N}(0, \sigma_{add} I)$. We wanted the *source* dataset to contain data with a relatively high SNR. Therefore, with respect to the white additive noise, the simulated SNR is equal to $\sim 30$ dB.

Then, we generate a number of *target* datasets in four different classes of experiments, with respect to a specific source of variability. Each time, we modulate a single parameter and generate 100 signals.

***Peak amplitude*** We model intra-session variability by increasing the average peak amplitude $s^{n_p}$ from 8 to to 12 $nA \cdot m$ with a 2 $nA \cdot m$ step. Note that, the measured peak amplitude in the sensor space resulting from a 12 $nA \cdot m$ peak amplitude at the target sources is equal to $\sim 8\mu V$, which corresponds to the maximum measured peak amplitude for dataset B. For each step in the peak amplitude increase, we modulate an increase of the trial-to-trial peak amplitude variability by setting $\sigma_{amp}$ to consecutive values between $0nA \cdot m$ until $2nA \cdot m$, with a step of $0.5nA \cdot m$. This results in 25 sets of 100 simulated datasets.

***Peak Latency*** We simulate experiments with three values for the average peak latency $n_p$ to model cross-session variability, according to the measured average peak latency of the experimental dataset: 360ms; 380ms; and 400ms. For each value, we increase the trial-to-trial peak latency variability by setting $\sigma_{lat}$ to consecutive values between 0 ms and 250 ms with a 50 ms step. This results in 25 sets of 100 simulated datasets.

***Background Activity*** We produce 16 values of $\sigma_b$ in the interval $\sigma_b \in [10^{-9}, 10^{-7}]$ in order to generate signals $\mathbf{n}^b$ with increasing spectral energy density. This results in 16 sets of 100 simulated datasets with these parameters.

***Additive Noise*** We produce 20 values of $\sigma_{add}$ in the interval $\sigma_{add} \in [10^{-7}, 40^{-6}]$ in order to generate signals with decreasing SNR. For the *source* dataset, $\sigma_{add} = 10^{-7}$. This results in 20 sets of 100 simulated datasets with these parameters.

***Forward Model*** We perform a second round of the all the experiments listed above to generate a second set of *target* datasets in order to model scalp topography variability. To this end, we place the neural sources described in section 4.3.1 in

different locations within the same regions, and generate a second forward model $\tilde{G}$, which is used in the second round of experiments.

## 4.4 Results

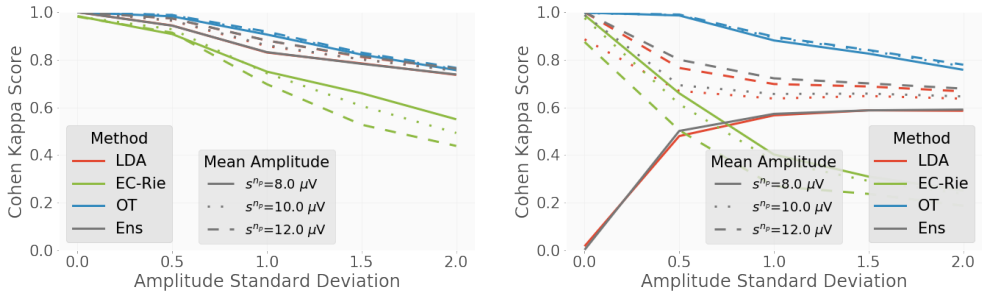
### 4.4.1 Classification Pipelines and Performance Measures

Each simulated signal is filtered with a bandpass 3rd order Butterworth filter between 0.1 and 20Hz, and decimated by a factor of 4. It is then segmented into 0.6 second trials $X_i$ starting from stimulus offset. We present the performances of three classification method, each one corresponding to one of the transfer learning methods described in chapter 3, section 3.3. Each classification method is trained on the *source* dataset and tested on each one of the *target* datasets.

The first classification method is a Riemannian classification algorithm known as the Minimum Distance to Riemannian Mean (MDRM). Presented by Barachant et al. in [Barachant et al., 2010] to classify features in Motor Imagery based BCI, this method uses the sample covariance matrix $\Sigma_i$ of a trial $X_i$ as a feature, and estimates the centroid of each class in the training set by calculating the Riemannian mean of all the class features. For each new feature, its Riemannian distance to all centroids is calculated, and the smallest among these distances defines the winning class. Since we have simulated a P300 experiment, we use the extended covariance matrix $\tilde{\Sigma}_i$ as a feature, presented in chapter 3, section 3.3.1 and call this method EC-Rie.

The second classification method, labeled OT, is based on Optimal Transportation theory. Initially, we train a Linear Discriminant Analysis (LDA) classifier over the *source* dataset using spatiotemporal features. Then, for each *target* dataset, we compute the transport plan using a squared Euclidean cost between the *target* and *source* domains and transport the *target* feature vectors onto the *source* domain. The entropic regularization parameter is equal to $\lambda = 0.001$; it is chosen to be as small as possible, so that the transport plan $\gamma$ is sparse. Our preliminary experiments (not displayed here) showed that the results of OT are robust with respect to the value of the second regularization parameter, which is set equal to $\eta = 0.1$. The transported feature vectors are classified with the trained LDA classifier.

The third classification method is an bagging LDA classifier trained over spa-




(a) *Source* and *target* simulations use the same forward model.

(b) *Source* and *target* simulations use a different forward model.

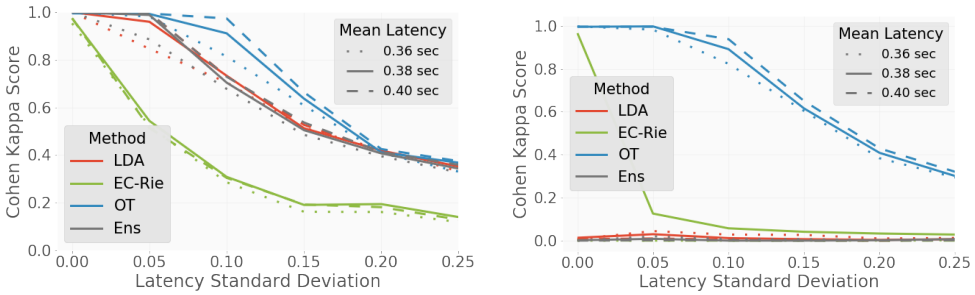
Figure 4.4: Performances of the three proposed transfer learning classifiers and the LDA classifier as we increase peak amplitude trial-to-trial variability. Each classification method is denoted by a different color. For each method, the different lines correspond to different average amplitudes, which is how we model cross-session variability.

tiotemporal features. We create 50 bootstrap samples from the *source* dataset and train 50 LDA classifiers. For comparison purposes, we also train a single LDA classifier over the entire *source* dataset.

Since the classes in each dataset are unbalance, we evaluate the outcome of each method using Cohen's kappa as a performance metric, as proposed by Thomas et al. in [Thomas et al., 2013]. Cohen's kappa is defined as $\mathrm{k} = 1 - \frac{1-acc}{1-p_{ch}}$, where $acc$ is the classification accuracy, and $p_{ch}$ is the hypothetical probability of chance agreement. For a binary classification problem, $p_{ch} = \frac{1}{I^2}((TP+FN)(TP+FP)+(TN+FP)(TN+FN))$, where $I$ denotes the total amount of trials in the *target* dataset and $TP, TN, FP$, and $FP$ denote the true positives, true negatives, false positives and false negatives respectively. Cohen's kappa takes values between -1 and 1, with 0 being the chance level.

### 4.4.2 Amplitude variability

Figure 4.4 displays the results obtained when we modulate amplitude variability. On figure 4.4a, we can see that, for the same head model, the most robust methods are OT, Ens and the simple LDA classifier. Since the SNR in these experiments is high, the Ens and LDA have an almost identical performance. EC-Rie also performs well when the amplitude standard deviation $\sigma_{amp}$ takes low values; the




(a) *Source* and *target* simulations use the same forward model.

(b) *Source* and *target* simulations use a different forward model.

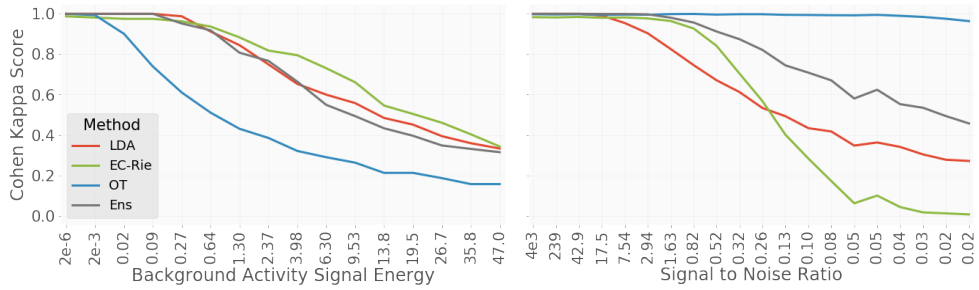
Figure 4.5: Performances of the three proposed transfer learning classifiers and the LDA classifier as we increase peak latency trial-to-trial variability. Each classification method is denoted by a different color. For each method, the different lines correspond to different average latencies, which is how we model cross-session variability.

performance decrease for higher amplitude standard deviations can be attributed to the fact that amplitude variability transforms only one of the two responses, i.e. the target response. Therefore the invariance property does not hold any longer.

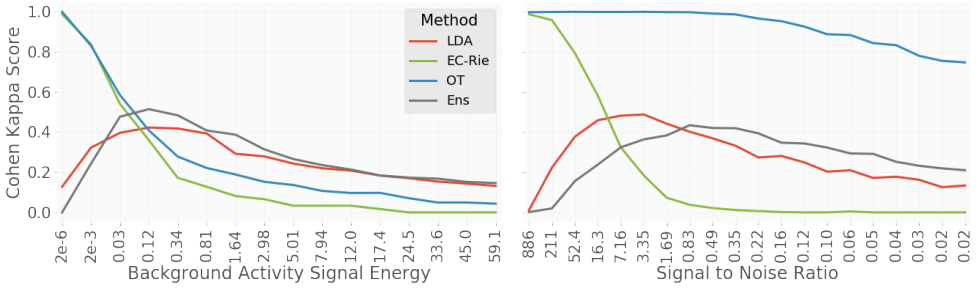
For the second forward model, the performance of all classifiers except for OT and EC-Rie are greatly changed. The simple LDA classifier only works well for high mean amplitude values, a behavior which is also reflected in the performance of the bagging classifier. This is due to the fact that we only change the mean amplitude of the target class, therefore the LDA features end up producing classes with a higher separability. When the amplitude is the same for both *target* and *source* domains, both methods classify every trial as a nontarget trial, which is why the classification performance is equal to the chance level.

### 4.4.3 Latency variability

On figure 4.5 we present the results of our experiments when we modulate the average peak latency $n_p$ and the trial-to-trial variability through parameter $\sigma_{lat}$. Concerning the results of the experiments when the forward model between *source* and *target* is the same, we can see a performance deterioration on figure 4.5a for all classification methods as the trial-to-trial peak latency variability increases. As expected, the EC-Rie method performs less well than all others, since it mostly de-




(a) *Source* and *target* simulations use the same forward model.




(b) *Source* and *target* simulations use a different forward model.

Figure 4.6: Performances of the three proposed transfer learning classifiers and the LDA classifier as we increase the noise. On the left, we show the effect of increasing the signal energy of the pink noise process that simulates background activity. On the right, we display the classification performances as a function of the SNR, which is decreased as we increase the standard deviation of the white additive noise. Each classification method is denoted by a different color.

pends on the correlation of each trial to the archetype target and nontarget responses. For a different forward model, we can see on figure 4.5b that only OT performs well, and its performance deteriorates as the trial-to-trial latency variability increases.

### 4.4.4 Background activity and noise variability

On figure 4.6, we can see the results of the experiments in which we modulate the background activity $N_b$ and those in which we modulate the additive noise $N_a$. Increases in the energy of the pink noise greatly affects OT for both the same forward model and for a different forward model. On the other hand, OT seems not at all affected by decreases in the SNR that originate from additive white

noise, which seems to mostly affect the EC-Rie method. While the LDA and the Bagging classifiers have a similar performance when the pink noise energy increases, the bagging classifier is more robust to additive white noise. When the forward model is different between *source* and *target*, the Ens and LDA classifiers seem to surprisingly perform better as the pink noise energy increases and as the SNR decreases. This is discussed in the next section.

## 4.5 Discussion and Conclusion

In this chapter, we have evaluated, through simulated experiments, how three different transfer learning methods respond to EEG signal variability. In particular, we were interested in the following factors: (i) average peak amplitude; (ii) peak amplitude standard deviation; (iii) average peak latency (iv) peak latency standard deviation; (v) $1/f$ noise; (vi) additive white noise; and (vii) forward model.

The correspondence between these factors and the observed variability is justified. Indeed, the first four are immediately connected to ERP variability, either across sessions, or within the same session [Polich and Kok, 1995; Polich, 2009]. Background activity in the form of a neural source signal $N_a$ can also be connected to ERP variability. ERP variability and BCI performance are both modulated by psychophysiological mental states of the users [Polich and Kok, 1995; Polich, 2009; Jeunet et al., 2016]. According to the results of [Ward, 2002], it is reasonable to model this modulation using a simulated $1/f^{\alpha}$ process. Additionally, signal artifacts or physiological signals that do not contribute to the forward model have often been modeled as additive white noise [Rivet et al., 2009; Blankertz et al., 2011; Rivet et al., 2011]. Finally, scalp topography variability is evidently connected to the forward model.

The results of our simulations highlight the generalization capacities of three transfer learning methods though the implementation of three corresponding classification algorithms: (i) EC-Rie for the Riemannian geometry framework; (ii) OT for the optimal transport framework; and (iii) Ens for the ensemble classification framework. For the most part, our findings align with our original hypothesis over each method.

Thus, we see that the Riemannian geometry framework is robust to modification of the forward model. However, in the particular case of ERP-Based BCI, the

classification algorithms rely on the correlation of the signals in each trial to the archetype target and nontarget signals. Modulating a second parameter at the same time, especially cross session peak latency variability, causes the performance of the classifier to degrade. On figure 4.5b, only the experiments where the *target* dataset has the same average latency as the *source* perform well for low values of trial-to-trial latency variability. However, for the same forward model, the method is only affected by trial-to-trial latency variability. EC-Rie is also affected by white additive noise. This can be explained using the mathematical formulation of the Riemannian distance.

Optimal transport is robust to modulations of the peak amplitudes, decreases in the SNR due to the additive noise and differences in the forward model. In fact, all these variability sources cause a drift in the target domain that OT can easily revert, since they neither affect the conditional probability distribution, nor induce changes in the target domain that result into misclassifications. Nevertheless, peak latency variability, especially when it is modulated across trials, caused OT to perform badly. In addition, the background activity noise signal energy increase greatly degraded the performance of the method.

Unsurprisingly, the bagging classifier performed better than the simple LDA classifier only when the SNR decreased due to the high variance of the additive noise, especially when the forward model is the same.

An interesting remark can be made upon observing the classification performance of the LDA classifier, which is conceived as a baseline. Indeed, for different parameters of variability, some transfer learning methods perform worse, a phenomenon known as negative transfer [Weiss et al., 2016]. For trial-to-trial peak amplitude and latency variability under the same forward model, this is the case for the Riemannian geometry classifier. The same result for a different forward model is only observed in the case of trial-to-trial peak amplitude variability. In the case of background activity, it is OT that induces negative transfer when the forward model is the same, and EC-Rie as well when the forward model is different. When the additive noise increases, EC-Rie once more induces negative transfer.

In the particular case of modulating the noise parameters for different forward models, an interesting observation can be made on figure 4.6b. As the signal energy or SNR increase, the performances of the LDA and Ens classifiers increase until a certain point and decrease again. This is the effect of two phenomena. First the

classification weights of LDA project the feature vector onto a 1-d space where the two classes are separable. For different head models, this projection causes the features to fall on the wrong side of the hyperplane (which in this particular scenario is a scalar). This explains the zero kappa score: everything is classified as nontarget. The two classes in the *target* dataset are still separable; moreover, the feature vector variability about the class center is low. Hence, when the noise signal variability increases, the variability of the two classes increases with it. This initially causes some of the target feature vectors to fall on the correct side of the hyperplane. As the class variability keeps increasing, some of the nontarget feature vectors find themselves in the target side, which causes the variability to decrease again.

Overall, these results indicate that, while it is possible for transfer learning methods to counter the effects of variability, each method is more "specialized" to a specific case. This implies that using a single transfer learning method might not be sufficient to create a zero-calibration BCI. Note that, in addition to the parameters that we have investigated, there are other parameters that can induce variability, such as target probability, flash interval and mislabeled training sets. Nevertheless, good performances can be achieved when combinations of these methods are used in a way that preserve their properties that counter variability. In the next part, we will present such combinations, which constitute the main contribution of this thesis.

# PART III

# CONTRIBUTED METHODS

# CHAPTER 5
# OPTIMAL TRANSPORT

In the previous part, we analyzed the sources of EEG signal variability, quantify them, and considered three different transfer learning methods that address them. In this chapter, we detail the optimal transport framework and propose two different methodologies. The first uses optimal transport in the feature extraction step, while the second uses optimal transport as a classifier. Then, we propose four transfer learning classifiers based on combinations of the previously discussed methods. We present our results on an experimental dataset and conclude this chapter with a discussion.

## 5.1 Introduction

Optimal Transport (OT) was initially formulated as a resource allocation problem. It was formalized by the french mathematician Gaspard Monge in 1781, who defined it as the search for a transport map that minimizes a certain cost. The original formulation was however ill-posed and had no solution in certain cases. In 1971, Kantorovic proposed an adaptation of the optimal transport problem which was well posed [Kantorovitch, 1958]. The original formulation of the problem was converted into a search for a probabilistic coupling which minimizes a cost function. A probabilistic coupling is a construction that allows to study a specific relation between two random variables. It is a probability measure $\gamma$ defined on the product space of two probability measures $\mu, \nu$, such that its marginals coincide with $\mu$ and $\nu$ [Villani, 2008; Santambrogio, 2015].

In our work [Gayraud et al., 2017], we proposed to use a discrete regularized adaptation of the Kantorovic formulation to handle covariate shift in ERP-based BCI. Our promising results encouraged us to continue exploring this particular

framework. In chapter 4, we saw that optimal transport was able to effectively deal with certain types of variability, such as peak amplitude and peak latency ERP variability. In addition, we observed that no transfer learning method was able to deal with every source of EEG signal variability.

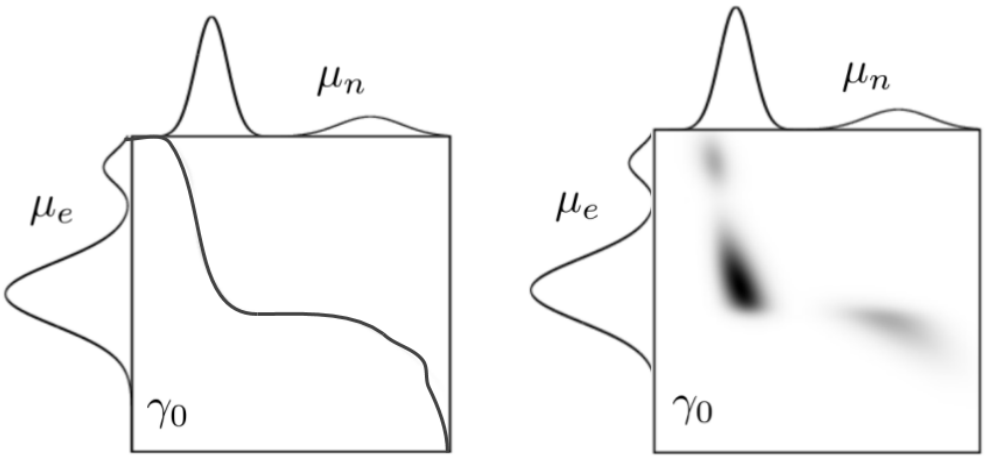
So far, we have seen uses of optimal transport in the feature extraction part of the BCI system. In the following sections, we detail the optimal transport problem and propose a second transfer learning technique based on optimal transport in the classification step. We propose to use these two technique in a classification scheme that combines optimal transport and bagging classification (chapter 3, section 3.3.3). We apply our methods on datasets A and B, described in chapter 2, section 2.3.1 and conclude this chapter by discussing the results.

## 5.2 Optimal Transport as a Transfer Learning Method

### 5.2.1 Regularized Discrete Optimal Transport

Let $\mathcal{D} = \{\mathcal{X}, \mathrm{P}(\mathbf{X})\}$ be the domain of a dataset acquired during an ERP-based BCI session, coupled with the corresponding labels $\mathbf{Y} = \{y_i\}_{i=1}^{I_n}$. We denote $\mathbf{X} = \{x_i\}_{i=1}^{I_n} \in \mathcal{X}$ the set of $I_n$ feature vectors; $\mathrm{P}(\mathrm{X}) \in \mathcal{P}(\mathcal{X})$ the probability distribution from which the sample $\mathbf{X}$ is drawn; and $\mathcal{P}(\mathcal{X})$ the space of all probability measures over $\mathcal{X}$. Let $\mathcal{D}^s$ be the *source* domain for which the labels $\mathbf{Y}^s$ are available, and $\mathcal{D}^t$ the *target* domain for which they are unknown. We seek to train a classifier to recover the unknown labels $\mathbf{Y}^t$.

In chapter 3, section 3.2.1 we discuss issues that are addressed by transfer learning. One of these issues is a phenomenon called covariate shift, in which the probability distributions of the *source* and *target* samples are different, i.e. $\mathrm{P}(\mathbf{X}^s) \in \mathcal{P}(\mathcal{X}^s) \neq \mathrm{P}(\mathbf{X}^t) \in \mathcal{P}(\mathcal{X}^t)$ [Shimodaira, 2000]. This phenomenon often occurs in BCI data [Clerc et al., 2016; Jayaram et al., 2016]. This problem can be handled by optimal transport, which allows us to transport probability mass between two probability distributions through the recovered probabilistic coupling $\gamma$. Hence, assuming that a transformation causes this drift between domains $\mathcal{D}^s$ and $\mathcal{D}^s$, we propose to recover a transport plan to map the *target* features onto the domain of the *source* features using the discrete formulation of optimal transport theory.




(a) optimal transport (b) optimal transport with entropic regularization.

Figure 5.1: An illustrative 1D example of the computation of $\gamma_0$ without and with entropic regularization. We see that the transport plan $\gamma_0$ between distributions $\mu_n$ and $\mu_e$ is sparse when $\lambda = 0$. However, for a higher value of $\lambda$, the solution is more dense. Adapted from [Solomon et al., 2015].

We formally define regularized discrete optimal transport in the following way: consider the estimated empirical marginal distributions $\mu_s = \sum_{i=1}^{I^s} p_i^s \delta_{x_i^s}$ and $\mu_t = \sum_{i=1}^{I^t} p_i^t \delta_{x_i^s}$ of the samples in $\mathbf{X}^s$ and $\mathbf{X}^t$. $I^s$ and $I^t$ denote the sizes of the *source* and *target* samples respectively, $\delta_{x_i}$ is the Dirac function at $x_i \in \mathcal{X}$ and $p_i$ is the probability mass associated to the $i^{th}$ sample element, where $\sum_{i=1}^{I^s} p_i^s = 1$ and $\sum_{i=1}^{I^t} p_i^t = 1$. We aim at a probabilistic coupling $\gamma_0 \in \mathcal{B}$ satisfying the following minimization problem:

$$\gamma_0 = \arg\min_{\gamma \in \mathcal{B}} \langle \gamma, \mathbf{C} \rangle_{\mathbf{F}} + \lambda \mathbf{R}_s(\gamma) \tag{5.1}$$

where $\langle \cdot \rangle_{\mathbf{F}}$ is the Frobenius dot product, and $\mathcal{B}$ is the set of all probabilistic couplings between $\mu_s$ and $\mu_s$. In practice, $\mathcal{B} = \left\{ \gamma \in (\mathbb{R}^+)^{I^s \times I^t} \mid \gamma \mathbf{1}_{I^t} = m_s, \gamma^{\mathbf{T}} \mathbf{1}_{I^s} = m_t \right\}$, where $\mathbf{1}_d$ denotes a $d$-dimensional vector of ones; and $m_s = (p_1^s, \cdots, p_{I^s}^s)$ and $m_t = (p_1^t, \cdots, p_{I^t}^t)$ denote the probability vectors of each sample feature vector set $\mathbf{X}^s$, $\mathbf{X}^t$ respectively.

The first term of equation 5.1 is the discrete adaptation of the Kantorovic formulation of the OT problem [Kantorovitch, 1958]. $\mathbf{C}$ is a cost function matrix, whose elements correspond to a distance between two points, $\mathbf{c}_{ij} = \mathrm{d}(x_i^s, x_j^t)$,

$x_i^s \in \mathbf{X}^s$, $x_j^t \in \mathbf{X}^t$. It can be intuitively understood as the effort required to move probability mass from $x_i^s$ to $x_j^t$. In this work, unless stated otherwise, the metric we use is the squared Euclidean distance $\mathrm{d}_E(x_i^s, x_j^t)^2 = \|x_i^s - x_j^t\|_2^2$, as it guarantees the existence of a unique coupling. When the squared Euclidean distance is used as the cost function, the first term leads to a sparse solution $\gamma_0$ [Villani, 2008].

The second term regularizes $\gamma_0$ by its entropy, as proposed by Cuturi et al. [Cuturi, 2013]:

$$\mathbf{R}_s(\gamma) = \lambda \sum_{i,j} \gamma(i,j) \log \gamma(i,j) \tag{5.2}$$

This allows for smoother variants of $\gamma_0$. In addition, the sparsity of $\gamma_0$ gradually decreases as $\lambda$ increases. This renders the transport more robust to noise, provided that outliers are assigned a small probability value. The regularization term $\mathbf{R}_s(\gamma)$ can also be interpreted as a Kullback-Leibler divergence between $\gamma$ and a uniform joint probability $\gamma_u = \frac{1}{N_s N_t}$, which allows for the use of a computationally efficient algorithm based on Sinkhorn-Knopp's scaling matrix approach [Knight, 2008]. An illustrative example of the computation of $\gamma_0$ between two distributions $\mu_e$ and $\mu_n$ is presented on figure 5.1.

### 5.2.2 Method 1: Optimal Transport in the Feature Space

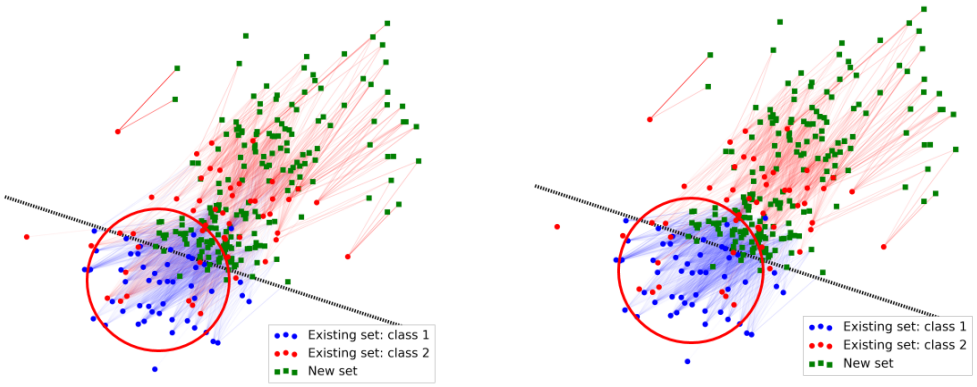
One application of OT, which is also described in chapter 3 section 3.3.2, is to transport the feature vectors of the *target* domain onto the *source* domain. Given a classifier trained on the *source* feature vectors, the transported *target* feature vectors can be classified using the original classifier from the *source* domain. The pipeline of this approach is detailed in figure 5.3a. First, we compute the probabilistic coupling $\gamma_0$ by adding a second regularization term proposed by Courty et al. in [Courty et al., 2017] to equation (5.1), which becomes

$$\gamma_0 = \arg\min_{\gamma \in \mathcal{B}} \langle \gamma, \mathbf{C} \rangle_{\mathbf{F}} + \lambda \mathbf{R}_s(\gamma) + \eta \mathbf{R}_c(\gamma) \tag{5.3}$$

Based on group sparsity, this term makes use of the available class labels $\mathbf{Y}^s$ of the *source* domain:

$$\mathbf{R}_c(\gamma) = \sum_j \sum_l \|\gamma(\mathcal{I}_c, j)\|_F \tag{5.4}$$

where $\mathcal{I}_c$ denotes the set of indices belonging to class $c$ and $\|\cdot\|_F$ denotes the Frobenius norm. In this way, although we do not know the labels of $\mathbf{X}^t$, we make




(a) Entropic regularization (b) Class label and entropic regularization.

Figure 5.2: An example of the difference between using entropic regularization (equation 5.1) and using entropic regularization and class label regularization(equation 5.3). Two toy datasets are simulated. One existing *source* dataset for which we know the class labels, and one new *target* datasets for which we need to recover these labels using the classifier trained on the *source* dataset. The lines connecting the feature vectors of the two datasets are the corresponding coefficients of $\gamma_0$. On 5.2a, we can see that some feature vectors of the *target* dataset have been coupled with feature vectors of the *source* dataset that belongs to different classes. This does not occur anymore in 5.2b, where we add class label regularization to the computation of $\gamma_0$.

sure that most vectors $\{x^s_{i_1}, x^s_{i_2}, \cdots, x^s_{i_{I_j}}\} \subset \mathbf{X}^s$ with which a vector $x^t_j \in \mathbf{X}^t$ was coupled belong to the same class (with $i_1, i_2, \cdots, i_{I_j}$ the $I_j \leq I_n$ non-zero indices of the $j^{th}$ column of $\gamma_0$). Parameter $\eta$ allows us to control the amount of regularization induced by this term. High values enforce the same-class criterion, while low values mean that some of the coupled feature vectors will belong to a different class.

The optimal transport solution $\gamma_0$ is a probabilistic coupling between the estimated empirical probability distributions of the *source* and *target* sets, but it is not a one-to-one mapping between the two sets. Nevertheless, the coefficients of each column of $\gamma_0$ indicate how much of probability mass is transported from the corresponding *target* element to each *source* element. Therefore, we can use the recovered $\gamma_0$ to map $\mathbf{X}^t$ onto $\mathbf{X}^s$ by computing a transformation based on barycentric

mapping as in [Courty et al., 2017],

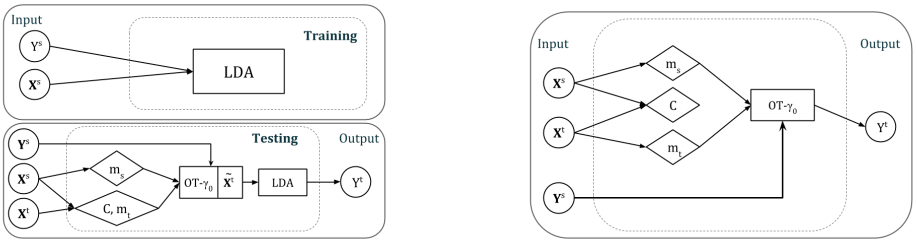
$$\tilde{\mathbf{X}}^t = \text{diag}(\gamma_0^\intercal \mathbf{1}_{N^s})^{-1} \gamma_0^\intercal \mathbf{X}^s \tag{5.5}$$

Each feature vector $x_i \in \mathbf{X}^t$ is thus be mapped onto the weighted barycenter of the features of $\mathbf{X}^s$ that it was coupled with in $\gamma_0$. An example of transporting the feature vectors of a new set using barycentric mapping is displayed on figure 5.2.

### 5.2.3 Method 2: Optimal Transport as a Classification Method

The solution of the optimization problem described in equation (5.1) can also be used as a classifier itself. The transport plan $\gamma_0$ maps elements from $\mathbf{X}^t$ to elements from $\mathbf{X}^s$. Hence, each *target* feature vector will distribute probability mass to a number of feature vectors in the *source* set. Instead of transporting the *target* feature vectors onto the space of the *source* vectors, we can directly use the labels of the *source* vectors each *target* vector is paired with in $\gamma_0$ to make a decision.

The pipeline of this approach can be seen in figure 5.3b. Instead of computing the barycenter of the feature vectors in $\mathbf{X}^s$, we compute the barycenter of its labels,




(a) Pipeline with OT in the feature space

(b) Pipeline with OT as a classifier.

Figure 5.3: During the training process, a set of trials $X^s$ is given as input along with the corresponding labels $\mathbf{Y}^s$. Then, the extracted features $\mathbf{X}^s$ are used to estimate $\mu_s$. When OT is not used as a classifier, we train an LDA classifier. When a new set $X^t$ is given as input to the trained pipeline, we compute the probability vector $\mu_t$; the cost matrix $C$; and solve the OT problem yielding $\gamma_0$. If OT is used in the feature extraction step, the barycentric mapping transported vectors $\tilde{\mathbf{X}}^t$ are given as input to the LDA classifier, which estimates $\mathbf{Y}^t$. Otherwise, $\mathbf{Y}^t$ is directly computed by the OT classifier. Note that, when OT is used as a classifier, there is no need for a training process.

modifying equation (5.5) as follows:

$$\mathbf{y}^t = (\gamma_0^\intercal \mathbf{1}_{N^s})^{-1} \gamma_0^\intercal \mathbf{y}^s \tag{5.6}$$

Vector $\mathbf{y}^s = (y_1, y_2, \cdots, y_{I^s})$ represents the aggregated class labels *source* domain. Vector $\mathbf{y}^t = (y_1, y_2, \cdots, y_{I^t})$ can be viewed as the classifier decision function, where $0 \leq y_i^t \leq 1$. In this work, we consider that a label $y_i$ belongs to the Target class, for which the assigned label is $y = 1$, if $y_i > 0.5$.

Note that, when in the previous section we transport the feature vectors of the target domain onto the source domain, we use the class label regularization term of equation (5.3) to enhance class separability. When using the solution of the OT problem as a classification method, we use equation (5.1), which does not include this term, as it would induce a strong bias on the classification result.

## 5.3 Application to P300-Speller Data

### 5.3.1 Experiment Description

We wish to evaluate the performances of these two OT-based transfer learning methods in offline experiments using datasets A and B (chapter 2, section 2.3.1), who contain EEG recordings acquired during P300-Speller session. Dataset A contains the recordings of 4 healthy subjects, while dataset B contains the recordings of 20 ALS patients. Each dataset consists of three calibration sessions per subject. Optimal transport computes the coupling between the probability vectors of a *source* and a *target* feature vector set. We compute the probability of each feature vector using Kernel Density Estimation. Therefore, we need to have an adequate number of feature vectors per set. Hence, in each experiment, the *target* set is a single session. The *source* set is composed of different calibration sessions of which we use the existing labels to calibrate an OT-based transfer leaning classifier. However, our preliminary experiments showed that, for the method to be computationally efficient, the *target* and the *source* set need to have a similar cardinality. This means that we need to select a subset of the *source* set so that it matches the size of the *target* set. We are thus seeking for a way to partition the *source* dataset, so that it matches a small *target* sample size.

In [Gayraud et al., 2017], we performed experiments where the training and test set of a classifier come from two different sessions whose size is the same. Our results were promising, but while in some cases optimal transport outperformed the state-of-the-art classifiers, in others it did not. These findings confirm the conclusion of the previous chapter, that a combination of transfer learning methods seems necessary to handle different types of variability. In chapter 4, section 4.4, we saw that each method was able to counter only certain types of variability. In particular, we saw that optimal transport was robust against peak amplitude and latency variability, and white additive noise. However, it was not robust against the variability of background activity. The bagging method was on the other hand more robust to that kind of variability than optimal transport. Hence, we can tackle the *source* size and variability problem at the same time by applying the bagging method to both optimal transport classification methods.

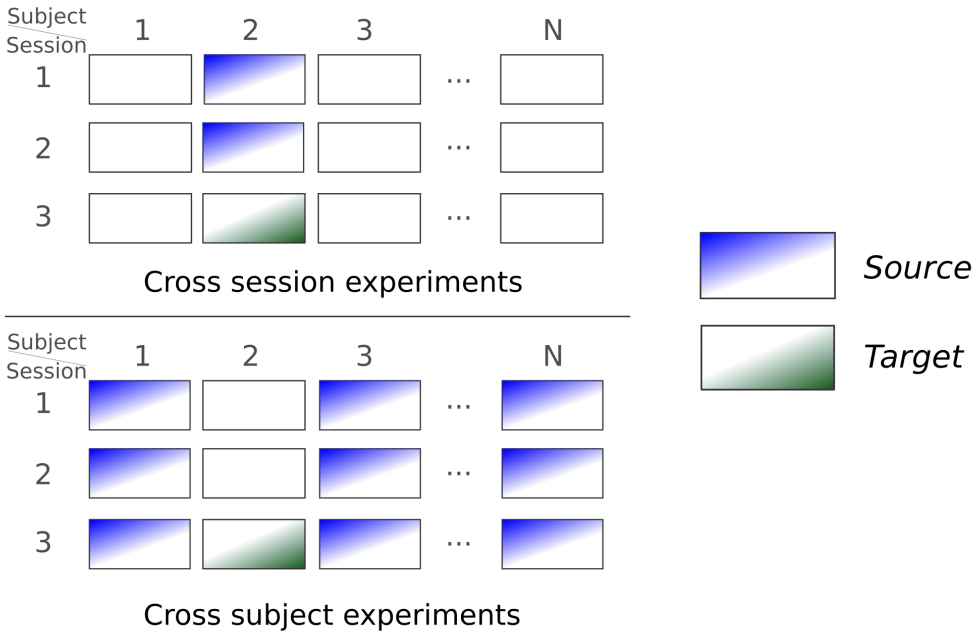



Figure 5.4: An example of the composition of the *source* dataset when the *target* dataset is the third session of subject 2, for cross-session and cross-subject experiments.

We evaluate the performance of each OT-based transfer learning method in a bagging scheme. We conduct both cross-session and cross-subject experiments.

The composition of the *source* dataset for a specific *target* dataset is illustrated on figure 5.4, for both types of experiments. In the cross-session experiments, the *source* set contains data from all the sessions of the *target* subject but the *target* session. In the cross-subject experiments, it contains data from all other subjects but the *target* subject. In each experiment, the *target* session contains data from a single session. The size of the *target* set is $I^t = 1200$. In the cross-session experiments, the *source* session, which is used to train each classifier, consists of all the remaining sessions of the subject which generated the *target* session. In these experiments, the size of the *source* set is $I^s = 2400$. In the cross-subject, it consists of all the sessions of all the other subjects except the subject which generated the *target* session. In these experiments, the size of the *source* set is $I^s = 68400$. The labels associated to the *target* set are not taken into consideration during any of the experiments, and are only used for evaluation purposes. We evaluate the performance of each classification method in terms of Cohen's kappa value.

### 5.3.2 Classification Pipeline

Let $x(t) \in \mathbb{R}^{I_c}$ be a measurement extracted from an EEG signal over $I_c$ electrodes at time $t$ during a P300-Speller session. After pre-processing the signal in the manner described in chapter 2, section 2.3.1, we segment it into trials that last 600 ms starting from stimulus onset. $X_i \in \mathbb{R}^{I_c \times I_w}$ denotes the $i^{th}$ trial whose columns are $I_w = 32$ time samples. We extract the feature vectors of the *source* and *target* trials by aggregating the rows of each trial $X_i$, yielding feature vectors $x_i \in \mathbb{R}^{I_c \cdot I_w}$, resulting in a total of $I_c \cdot I_w = 384$ features. In both optimal transport applications, the regularization term $\lambda$ is set to a value that is low enough so that the matrix $\gamma_0$ is still a sparse matrix. This value was empirically set to $\lambda = 0.001$.

When optimal transport is used in the feature extraction step, the class label regularization term is set to $\eta = 0.1$. Note that, our preliminary results showed that the method was robust to different values of $\eta$. Then, a Linear Discriminant Analysis (LDA) classifier is trained on $\mathbf{X}^s$, and used to predict the labels $\{y_i^t\}_{i=1}^{I^t} = \mathbf{Y}^t$ that correspond to $\{\tilde{x}_i^t\}_{i=1}^{I^t} = \tilde{\tilde{X}}^t$.

Each method is integrated in a bagging scheme. We create $k = 50$ bootstraps of length $I^t = 1200$ by sampling the training set uniformly and with replacement, respecting the class imbalance. We train an classifier instance on each bootstrap.

During testing, each instance produces a prediction. All of the predictions are aggregated via a voting scheme, that is, a majority vote, to produce the final result.

We compare four different classification methods:

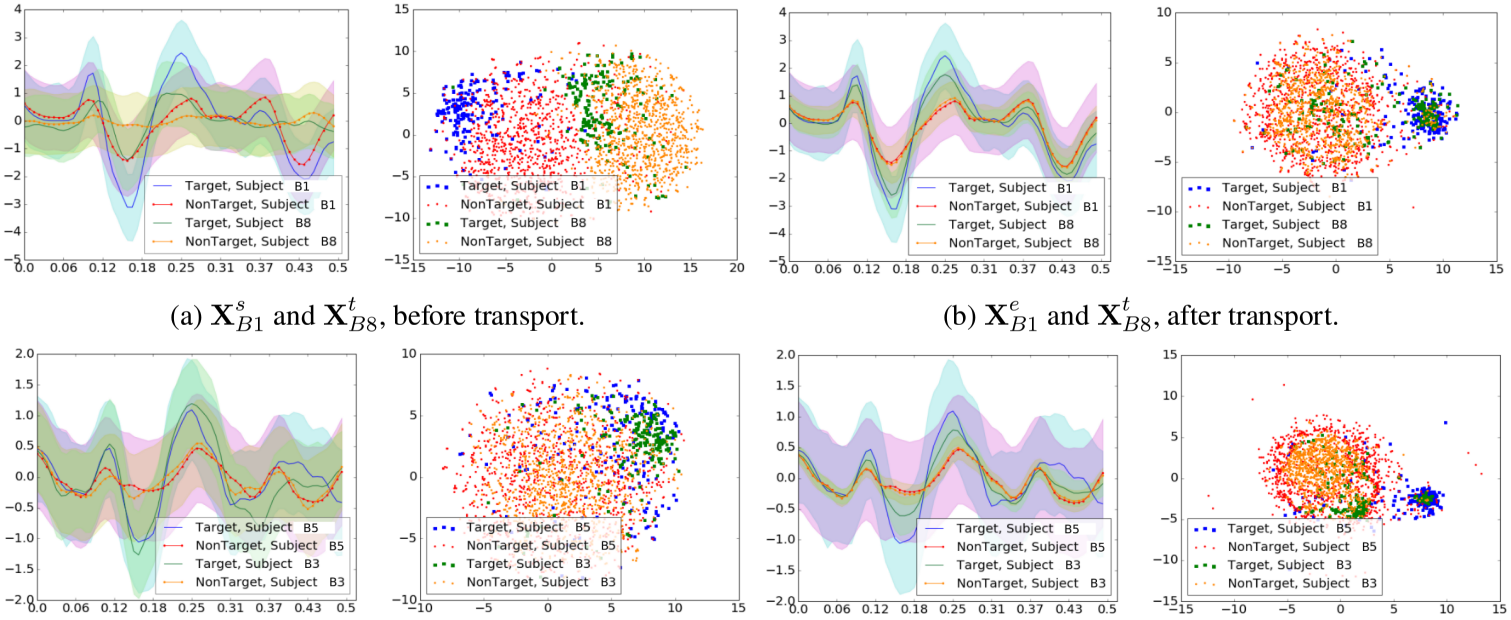
1. **LDA**, a simple LDA classifier trained on the entire *source* set,
2. **Ens+LDA**, a Bagging LDA classifier,
3. **Ens+OT+LDA**, a Bagging LDA classifier who uses optimal transport in the feature space
4. **Ens+OT**, a Bagging optimal transport classifier

## 5.4 Results

### 5.4.1 Feature Transportation Example

We introduce this section by illustrating an example of a transport between the feature vectors of two randomly chosen pairs of sessions in dataset B. We display two examples of the estimated optimal transport in figure 5.5. In the first, the *source* and *target* feature vector sets are $\mathbf{X}^s_{B1}$ and $\mathbf{X}^t_{B8}$ respectively, while in the second, $\mathbf{X}^s_{B5}$ and $\mathbf{X}^t_{B3}$. The subscripts denote the subject indices. Recall that each subject performed three calibration sessions. We only used the first session of these two subjects in this example. Figures 5.5a and 5.5c show the original datasets, while figures 5.5b and 5.5d illustrate the outcome after computing $\tilde{\mathbf{X}}^n_{B8}$ and $\tilde{\mathbf{X}}^n_{B3}$. On the right side of each figure, we display a 2D projection of the features using t-distributed stochastic neighbor embedding (t-SNE) [Maaten and Hinton, 2008]. On the left side of each figure, we can observe the average ERP response and standard deviation. The ERP response was computed on using the first Xdawn filter, estimated on $\mathbf{X}^s_{B1}$ and $\mathbf{X}^s_{B5)}$ using the algorithm described in Rivet et al. [2009], for both sessions and both classes.

The examples illustrated on figure 5.5 give us some insight on the process and how it acts on the components of the EEG signal. By looking at figures 5.5b and 5.5d, we can see that the transport causes a decrease in the variance of the response, for both the Target and Nontarget classes. This is the effect of the entropic regularization




(a) $\mathbf{X}^{s}_{B1}$ and $\mathbf{X}^{t}_{B8}$, before transport.

(b) $\mathbf{X}^{e}_{B1}$ and $\mathbf{X}^{t}_{B8}$, after transport.

(c) $\mathbf{X}^{e}_{B5}$ and $\mathbf{X}^{t}_{B3}$, before transport.

(d) $\mathbf{X}^{e}_{B5}$ and $\mathbf{X}^{t}_{B3}$, after transport.

Figure 5.5: Examples of the barycentric mapping induced by $\gamma_0$ for pairs of sessions. On the left we see the average response and standard deviation of the first Xdawn filter projection for the Target and Nontarget classes Rivet et al. [2009]. On the right side, we see the 2D projection of the features, projected using t-SNE.

parameter, which in this particular example was equal to $\lambda = 0.01$. We selected this high value for $\lambda$ to demonstrate the effect of this parameter on the transport. Low values of $\lambda$ map each *target* vector to a small number of *source* vectors, whose barycenter does not approximate the class center well. Higher values lead to denser results for the coupling $\gamma_0$. This implies that the barycenter of each transported feature vector is computed from a larger sample of *source* vectors. Therefore, it becomes a better approximation of the class mean and the *target* vectors are transported closer to that mean, which results in a decrease in the variance of each class.

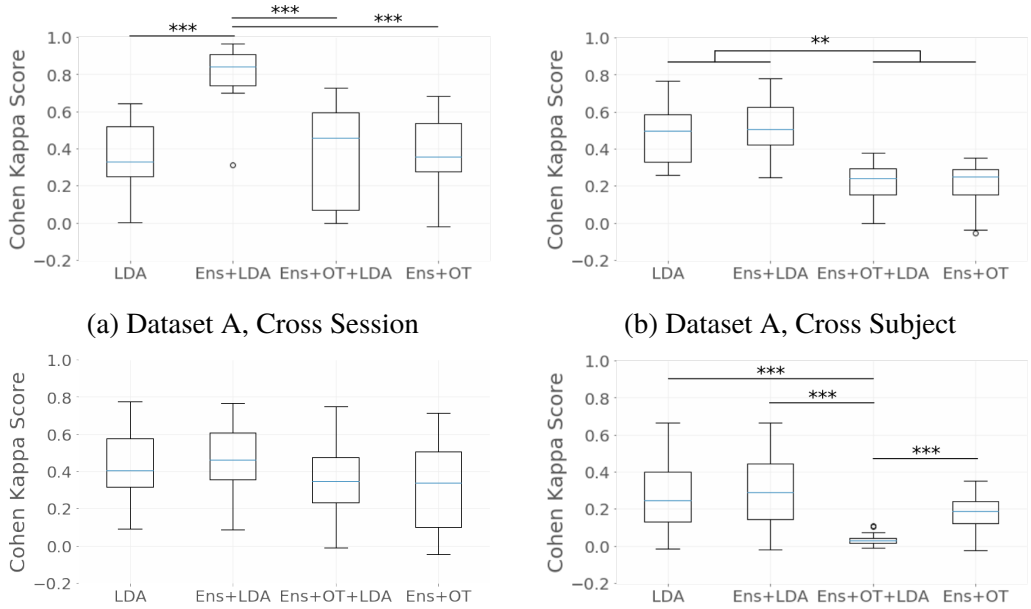
We see that, after the transport, the average values of the ERP components match, especially for the nontarget response. However, due to the presence of a much larger number of Nontarget class elements in the training set, it appears that samples whose P300 peak amplitude is low are drawn to the *source* Nontarget class mean.

### 5.4.2 Cross Session Offline Experiments

Figures 5.6a and 5.6c show a box plot of the performances of the four proposed classification methods, for dataset A and B respectively. For dataset A, the method with the best performance is Ens+LDA, with an average kappa score of 0.79. All other methods have an equivalent kappa score, which on average was equal to 0.37. The performances of the four transfer learning methods are equivalent for all four methods for dataset B. The average kappa scores of LDA and Ens+LDA are respectively equal to 0.44 and 0.47. Both OT-based methods have an average performance of $0.33$.

### 5.4.3 Cross Subject Offline Experiments

The results of the cross-subject experiments were, unsurprisingly, not as good as those of the cross-session experiments. For dataset A, we can see on figure 5.6b that the LDA and Ens+LDA outperform the OT-based classifiers. The average performances of the first two are equal to 0.47 and 0.50 respectively. The OT-based methods both have an average kappa score equal to 0.20. For dataset B, all methods but Ens+OT+LDA had a similar performance. The low performance of OT in this case is discussed in the following section. The average kappa score was equal to 0.27




(a) Dataset A, Cross Session

(b) Dataset A, Cross Subject

(c) Dataset B, Cross Session

(d) Dataset B, Cross Subject.

Figure 5.6: Performance of Transfer Learning Methods. The first is an LDA classifier trained on the entire *source* dataset. The second is a bagging LDA classifier. The third is a bagging LDA classifier, where optimal transport has been used to transport the *target* data onto each bootstrap *source* sample. The fouth is the optimal transport classifier presented in this chapter. The statistical significance of the differences between each method were computed using Wilcoxon's signed rank test. The significance threshold was set equal to $\alpha = 0.05$, The resulting p-values were corrected with the Benjamini/Hochberg method.

for the LDA classifier, 0.33 for the Ens+LDA method, 0.03 for the Ens+OT+LDA method and 0.19 for the Ens+OT method.

## 5.5 Discussion

Our experimental results provide us with an insight on the type of variability that is present in the data. We discuss these results in light of the results of chapter 4. Recall that, in that chapter, we conducted experiments that simulated different types of variability. In these results, we saw that optimal transport is robust to peak amplitude and latency variability, as well as additive white noise. This remains unchanged when the forward model is not the same between the simulated

*source* and *target* datasets. Nevertheless, the same method proved ineffective in the presence of background activity.

First of all, in both cross-session and cross-subject experiments the forward models can be assumed to differ between the *source* and *target* data. In cross–session experiments, this can be attributed to a different placement of the electrodes, while in cross–subject experiments the difference should be more pronounced as there are also neurophysiological differences between users. In this chapter, we saw that both the LDA method and the Ens+LDA method mostly outperformed the OT-based methods, especially in the cross-subject experiments. Based on our findings, this implies that the background activity is also different across sessions, and even more so across subjects.

In the cross subject experiments of dataset B, the Ens+OT+LDA classifier had a very poor performance. Upon scrutiny of the data, we observed that both the *source* and *target* datasets had a substantial amount of outliers. The solution of equation (5.1) produced a denser $\gamma_0$ matrix, which resulted in transportation plans that move all of the *target* samples onto the barycenter of the entire *source* dataset. In such cases, the use of the entropic regularized optimal transport is advised. Nevertheless, we maintained this result to point out this issue.

The obtained results show that cross-session transfer learning yields better performances than cross-subject transfer learning. Nevertheless, the performances of these transfer learning classification methods are characterized by variability, which mirrors EEG signal variability.

## 5.6 Conclusion

In this chapter, we provided a detailed description of the optimal transport method. Additionally, we proposed four transfer learning methods, which we evaluated in cross-session and cross subject experiments. Our results show that transfer learning improves the generalization capacities of existing classification methods. However, OT-based classifiers performed poorly, suggesting that the EEG variability both across sessions and across subjects lies mostly in the background brain activity.

# CHAPTER 6
# RIEMANNIAN FEATURES: ASSESSING CLASSIFICATION CONFIDENCE

In the previous chapter, we presented the results of combining optimal transport and ensemble learning. In this chapter, we propose to apply this methodology to Riemannian geometry-based feature. First, we study the variability in the feature space, i.e. the Riemannian manifold of Symmetric Positive Definite matrices. We propose a way to quantify the quality of the training set and use that information to generate a marker of classification confidence, based on high-dimensional statistics. The first section provides the basic principles of high-dimensional statistics and the geometry of the manifold. In the second section, we formally present the separability marker. In the third section, we propose a method that, using the separability marker, combines ensemble learning; optimal transport; and Riemannian geometry. We present our results on an experimental dataset in the fourth section and discuss them in the conclusion of this chapter.

## 6.1 Introduction

The aim of a transfer learning classification method is to acquire knowledge in an intelligent way, so that it may be used in a classification task even when something has changed between training and using the classifier. These changes occur between the domain of a *source* dataset, which we assume is used to train the classifier, and the domain of a *target* dataset. We formally defined them in chapter 3. In ERP-based BCI, EEG signal variability is one of the primary causes

of drifts between two domains [Clerc et al., 2016]. So far, we have seen how various methods transfer knowledge in the presence of that variability. Riemannian geometry, optimal transport and ensemble learning have shown promising results in simulated data. As we saw in chapter 4, each one of these transfer learning methods was able to deal with certain types of EEG variability. Our objective is to find a combination of the three methods that exploits their properties.

In the previous part we saw that Riemannian geometry is robust to affine transformations and therefore to changes that affect the forward model. We will hence consider the feature space of covariance matrices. In chapter 3, section 3.3.1, we saw that this space is a Riemannian manifold: the manifold of Symmetric Positive Definite matrices (SPD). The SPD manifold is a high-dimensional space; its embedded dimension is equal to $d = I_c(I_c + 1)/2$. In non-invasive EEG-based BCI where the classification feature is the sample covariance matrix, $I_c$ corresponds to the number of electrodes used for the recording. For instance, the dimension of the manifold will be $d = 78$ for $I_c = 12$ electrodes. High dimensionality leads to various problems often described as "curse of dimensionality" [Beyer et al., 1999; Lotte et al., 2007; Sugiyama and Kawanabe, 2012; Lotte et al., 2018]. Nevertheless, high-dimensional spaces possess properties that allow us to gain significant insight on the shape of the multidimensional feature space of covariance matrices [Hopcroft and Kannan, 2014]. Under the assumption that this space, embedded with the Riemannian metric, can be approximated as a set of random variables drawn from multidimensional Gaussian distributions, we use established properties of multidimensional Gaussians to develop our separability marker.

In the previous chapter, we evaluated two classification methods based on optimal transport and ensemble learning. We saw that bootstrap aggregating (bagging) enhances the classification results of an LDA classifier. Recall that the first step of bagging is the generation of training samples by randomly selecting feature vectors out of the *source* set with replacement. Typically, each bootstrap sample has the same size as the *source* set and is used to train a single classification pipeline. In order for the method to be efficient, bagging necessitates the generation of a large amount of training samples from the *source* set [Aslam et al., 2007]. This can prove inefficient during online BCI use. We tackle this issue by considering each session in the dataset as a sample and train an equal amount of classifiers. Then, we propose to use the separability marker as a classification confidence weight. These

weights are used in the voting step of the ensemble learning method.

In the following sections, we present the contribution of our research. First, we provide theoretical assumptions on the distribution of the feature vectors in the feature space, i.e. the Riemannian manifold of SPD matrices. We present relevant geometric properties on high-dimensional spaces and give an intuition on the shape of high-dimensional Gaussian distributions. Then, we assess whether these high-dimensional properties apply to the distribution of the feature vectors on the SPD manifold. Based on this analysis, we present the separability marker, which provides a marker of confidence that can be relayed with the result of any classification algorithm that uses the Riemannian distance in its decision function. We use the separability marker to combine ensemble classification, optimal transport, and Riemannian geometry, in a unified transfer learning method. We evaluate the performance of this method and compare it to a state-of-the art Riemannian classification method. We present our results, discuss them and conclude this chapter with possible future extensions of our work.

## 6.2 Geometrical and Statistical Properties

### 6.2.1 Theoretical Assumptions on the Feature Space

To describe and understand how the distribution of the feature vectors is shaped on the SPD manifold, we initially need to establish some assumptions on our data. Our features are sample covariance matrices that follow a Wishart distribution. Therefore, for a $I_c \times I_c$ covariance matrix $\Sigma_i$, we can write $\Sigma_i \sim \mathcal{W}(\Sigma, I_c)$, where $\Sigma$ is the covariance matrix of the multivariate Gaussian distribution we assume is generating the trials $X_i$ of a single class, and $I_c$ corresponds to the number of electrodes used by the BCI during the acquisition of the EEG signal.

The Wishart distribution is a multivariate generalization of the chi squared distribution. Hence, the feature space of sample covariance matrices of a single class can be approximated by a spherical Gaussian distribution for large values of $I_c$.

Under that assumption, we describe the shape of that space by using some of the properties that apply to random variables drawn from high-dimensional spherical multivariate Gaussian distributions.

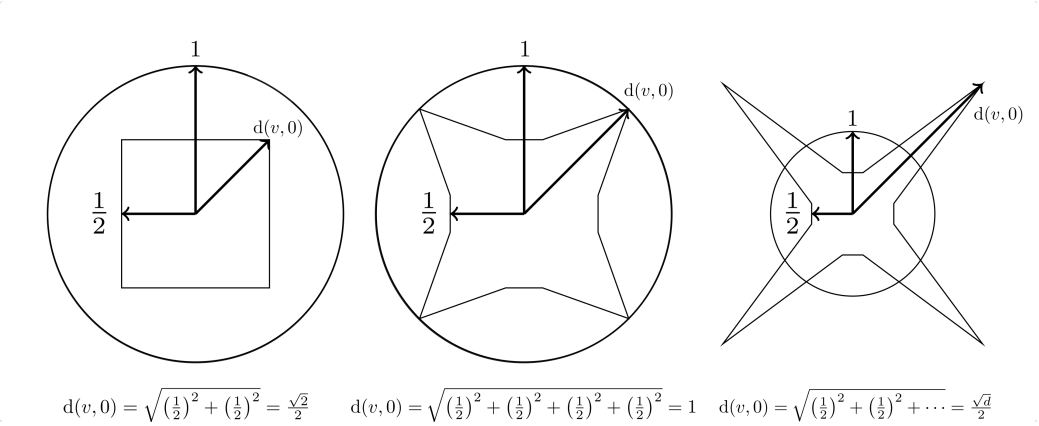



Figure 6.1: Example of the distance $\mathrm{d}(v, 0)$ between a vertex $v$ of a unit cube contained within a unit sphere and the origin, in dimensions 2, 4 and $d$. When $d = 2$, all of the cube lies within the sphere. As $d$ increases, the vertices of the cube move outside the sphere. However, the rightmost illustration is deceptive: the edges of the cube will still lie inside the sphere. Adapted from [Hopcroft and Kannan, 2014].

### 6.2.2 Gaussian Distributions in High Dimensional Spaces

The geometry of high-dimensional spaces presents us with a very counter-intuitive set of phenomena [Hopcroft and Kannan, 2014]. An intriguing effect of high-dimensionality is for example to observe which part of the unit cube is contained inside the unit sphere in $d$ dimensions, displayed on figure 6.1. These "unnatural" properties make the geometrical analysis of the high-dimensional spaces a complicated endeavor. In the specific case of high-dimensional Gaussian distributions, Hopcroft et al. [Hopcroft and Kannan, 2014] present some interesting observations that elucidate some of their geometrical characteristics. The following observations allow us to construct a marker of class separability, when each class is a set of random variables that follows a Gaussian distribution.

1. Lower-dimensional Gaussian distributions have their mass concentrated near their expected values. In high dimensions, there is very little mass located near the expected value of a multivariate Gaussian distribution. Most of the mass of a spherical Gaussian multivariate distribution is concentrated within an annulus of constant width. The width of this annulus is equal to the standard deviation $\sigma$ of the distribution.

2. The expected squared Euclidean distance of a random variable from the expected value of the Gaussian distribution it is drawn from, is in fact on the order of $d\sigma^2$, where $d$ is the dimension and $\sigma$ is the standard deviation of the distribution.

3. Any two randomly drawn points will almost surely be orthogonal with respect to the expected value of the distribution. This implies that,

   - for a binary classification problems where the features are multidimensional, all features belonging to a class will be almost equidistant.
   - the average distance $\delta_I$ between two features that belong to the same class will be related to the average distance to the distribution center through the equation $\sqrt{2}\delta_C = \delta_I$.

4. Given two spherical Gaussians with centers p and q separated by a distance $\delta$, the distance between a randomly chosen point x from the first Gaussian and a randomly chosen point y from the second is close to $\sigma\sqrt{\delta^2 + 2d}$

### 6.2.3 Geometric Properties of the Riemannian Manifold

All of the above observations are proven in [Hopcroft and Kannan, 2014] for Euclidean high-dimensional spaces. Nonetheless, our data lives in a Riemannian manifold, and we have to know whether these observations hold, in spite of the curvature of this particular space. Since these properties use trigonometric properties, we will use the works of [Berger, 2012] to get an insight on the effect of the curvature of the manifold.

Let $T$ be a a geodesic triangle, that is, a triangle on the manifold whose edges are minimizing geodesics. Because geodesics are uniquely defined on the Riemannian manifold of symmetric positive definite matrices under the Riemannian distance $\mathrm{d}_R$, $T$ can be uniquely mapped onto the tangent space $Tp\mathcal{M}$ of the manifold, by fixing point $p$ to one of its three vertices. The Topogonov theorem states that the edges and angles of $\tilde{T} \in Tp\mathcal{M}$ have upper and lower bounds with respect to the bounds of the sectional curvature of the manifold [Berger, 2012].

It has been shown that the lower and upper bounds $K_C^-$, $K_C^+$ of the sectional curvature $K_C$ of the SPD manifold are $K_C^- = -1/2 \leq K_C \leq K_C^+ = 0$ [Bridson

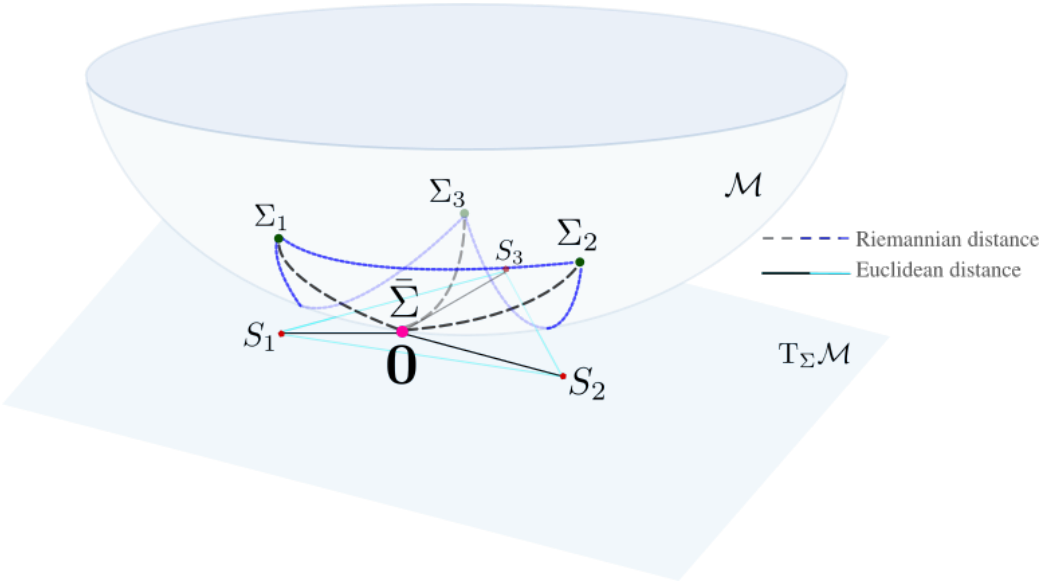



Figure 6.2: Conceptual example of four geodesic triangles on manifold $\mathcal{M}$ formed by the covariance matrices $\Sigma_1$, $\Sigma_2$, $\Sigma_3$, $\bar{\Sigma}$ and their projection onto the tangent space at $\bar{\Sigma}$. Four new triangles are formed by the projections $S_1$, $S_2$, $S_3$, $\mathbf{0}$, where $\mathbf{0}$ denotes the origin on the tangent space (which is the tangent space projection of $\bar{\Sigma}$). These triangles have approximately the same size as the geodesic triangles, provided that $\bar{\Sigma}$ is the Riemannian mean of the three covariance matrices. Note that $\mathrm{d}_E(\mathbf{0}, S_i) = \mathrm{d}_R(\bar{\Sigma}, \Sigma_i)$, where $\mathrm{d}_E$ , $\mathrm{d}_R$ denote the Euclidean and Riemannian distance respectively.

and Haefliger, 1999; Pennec, 2009]. This implies that the Riemannian distance between two features on the Riemannian manifold can be approximated with little error by the Euclidean distance between their projection on the tangent space, provided that we choose an appropriate reference point. Typically, that reference point is the Riemannian mean of all the features. Figure 6.2 displays an illustrative example of tangent space projection for three covariance matrices.

Therefore, we can use the above described properties of multidimensional Gaussian distributions to establish a separability marker for a two-class set of features.

### 6.2.4 The Separability Marker

Our goal in designing the separability marker is to quantify the quality of a training dataset, which in turn gives us a weight that denotes the confidence we can put on the results of a classifier that was trained with that set. One way to do that is to obtain a measure of the amount of overlap between the two classes in that training set. When the feature vectors are high-dimensional, this boils down to estimating the overlap in the annular regions where their distributions are concentrated. Using the properties of high-dimensional spherical Gaussian distributions, we will define a region where the possibility of class overlap is increased, by taking into account the distance between the two centroids.

We begin by scaling the distances between the features and their respective centers by $\sqrt{d}$, to obtain distributions that are no longer affected by the dimensionality of the space. Then we calculate the separability marker $SM$ in the following way. For each class, we estimate the probability density function of the distribution of distances between the class mean and each feature vector from that class. This gives us two curves which correspond to the estimated probability density functions. We scale and translate the two curves so that the distance between their expected values is equal to the distance between the two class means. If there is an overlap region, the curves will intersect.

We formally introduce the separability marker $SM$ of a domain $\mathcal{D} = \{\mathcal{X}, P(\mathbf{X})\}$, where $\mathbf{X} = \{x_k\}_{k=1}^{I_n} \subset \mathcal{X}$ denotes a sample of $I_n$ $d$-dimensional feature vectors from one of two classes, which we label $T$ for Target and $N$ for Nontarget.

**The feature domain** Let $x_i^T, i \in \{1, \cdots, I^T\}$ and $x_j^N, j \in \{1, \cdots, I^N\}$ denote a feature vector that belongs to class $T$ and $N$ respectively, where $I^T$ and $I^N$ are the sample sizes of each class respectively. We denote by $\bar{x}^T, \bar{x}^N$ the estimated mean of each class. Depending on whether the feature space is the Euclidean space or the SPD manifold, the mean can be the Euclidean mean or the Riemannian mean, defined in chapter 3, section 3.3. We denote by $\delta = \mathrm{d}(\bar{x}^T, \bar{x}^N)$ the distance between the two class estimated means. As before, the distance can be either the Euclidean or the Riemannian distance.

**Distances between feature vectors** Let $\delta_i^T = \frac{1}{d}\mathrm{d}(\bar{x}^T, x_i^T)$, $\delta_j^N = \frac{1}{d}\mathrm{d}(\bar{x}^N, x_j^N)$ be the distance of each feature vector to its class center (Target / Nontarget), scaled by the dimension $d$ of the feature vectors. For simplicity, we assume that the feature vectors are distributed in a spherical Gaussian centered at the class mean. When the features are covariance matrices, this assumption is supported by the fact that they are drawn from a Wishart distribution, which can be approximated by a spherical Gaussian when the degrees of freedom are sufficiently high. If we consider these distances as random variables, whose expected value is $E\left[d\delta^T\right] = d\sigma_T^2$, then $E\left[d\delta^N\right] = d\sigma_N^2$ (observation (2)); $\sigma_T, \sigma_N$ are the standard deviations of the Gaussian distributions that generate the feature vectors of each class. This implies that, $E\left[\delta^T\right] = \sigma_T^2$, $E\left[\delta^N\right] = \sigma_N^2$

**Probability distributions of distances** Let $\bar{\delta}^T = E\left[\delta_i^T\right]$, $\bar{\delta}^N = E\left[\delta_j^N\right]$ be the expected value of the distribution of the above defined scaled distances. In order to define a region of overlap between the feature vectors of the two classes, we apply an affine transformation to the distances by taking into account observation (4). We define the following random variables: $\Delta^T = \frac{\delta_i^T + \delta/2}{\bar{\delta}^T}$ and $\Delta^N = \frac{\delta_i^N + \delta/2}{\bar{\delta}^N}$. We denote $\mathrm{p}_T(\Delta^T)$ and $\mathrm{p}_N(\Delta^N)$ the probability density functions of these distributions. Note that, $E[\Delta^T] - E[\Delta^N] = \delta$.

**The Separability Marker** Let $\mathrm{U} = \int \max[\mathrm{p}_T(\Delta), \mathrm{p}_N(\Delta)]d\Delta$ be the area under the union of these two curves and $\mathrm{I} = \int \min[\mathrm{p}_T(\Delta), \mathrm{p}_N(\Delta)]d\Delta$ the area under the intersection of the two curves. We define the separability marker as $SM = (\mathrm{U}-\mathrm{I})/\mathrm{U}$. Intuitively, this marker gives us a comparative measure of the separability. A small value corresponds to a big overlap, so that the classes are harder to separate, whereas a large value corresponds to a small overlap.

## 6.3 Application to P300-Speller Data

### 6.3.1 Geometrical Analysis

We calculate the average distances $\bar{\delta}^T$ and $\bar{\delta}^N$ and the centroids $\bar{\Sigma}$ of each class using the method described in [Pennec et al., 2006]. We also compute the average distance between same-class features, $\delta_I = \mathrm{avg}(d_R(\Sigma_i^c, \Sigma_j^c))$.

Table 6.1: Average distance to centroid and average distance between features for Target (T) and Nontarget (N) class

| $\delta_C^T$ | $\delta_I^T$ | $\delta_C^N$ | $\delta_I^N$ |
|---|---|---|---|
| $4.59 \pm 0.78$ | $5.88 \pm 0.75$ | $4.53 \pm 0.73$ | $5.79 \pm 0.72$ |

Table 6.1 displays the average over all subjects for $\delta_C$ and $\delta_I$ as well as the standard deviation of that average, calculated on the first session of each subject, for both classes. In this case study, our sample covariance matrices are $12 \times 12$ matrices, so the dimension is $d = 78$, and $\sqrt{d} \approx 8.83$.

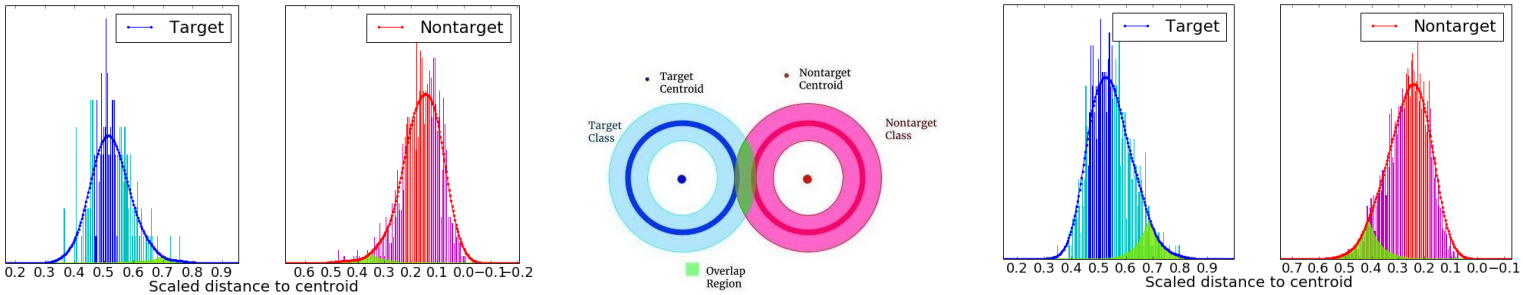
The result of that analysis show that, for both classes, the distance $\delta_C$ between class centroids and class features appears to be on the order of $\sqrt{d}$. Additionally, the features are almost equidistant to each other, which can deducted by observing the standard deviations on Table 6.1.

Finally, if we compare the averages on Table 6.1 for the distances between features to the distances between each feature to its centroid, we can verify that $\sqrt{2}\delta_C = \delta_I$ appears to be holding; we observe only a small deviation that is on average equal to $\delta_I - \sqrt{2}\delta_C \approx 0.61$; it can be attributed to the effect of the curvature on the manifold.

Overall, we see that the Euclidean high-dimensional properties of Gaussian distributions can be applied to the SPD manifold. We proceed by making a cross-class comparison and try to calculate a marker of the amount of overlap that occurs between the two classes. This will give us a significant tool to evaluate the separability of Target and Nontarget classes.

We perform an analysis that is based on the description of the shape of a high-dimensional Gaussian distribution. A 2-dimensional schematic of this analysis is presented in figure 6.3b. We plot a histogram of these distances, and approximate their distribution with a Gaussian kernel.

This analysis allows us to visualize the width of the annulus in which the features are contained, as shown in figure 6.3a and 6.3c. Note that, for the Nontarget class, the distances to the centroid are reversed, so that the histograms are coherent with the representation of figure 6.3b. We estimate the distance distributions of the two classes from the distance distribution histograms using a Gaussian kernel.

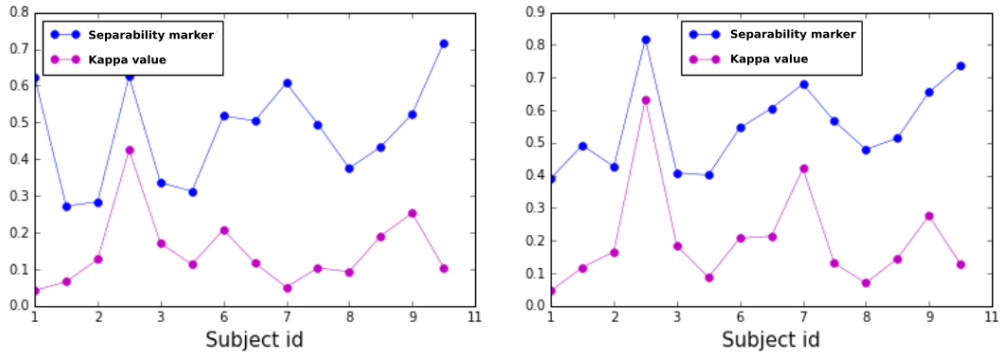



(a) Subject B7, Session 1 (b) Representation in 2D (c) Subject B7, all sessions

Figure 6.3: A visualization of the distribution of distances between the class features and the estimated centroid. The distances have been scaled down by $\sqrt{d}$. The histograms represent the scaled distances distribution, approximated by a Gaussian probability distribution function. On (a), the features are drawn from a single session, whereas (c) shows the distribution of features from three different sessions. (b) provides a 2D visualization of two multidimensional Gaussian distributions. The annulus, where the mass of the Gaussian is concentrated, is color coded to match the standard deviation of each distance distribution. The area in green is a 2D illustration of the region where the two classes have the highest chance of overlapping.

We display the results of this analysis for a single subject, which we have randomly chosen; subject 7. We perform the analysis twice, one to see the separability of the two classes within a single session (the first session), and once more for the union of all sessions. Observe that the distance distributions suggest a Gaussian probability density function; this is in accordance with our theoretical assumptions. We can also see that the features of the Nontarget class are closer to their centroid; the radius of the annulus is smaller. This can also be seen on Table 6.1 by comparing $\delta_C^N$ to $\delta_C^T$.

This overlap region is represented in 2D for a general case in figure 6.3b; histograms 6.3a and 6.3c can be seen as 1D projections of the general case. On figure 6.3a we can observe that, for a single session, the histograms that represent the two classes do not significantly overlap. On the other hand, the overlap is more important in figure 6.3c when the class features come from three different sessions. This is due to the cross-session variability, which is causing an increase in the width of the annulus.




(a) Session 1 (b) All Sessions

Figure 6.4: Comparison between the separability marker $SM$ and the value of Cohen's kappa after classifying the data with the MDRM algorithm. (a) for the only the first session, and (b) for all sessions.

To assess whether the $SM$ is correlated to the performance of a Riemannian classifier, we perform preliminary experiments on a randomly selected subset of sessions from dataset B. We perform single-session and cross session experiments in which we train and test an MDRM classifier, evaluating its performance in terms of Cohen's kappa. This performance is compared to the $SM$, computed over the

training dataset. Recall that the MDRM algorithm uses the covariance matrix $\Sigma_i$ of a trial $X_i$ as a feature, and estimates the centroid of each class in the training set by calculating the Riemannian mean of all the class features. For each new feature, its Riemannian distance to all centroids is calculated, and the smallest among these distances defines the winning class.

Figure 6.4a displays the analysis and results for each subject for the first session, where a 5-fold cross-validation is performed to select a training and testing set. Figure 6.4b is a cross-session evaluation, where the training set of the classification contains the first two sessions, and the test set contains the third. We can see that $SM$ is correlated to the classification performance. High $SM$ values coincide with high classification scores, in both the single-session and the cross-session experiments.

### 6.3.2 $SM$-Weighted Ensemble Learning

The Separability Marker has been applied in [Gayraud et al., 2017] as a classification confidence assessment tool. The results of this research show that it provides meaningful information on the geometrical properties of Riemannian features. We propose a similar application of $SM$ in the context of ensemble learning methods. We saw in the previous chapters that bagging increases the generalization capacity of the LDA algorithm. In addition, methods based on optimal transport perform better when the sample sizes of the *source* and *target* datasets are balanced.

In the previous chapter, we saw that cross-subject experiments have lower results that cross-session experiment. Hence, in this chapter, we focus on cross-subject experiments. We perform our experiments on datasets A and B, described in chapter 2, section 2.3.1. We present three ensemble learning methods that combine different sessions to produce a single classification result. The main pipeline of each method is the following:

- The *target* set is used for testing and consists of a single session.
- The *source* set is used to train an ensemble learning classifier. It consists of all the sessions that belong to different subjects, except for the subject who produced the *target* session.

- The *source* set is divided into training samples, where each session constitutes a sample. We compute $SM$ for each sample, which is later used as a marker of classification confidence.

- The classification features are the tangent space projections of Extended Covariance matrices $\tilde{\Sigma}_i$, described in chapter 3, section 3.3.1. These features are constructed from the concatenation of the archetype Target and Nontarget responses to stimulus,

$$\tilde{X}_i = \begin{bmatrix} \bar{A}_t \\ \bar{A}_n \\ X_i \end{bmatrix}, \qquad \tilde{\Sigma}_i = \frac{1}{I_w} \tilde{X}_i \tilde{X}_i^\intercal, \qquad \tilde{\Sigma}_i \in \mathbb{R}^{3I_c \times 3I_c}$$

  where $I_c$ denotes the number of electrodes and $I_w$ the number of time samples in each trial.

- A classifier is trained over each sample in the training set. When we test the method, each classifier produces a classification score. The final decision of the classifier is computed as a weighted average of these single scores. The weights correspond to the computed $SM$s of each sample, aggregated into a normalized weight vector.

The three ensemble learning classification methods are the following. The first one, which we denote as **C1**, integrates a Riemannian classification algorithm based on tangent space projection into this ensemble learning scheme. For each session, we compute the Riemannian mean of all the extended covariance matrices and use it to project them onto the tangent space at that point. Then, an LDA classifier is trained on the projected matrices. When we test the classifier, each new sample is projected onto the tangent space at the mean of each training sample. The new sample is assigned a classification score by LDA.

The second classification method is denoted **C2**. It adds an additional step between in the tangent space, which is the computation of the transport plan $\gamma_0$. This transport plan is used to transport the *target* tangent space feature vectors onto the domain of the existing ones. We use barycentric mapping to compute the transportation, as described in chapter 5, section 5.2.2. The third classification method **C3** uses optimal transport as well. However, we do not train an instance of

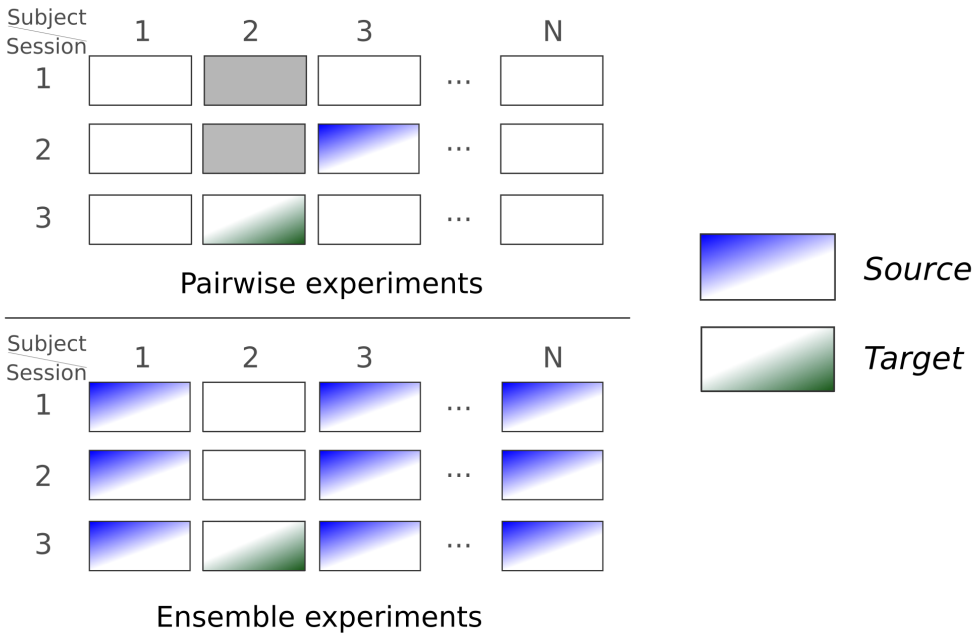



Figure 6.5: An example of the composition of the *source* dataset when the *target* dataset is the third session of subject 2, for pairwise and ensemble experiments. When we perform pairwise experiments, the solid gray boxes indicate sessions that can never be selected.

the LDA algorithm for each sample. Instead, the classification scores are computed directly from the transport plan as label barycenters, according to the method described in chapter 5, section 5.2.3.

Our objective is to prove the potential of combining multiple sessions to improve classification performance, as opposed to using a single session. Therefore, to evaluate the aforementioned methods, we produce the results of pairwise experiments. In each one of these experiments, the *source* and *target* datasets are each composed of the trials of a single session. Naturally, the *source* and *target* sessions cannot be the same session. An illustrative example of the composition of the *source* dataset for a given *target* dataset is presented in figure 6.5, for pairwise and ensemble experiments. We perform pairwise experiments for the following transfer learning methods, which were also presented in chapter 4, section 4.4.1:

1. EC-Rie, which denotes an MDRM classifier trained on Extended Covariances as features,

2. OT, which denotes an LDA classifier where the feature vectors are transported using optimal transport during testing,

3. Ens, which denotes a bagging LDA classifier,

4. LDA, which denotes a simple LDA classifier.

## 6.4 Results

The results of our experiments are presented in figure 6.6. The classification performance metric is Cohen's kappa. For both datasets, the pairwise experiments perform significantly less well than the ensemble learning classifiers. For database A, we can see on figure 6.6a that all three ensemble learning methods perform well, having an average kappa score of $\approx 0.50$. In contrast, the pairwise experiments produce lower performances. On figure 6.6b, we can observe that the results for database B are not the same for the OT-based classification methods **C2** and **C3**. In addition, we notice that the pairwise experiments produce a large number of positive outliers. This indicates a larger amount of variability in the EEG signals of dataset B; a result that was also observed in the previous chapter.

## 6.5 Discussion

In this chapter, we studied the shape of distributions of sample covariance matrices on the Riemannian manifold of symmetric positive definite matrices. When the sample covariance matrices are classification features of a binary MDRM classifier, obtained from BCI applications such as the P300 speller, we can approximate their shape by using theorems that apply to high-dimensional Gaussian distributions. This allows us to define a distance distribution-based separability marker $SM$. We used this marker to combine the transfer learning methods that we have been interested in throughout this thesis: ensemble learning methods, optimal transport, and Riemannian geometry, coupled with the LDA algorithm; and the LDA algorithm itself.

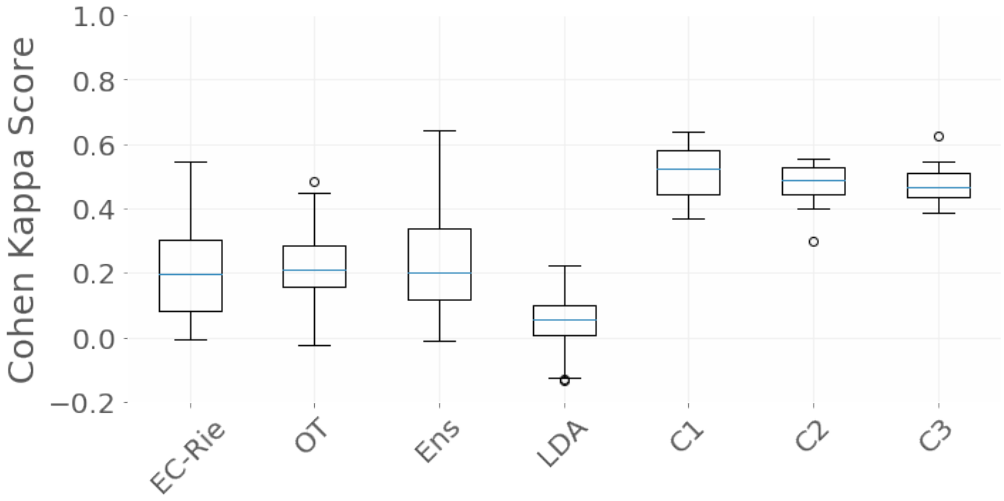
We performed two types of experiments: pairwise experiments in which the *source* and *target* sets consist of a single session, and experiments in which the

*source* set consists of a union of different sessions. While it may seem that these results are not comparable, our objective is not to compare the performances of these seven classification methods between each other. Instead, we wish to show that combining different sessions in the training dataset yields better results than using a single session. Our separability marker provides us with a way to combine these different training sessions in an ensemble learning classification scheme.

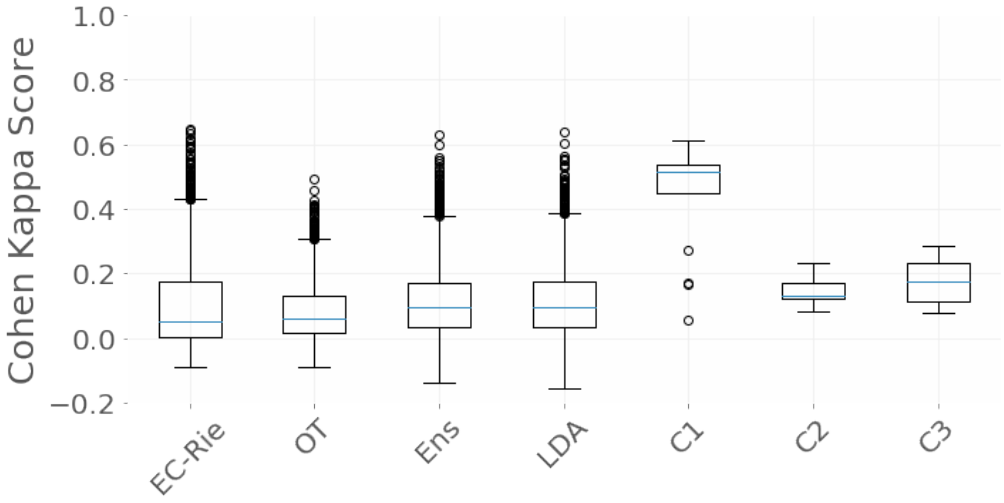
Our results once again demonstrate the high variability of the EEG signal. We saw that all methods performed poorly in the pairwise experiments. However, in dataset B, a few of the pairwise performances were much higher that the average. Moreover, the LDA algorithm performed much better in the pairwise experiments of dataset B than those of dataset A. Recall that dataset B contains EEG recordings from ALS patients, who had never used a BCI in the past. In chapter 4, we saw that LDA outperforms the transfer learning classifiers in the presence of background brain activity, that is assumed to be unrelated to the task and therefore not time-locked to the stimulus. This can also explain why the OT-based methods perform poorly in dataset B, even though their performance still outperform the pairwise experiments.

## 6.6 Conclusion

In this chapter, we proposed a separability marker for Riemannian-based classification methods. We designed this separability marker using geometrical and statistical properties of high-dimensional spaces. We used this separability marker to combine three transfer learning methods. Our results demonstrate that transfer learning can also enhance performances in cross-subject classification. Nevertheless, we still observe a lot of variability in the performances of transfer learning methods. In the next chapter, we propose an unsupervised learning method for a P300-Speller.




(a) Dataset A




(b) Dataset B

Figure 6.6: Results of the cross-subject experiments on datasets A and B. The first four classification methods were evaluated over pairwise experiments. The next three classifiers are ensemble learning classifiers, trained on a mixture of sessions.

# CHAPTER 7
# UNSUPERVISED LEARNING

In the previous chapters we proposed transfer learning methods that tackle EEG signal variability. Such methods necessitate the existence of a labeled set to train the classifier. In this chapter, we introduce an unsupervised classification method that takes into account the structure of a specific ERP-based BCI paradigm: the P300 speller. First, we provide a brief overview of such methods. Then, we introduce the methodology of our approach. We perform simulations of online experiments using an experimental dataset and discuss their results. Finally, we conclude with future perspectives.

## 7.1 Introduction

EEG signal variability is one of the reasons why classification methods fail to generalize [Clerc et al., 2016]. One solution to this problem is the use of transfer learning methods. Given an existing *source* domain transfer learning allows us to train a classification pipeline to adapt to drifts in a new *target* domain. In practice, one can use existing calibration sessions to train such a classifier.

Throughout this thesis, we have thoroughly explored the strength and limitations of this approach in ERP-based BCI. The main advantages of transfer learning methods are that they provide priors. These priors make the resulting classifiers more robust to variability and can be used in a zero-calibration BCI. Their limitation is that they only handle certain types of variability. When the transferred knowledge is not pertinent to the *target* dataset, transfer learning performs worse than the baseline, i.e. a classifier trained on the same domain. We saw such an example in chapter 6 in cross-subject experiments, whose poor results suggest that transfer learning methods cannot efficiently deal with inter-subject variability.

A different approach than transfer learning is the design of unsupervised classification methods. Unsupervised classification methods have recently become the subject of active research in the BCI community. In ERP-Based BCI, the first unsupervised classification method was introduced in 2012 by Kindermans et al. [Kindermans et al., 2012b]. The authors propose an approach which uses the Expectation Maximization algorithm to find the parameters of a simple linear classifier. They later prove the efficiency of their approach on an online study of an auditory P300 speller in [Kindermans et al., 2014]. This work is further extended by Hübner et al. in [Hübner et al., 2017] where the authors modify the keyboard interface in a P300-Speller paradigm to induce priors over the label proportions. The two approaches are combined in the work of Verhoeven et al [Verhoeven et al., 2017] and verified in an online study by Hübner et al. [Huebner et al., 2018].

Unsupervised classifiers learn by adapting to unlabeled data. Typically, an unsupervised classifier tries to infer class distributions as EEG data are acquired. This often implies a long "warm-up" period, during which the BCI feedback might be wrong or non-existent [Kindermans et al., 2012b, 2014]. Inspired by these previous works, we present a preliminary approach to an unsupervised classifier that takes advantage of the structure of the P300-speller paradigm. In the following sections, we detail our method, evaluate the performance of our approach in a simulation of an online experiment using experimental data. Our initial results serve as a proof-of-concept for our unsupervised P300-Speller.

## 7.2 Unsupervised P300-Spelling: A Proof of Concept

### 7.2.1 Flashing Strategies in a P300-Speller experiment

During a visual P300-Speller experiment, the user is looking at an on-screen keyboard. We can define two periods with respect to the graphic interface: a flashing period, during which some characters are flashing, and a rest period during which a character has been proposed, and the system is preparing for the next character. Typically, the user is asked to focus on the character they wish to spell. Each time a character flashes, a P300 response is generated by their brain. The P300 peak amplitude and latency are modulated by various factors, such as attention levels, stress, and fatigue [Polich, 2009]. To increase the spelling speed, the characters do

not flash one by one; instead, they flash in groups. In the first P300-Speller, these groups were the rows and columns of the keyboard, so that any two groups only had one character in common [Farwell and Donchin, 1988]. Other methods group the characters in a different way, such as the flashing methods proposed by [Townsend et al., 2010; Thomas et al., 2014] as an alternative to row/column flashing.

Hence, for each character the user wants to spell, the system generates a sequence of groups of characters to be flashed. These sequences can easily be generated before the BCI starts to flash. Additionally, even if the groups are generated during the flashing period, each group needs to be generated right before it flashes. In other words, there is a group of characters associated to each stimulus. This means that we can obtain the the indices of previous stimuli for each single character on the screen. We propose a method that extracts one feature per character based on this prior information.

### 7.2.2 Feature Extraction

Let $I^n$ be the number of groups of characters that have already flashed, that is, the number of stimuli. Recall that a trial $X_i \in \mathbb{R}^{I_c \times I_w}$, $i \in \{1, \cdots, I^n\}$ is a pre-processed EEG signal segment of length $I_w$, where $I_c$ denotes the number of electrodes. For each character $l \in \{1, \cdots, I_l\}$ we can obtain trials $\mathbf{X}_T^l = \{X_i\}_{i \in \mathbf{G}_l}$, where $I_l$ denotes the total characters on the keyboards (that can be flashed), and $\mathbf{G}_l$ is the set of stimulus indices associated to the groups that contain $l$. Similarly, we can obtain a set of trials $\mathbf{X}_N^l = \{X_i\}_{i \notin \mathbf{G}_l}$. For each one of these two sets, we can compute the average of each set, $\hat{X}_T^l = \frac{1}{I_T^l} \sum_{i \in \mathbf{G}_l} X_i$ and $\hat{X}_N^l = \frac{1}{I_N^l} \sum_{i \notin \mathbf{G}_l} X_i$, where $I_T^l$, $I_N^l$denotes the cardinality of $\mathbf{X}_T^l$ and $\mathbf{X}_N^l$ respectively. We call $\hat{X}_T^l$ the proxy Target average of character $l$ and $\hat{X}_N^l$ its proxy Nontarget average.

Note that, if $l$ is the Target character, i.e. the one attended to by the user, each row of matrix $\hat{X}_T^l$ will enclose the P300 response. In addition, having a sufficient number of trials in the proxy Target set will remove some of the variability upon averaging. Therefore, the peak amplitudes of each row in the real Target average should have the maximum value among all Target averages. Moreover, the Target character should maximize the difference between its Target and Nontarget average, since there should be no high amplitude component in the Nontarget average.

Hence, we can formally define a criterion to select a character $l$. For each

character and each average, we construct a vector $\hat{\mathbf{x}} = (\|\hat{X}_1\|_\infty, \cdots, \|\hat{X}_{I_c}\|_\infty) \in \mathbb{R}^{I_c}$ whose elements are the maximum-norm of every row $\hat{X}_i \in \mathbb{R}^{I_w}$ of matrix $\hat{\mathbf{X}} \in \mathbb{R}^{I_c \times I_w}$. If $\hat{\mathbf{X}}_l^T$ is the real Target average, $\hat{\mathbf{x}}_l^T$ will be the vector of the peak amplitudes of each electrode. The Target character is thus given by the solution of the following equation:

$$\hat{l} = \arg\max_l \|\hat{\mathbf{x}}_N^l - \hat{\mathbf{x}}_T^l\|_F \tag{7.1}$$

This criterion can be used to select a character during a single flashing period.

### 7.2.3 Experiment Description

We perform experiments on a subset of Dataset A, for which the information on which group of characters associated to each stimulus was available. Each session is a calibration session where the subject had to spell the word "CALIBRATION", a total of 11 characters. Hence, there are 11 flashing periods per session, each one consisting of 36 trials. Therefore, since the Target/Nontarget ratio is 1/5 and each character flashes 6 times, $I_T^l = 6$ and $I_N^l = 30$.

For each flashing period in each session, we select a character $l$ among a set of 36 characters, according to the criterion of equation (7.1). In comparison, we simulate a supervised character selection method. We train an LDA classifier using the Xdawn feature extraction method described in [Rivet et al., 2009]. For each flashing period, the set of trials that corresponds to that flashing period is kept apart, and we train the Xdawn spatial filters and LDA classifier using the remaining trials of that session. Then, we simulate the online use of that classifier and select a character using the evidence accumulation method described in [Thomas et al., 2014].

## 7.3 Results

We present our preliminary results in figure 7.1. The performance is measured in terms of correctly guessed characters over the total number of characters. We can see that our method produces results that are comparable to the results of a calibrated classifier. Note that both approaches perform poorly for some sessions.

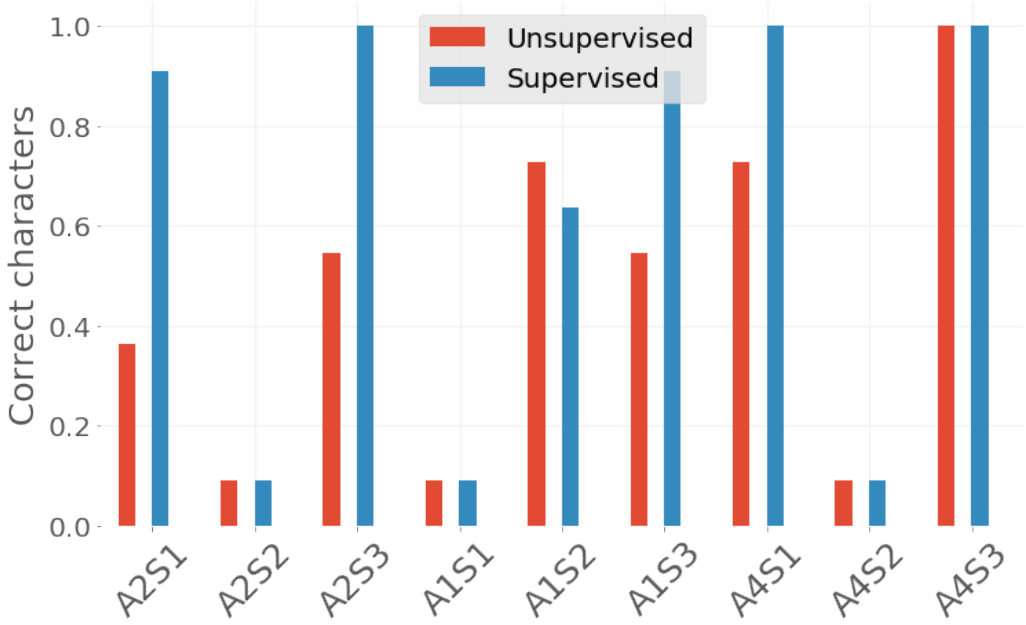



Figure 7.1: Comparison of our unsupervised method and a calibrated classifier for 9 sessions. The results of our method are comparable to the calibration results.

A possible extension of our method is to consider the highest $n$ scores and use a prior to enhance the probability of correctly guessing the character. Such a prior can take the form of a language model. At the same time, we wish to be able to guess the correct character as fast as possible. To assess the feasibility of such an approach, we compute how many times the correct character was found after the 12th, 18th, 24th, 30th and 36th flash, as a function of the number of maximizers of equation 7.1 (top scoring characters). This analysis is displayed on figure 7.2. Note that the the correct character is likely to be in the top 5 scoring characters after only 18 flashes.

## 7.4 Discussion and Conclusion

In this thesis, we have analyzed EEG signal variability and proposed several transfer learning methods and combinations of these methods to deal with this variability. Nevertheless, we saw that EEG signal variability is often so important

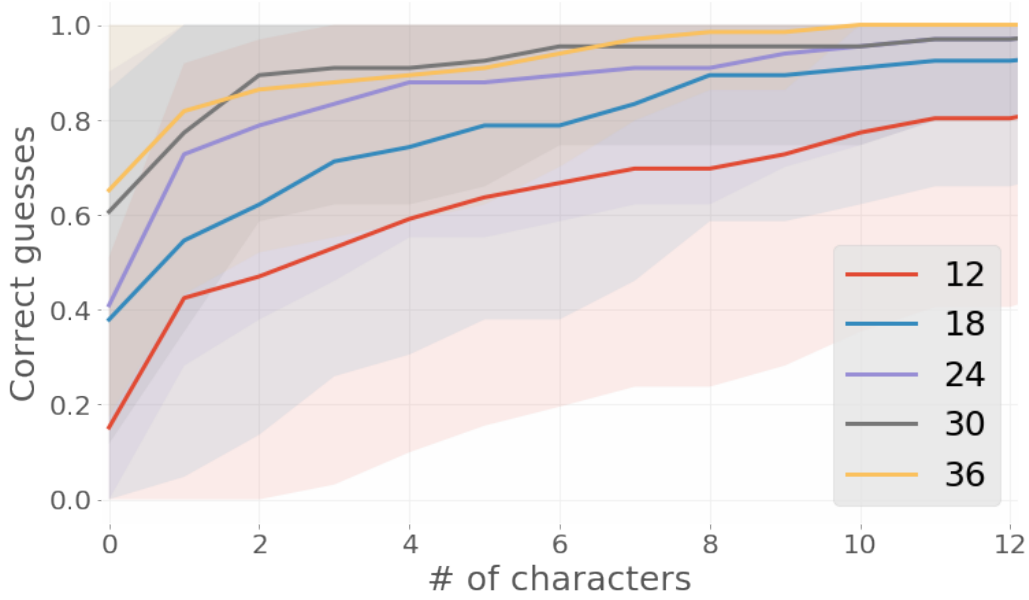



Figure 7.2: Number of times the correct character was found after the 12th, 18th, 24th, 30th and 36th flash, as a function of the number of maximizers of equation 7.1 (top scoring characters), for each session

that the performances of these methods are hindered. At the same time, transfer learning methods depend on the quality of the training set.

Unsupervised classification methods on the other hand only depend on the current session. Such methods are gaining more attention within the community, since their only downside is a sometimes high warm up period. In this chapter, we propose an unsupervised classification method that potentially has a low warm up period. We provide preliminary results that attest to its feasibility. These results, along with existing research on unsupervised classification in BCI, ultimately show that unsupervised methods an attractive alternative to transfer learning. While inter-session variability is not an issue for unsupervised learning methods, they also need to be adapted to deal with intra-session variability. One possible improvement over our method, in addition to word prediction, is the inclusion of learning methods that adapt to variability, such as the methods proposed by Hitziger et al. in [Hitziger et al., 2013] and [Hitziger et al., 2017].

# PART IV

# CONCLUSION

# GENERAL DISCUSSION AND PERSPECTIVES

BCI can offer an alternate means of communication to people with severe motor disabilities. Nevertheless, a number of obstacles remain which forestall their broad use [Wolpaw et al., 2002; Guger et al., 2009a; Lotte et al., 2018]. In this thesis, we focused on the issue of EEG signal variability, which is one of the primary reasons why BCI systems necessitate advanced machine learning and signal processing methods to function [Clerc et al., 2016]. The objective of our research was to study and propose adaptive machine learning methods for ERP-based BCI (taking different variability types into account). Throughout this thesis, we have analyzed different types of EEG variability and investigated their effect on the performance of different adaptive machine learning methods.

First, we detailed an ERP-based BCI system and exposed our problematic. We saw how multiple components interact in a specific paradigm, the P300-Speller. Each component of the system encloses a set of functions. The parameters and functions that apply need to be chosen carefully. We studied the information extraction component in particular, where the EEG signal is converted into a classification result. This component is a pipeline which converts the EEG signal into vector of features, and assigns to this vector a probability of belonging into a class. Feature extraction and classification methods have been extensively researched in the BCI literature [Lotte et al., 2007; Lotte and Congedo, 2016a; Lotte et al., 2018]. However, we saw that, due to EEG signal variability, these methods often generalize poorly across sessions and across subjects. As a result, adaptive approaches are becoming a fundamental part of BCI systems [Mattout et al., 2015; Lotte et al., 2018; Huebner et al., 2018]. In this work, we studied adaptive machine learning methods applied to the P300-Speller. Our focus was on gaining a broad understanding of how these methods deal with EEG signal variability. Note that, our findings can be generalized to other types of ERP-based BCIs.

To accomplish our target, we begun with a literature review of the different types of EEG signal variability in ERP-based BCIs. First, we reviewed ERP component variability, which we saw is typically measured in terms of peak amplitude, peak latency and scalp topography variability. We saw that the ERP components which are the most affected by this type of variability are the P3a and P300 components. Various environmental and physiological factors contribute to that variability. These factors can be grouped according to whether their effects will appear within a BCI session; across BCI sessions of the same user; or across different BCI users. Hence, we saw that all these factors contribute to cross-subject variability, while some of them can appear even in the same session, producing intra-session variability. In addition, the EEG signal contains brain activity that is not time-locked to the stimuli that generate the ERPs. This signal has been proven to contribute to ERP variability as well [Polich, 1997].

We performed an EEG signal variability analysis on two experimental datasets. Both datasets consist of EEG recordings during the calibration session of a P300-Speller. The first dataset was recorded on healthy subjects, while the second dataset was recorded on ALS patients. We selected these particular datasets because the BCI system parameters are nearly identical in both datasets: the same amplifiers were used, the same pre-processing was performed on the EEG signal, and the P300-Speller interface was the same. We quantified the types of variability in the following manner:

- **Intra-session peak amplitude and latency variability.** For each session, we computed the average and standard deviation of the peak amplitude and latency. The values of the standard deviations are a measure of the trial-to-trial variability within each session.

- **Power spectral density of the noise.** After computing and extracting the stimulus responses from the EEG signal of each session, we compute the power spectral density of the residue. This gave us an insight on the type of noise that is present in the EEG signal, and how this noise varies across the two datasets.

- **Scalp topography.** For each dataset, we computed a set of spatial filters using the entire dataset using the algorithm in [Rivet et al., 2011]. These

filters have been designed to extract the ERP component while minimizing the SNR of the signal. They provide us with coefficients that can be interpreted as a scalp topography of the ERP.

Our results in this variability analysis corroborate the literature. The averages and standard deviations of the peak amplitudes and latencies were in accordance with the bibliography. We noted a non-negligible amount of cross-session, cross-subject and intra-session variability. The scalp topography was different for the two datasets and the noise analysis showed that that, while the noise in both datasets and across all sessions is an $1/f^{\alpha}$ process, the value of $\alpha$ was also different, and so was the energy of the noise. Since the same acquisition device was employed and the same pre-processing was performed in both datasets, we can suppose that these factors did not contribute to cross-database differences in our findings. Note that, while the experimental protocol was the same in the two sets of experiments, the environment was different in the two experiments. Such differences could interpret the across dataset variability, in addition to the physiological differences of the subjects and the probable effect of ALS for the patient dataset.

One of the contributions of this thesis is the study of how the aforementioned types of EEG signal variability affect classification performance, when the classification pipeline uses transfer learning methods. Advanced transfer learning methods are specifically designed to counter variability between two domains, who in that context are referred to as the *source* domain and the *target* domain. Here, the *source* domain was always composed of labeled data and used to train the classification method, while the labels of the *target* were strictly only used for evaluation. We closely examined three transfer learning frameworks, who were not selected so much on account of their popularity in the field (although they have all been applied to BCI, see for example [Congedo et al., 2013; Rakotomamonjy and Guigue, 2008; Gayraud et al., 2017]), but more on account of their capacity to deal with EEG signal variability. These frameworks are: 1. Riemannian geometry; 2. optimal transport; and 3. ensemble learning.

Making use of our analysis on EEG variability, we proposed a parameterized model of the EEG signal that incorporates all the aforesaid types of variability. This model allowed us to simulate EEG recordings during P300-Speller experiments, which we used to evaluate the performances of these transfer learning methods as a function of specific variability factors, such as average peak amplitude variability

and intra-session peak amplitude variability. Then, we proposed various classification pipelines that combine these three frameworks and evaluated them on our experimental datasets, interpreting our results though the prism of inter-session and inter-subject variability.

Thus, we saw that Riemannian geometry provides a framework that is robust to affine transformations of the signal. Unsurprisingly, it proved robust to changes in the forward model. Nevertheless, it proved to not be robust to other variability types for high parameter values. In the case of peak latency variability, this result corroborate the findings of Barachant et al. [Barachant and Congedo, 2014] and are attributed to the use of the extended covariance matrix (see chapter 3, section 3.3.1). Regarding high amplitude variability values, both inter-session and intra-session, it can be attributed to the fact that we only added variability to the target response. Let $X$ denote the signal during a session. We consider a simplified version of our model described by equation (4.7):

$$X = X_t + X_n = GS_t + GN_b^t + GS_n + GN_b^n$$

where $S_t$, $S_n \in \mathbb{R}^{I_s \times I_t}$. $N_b^t$ and $N_b^n$ represent background activity that is not time-locked to the stimulus. For simplicity, assume that the interval between two stimuli is chosen so that there are no overlaps between responses to target and non-target stimuli. Then, $N_b^t$ encloses the background activity during target responses and $N_b^n$ the background activity between nontarget responses. We compute the empirical covariance matrices of $X_t$ and $X_n$,

$$\begin{aligned}
\Sigma_t &= \frac{1}{I_t} X_t X t^\intercal \\
&= \frac{1}{I_t}[(GS_t + GN_b^t)(GS_t + GN_b^t)^\intercal] \\
&= \frac{1}{I_t}[GS_t S_t^\intercal G^\intercal + GN_b^t N_b^{t\intercal} G^\intercal + GS_t N_b^\intercal G^\intercal + GN_b S_t^\intercal G^\intercal] \\
&= \frac{1}{I_t} G[S_t S_t^\intercal + N_b^t N_b^{t\intercal} + S_t N_b^{t\intercal} + N_b^t S_t^\intercal] G^\intercal \\
&= \frac{1}{I_t} G M_t G^\intercal
\end{aligned}$$

where $M_= S_t S_t^\intercal + N_b^t N_b^{t\intercal} + S_t N_b^{t\intercal} + N_b^t S_t^\intercal$. $\Sigma_n$ can be computed in the same way. Observe that the terms $S_t N_b^{t\intercal}$ and $N_b^t S_t{}^\intercal$describe the correlation between the

stimulus response and the background noise, which might not be completely uncorrelated. Recall that the Riemannian distance is invariant to affine transformations, hence for two signals $X^A$, $X^B$ and their respective forward models $G^A$, $G^B$ :

$$\begin{aligned} \mathrm{d}(\Sigma_t^A, \Sigma_n^A) &= \mathrm{d}(\frac{1}{I_t} G^A M_t^A G^{A\intercal}, \frac{1}{I_t} G^A M_n^A G^{A\intercal}) \\ &= \mathrm{d}(\frac{1}{I_t} G^B M_t^B G^{B\intercal}, \frac{1}{I_t} G^B M_n^B G^{B\intercal}) \end{aligned}$$

The above equation will only hold if the difference between $M^A$ and $M^B$ is negligible, for both target and nontarget responses. This is not the case when the background noise $N_b$ or the amplitude variability of the target response $S_t$ change across sessions. However, this method is undeniably useful to cope with EEG variability, as proven by experimental results [Lotte et al., 2018].

Optimal transport is invariant to many types transformations in the feature space, due to our choice of the squared euclidean distance as a cost [Villani, 2008; Courty et al., 2017]. In fact, the most appealing property of optimal transport under this choice of cost is that it can find a plan between any two probability distributions [Villani, 2008]. Hence, we can register any two sets of features vectors if we assign a probability to each feature vector. The problem arises from the fact that our data belong to two classes whose inter class separation we wish to preserve. Additionally, we need to ensure that we do not transport features that belong to one class onto features that belong to the other class. Unfortunately, this means that optimal transport will fail for any transformation which causes significant rotations to the feature space. When we work with distributions of feature vectors that result from trials, another downside of the optimal transport framework is that we cannot perform single trial classification at the very beginning of the session, since we require to estimate both *source* and *target* distributions. We proposed two classification methods that use optimal transport. In both these methods, we considered the domain to be composed of high dimensional features. In our experiments, these features were spatiotemporal features in chapter 5 and tangent space projections of covariance matrices in chapter 6. The downside of using high dimensional features it that we cannot estimate their probability distribution. We are hence forced to assume that feature vectors are drawn from a uniform probability distribution. This makes the method less robust to outliers.

Nevertheless, considering the robustness of the optimal transport framework to most EEG variability factors, as we saw in the results of the simulated BCI experiments of chapter 4, we still believe that it will allow to deal with EEG signal variability, provided that these issues are dealt with.

Ensemble learning methods like boosting are effective against noise, but require a large number of samples to provide robust performance. This can prove computationally inefficient. Consider the case where we have a dataset consisting of a substantial amount of calibration sessions. An ensemble method such as bootstrap aggregating would require to produce a large number of bootstraps, each one's cardinality being equal to the total number of feature vectors in the entire dataset. We propose a solution which does not create bootsraps, but instead trains one classifier per available calibration session. While this method does not have the same mathematical properties as other ensemble learning methods such as boosting (chapter 3, section 3.3.3), it allows us to obtain priors on each sample and use them in the final result. In chapter 6, we introduce such a prior in the form of a separability marker. The separability marker allows us to assess the inter-class separation in a sample. We use the separability marker to weigh the decision of each classifier in the aggregation step of the ensemble. Other priors could also be included in the computation of the classification result weighting process. One such prior could be the similarity of the *target* dataset to each training session. Dissimilarity measures such as the Kullback-Leibler divergence or the Wasserstein distance can be used, provided that we have collected a sufficient amount of *target* feature vectors.

Our study of the transfer learning frameworks confirms that it is not trivial to perform cross-session and cross-subject classification. Even when advanced methods are employed and combined, the underlying variability still hinders performances. Moreover, measuring variability is not an obvious task. While we were able to quantify some types of EEG variability, BCI systems are subject to multiple sources of variability that are not easy to track [Clerc et al., 2016]. In our research, this conclusion was supported by the fact that, while we were able to obtain adequate results for cross-subject and cross-session classification, cross-database classification was unachievable. This led us to consider an unsupervised classification method, presented in chapter 7. Our preliminary results demonstrate the feasibility of this approach for the P300-Speller paradigm. While this approach

does not require any training data, possible extensions do not exclude the combined use of a transfer learning method for initialization purposes, as in [Kindermans et al., 2012a].

Our research provided us with considerable information about the effects of EEG variability. Exploring different methodologies allowed us to obtain a greater insight on the type of variability parameters which classification methods need to take into account. In conclusion, we can safely affirm that the future of BCIs lies in their ability to adapt.

# BIBLIOGRAPHY


Aslam, J. A., Popa, R. A., and Rivest, R. L. (2007). On estimating the size and confidence of a statistical audit. *EVT*, 7:8.

Baillet, S., Mosher, J. C., and Leahy, R. M. (2001). Electromagnetic brain mapping. *IEEE Signal processing magazine*, 18(6):14–30.

Barachant, A. (2015). Bci challenge ner 2015. https://github.com/alexandrebarachant/bci-challenge-ner-2015.

Barachant, A., Bonnet, S., Congedo, M., and Jutten, C. (2010). Riemannian geometry applied to bci classification. In *Latent Variable Analysis and Signal Separation*, pages 629–636. Springer.

Barachant, A., Bonnet, S., Congedo, M., and Jutten, C. (2013). Classification of covariance matrices using a riemannian-based kernel for bci applications. *Neurocomputing*, 112:172–178.

Barachant, A. and Congedo, M. (2014). A plug&play p300 bci using information geometry. *arXiv preprint arXiv:1409.0107*, pages 1–9.

Berger, M. (2012). *A panoramic view of Riemannian geometry*. Springer Science & Business Media.

Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U. (1999). When is "nearest neighbor" meaningful? In *International conference on database theory*, pages 217–235. Springer.

Blankertz, B. (2004). Documentation Wadsworth BCI Dataset ( P300 Evoked Potentials ) BCI Competition III Challenge 2004. *Interface*, pages 1–8.

Blankertz, B., Dornhege, G., Müller, K.-R., Schalk, G., Krusienski, D., Wolpaw, J. R., Schlogl, A., Graimann, B., Pfurtscheller, G., Chiappa, S., et al. (2005). Results of the bci competition iii. In *BCI Meeting*.

Blankertz, B., Lemm, S., Treder, M., Haufe, S., and Müller, K. R. (2011). Single-trial analysis and classification of ERP components - A tutorial. *NeuroImage*, 56(2):814–825.

Bledowski, C. (2004). Localizing P300 Generators in Visual Target and Distractor Processing: A Combined Event-Related Potential and Functional Magnetic Resonance Imaging Study. *Journal of Neuroscience*, 24(42):9353–9360.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

Bridson, M. R. and Haefliger, A. (1999). *Metric spaces of non-positive curvature*, volume 319. Springer Science & Business Media.

Cabestaing, F. and Derambure, P. (2016). Physiological markers for controlling active and reactive bcis. In Clerc, M., Bougrain, L., and Lotte, F., editors, *Brain-Computer Interfaces 1: Methods and Perspectives*, chapter 4, pages 209–232. John Wiley & Sons.

Clerc, M., Daucé, E., and Mattout, J. (2016). Adaptive methods in machine learning. In Clerc, M., Bougrain, L., and Lotte, F., editors, *Brain-Computer Interfaces 1: Methods and Perspectives*, chapter 10, pages 209–232. John Wiley & Sons.

Congedo, M., Afsari, B., Barachant, A., and Moakher, M. (2015). Approximate joint diagonalization and geometric mean of symmetric positive definite matrices. *PloS one*, 10(4):e0121423.

Congedo, M., Barachant, A., and Andreev, A. (2013). A new generation of brain-computer interface based on riemannian geometry. *arXiv preprint arXiv:1310.8115*.

Congedo, M., Goyat, M., Tarrin, N., Ionescu, G., Varnet, L., Rivet, B., Phlypo, R., Jrad, N., Acquadro, M., and Jutten, C. (2011). "brain invaders": a prototype of an open-source p300-based video game working with the openvibe platform. In *5th International Brain-Computer Interface Conference 2011 (BCI 2011)*, pages 280–283.

Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. (2017). Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865.

Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, pages 2292–2300.

Dalebout, S. D. and Robey, R. R. (1997). Comparison of the intersubject and intrasubject variability of exogenous and endogenous auditory evoked potentials. *J Am Acad Audiol*, 8:342–354.

Delorme, A., Miyakoshi, M., Jung, T.-P., and Makeig, S. (2015). Grand average erp-image plotting and statistics: A method for comparing variability in event-related single-trial eeg activities across subjects and conditions. *Journal of neuroscience methods*, 250:3–6.

Dinteren, R., Arns, M., Jongsma, M. L. A., and Kessels, R. P. C. (2014). P300 development across the lifespan: A systematic review and meta-analysis. *PLoS ONE*, 9(2).

Donchin, E., Ritter, W., McCallum, W. C., et al. (1978). Cognitive psychophysiology: The endogenous components of the erp. *Event-related brain potentials in man*, pages 349–411.

Donchin, E., Spencer, K. M., and Wijesinghe, R. (2000). The mental prosthesis: Assessing the speed of a P300-based brain- computer interface. *IEEE Transactions on Rehabilitation Engineering*, 8(2):174–179.

Farwell, L. A. and Donchin, E. (1988). Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and clinical Neurophysiology*, 70(6):510–523.

Fazli, S., Popescu, F., Danóczy, M., Blankertz, B., Müller, K.-R., and Grozea, C. (2009). Subject-independent mental state classification in single trials. *Neural networks*, 22(9):1305–1312.

Förstner, W. and Moonen, B. (2003). A metric for covariance matrices. In *Geodesy-The Challenge of the 3rd Millennium*, pages 299–309. Springer.

Gayraud, N. T., Foy, N., and Clerc, M. (2016). A separability marker based on high-dimensional statistics for classification confidence assessment. In *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*, pages 003193–003198. IEEE.

Gayraud, N. T., Foy, N., and Clerc, M. (2017). A Separability Marker based on high-dimensional statistics for classification confidence assessment. In *2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016*, pages 3193–3198.

Gonsalvez, C. J., Barry, R. J., Rushby, J. A., and Polich, J. (2007). Target-to-target interval, intensity, and p300 from an auditory single-stimulus task. *Psychophysiology*, 44(2):245–250.

Gramfort, A., Keriven, R., and Clerc, M. (2010). Graph-based variability estimation in single-trial event-related neural responses. *IEEE Transactions on Biomedical Engineering*, 57(5):1051–1061.

Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Goj, R., Jas, M., Brooks, T., Parkkonen, L., et al. (2013). Meg and eeg data analysis with mne-python. *Frontiers in neuroscience*, 7:267.

Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Parkkonen, L., and Hämäläinen, M. S. (2014). Mne software for processing meg and eeg data. *Neuroimage*, 86:446–460.

Guger, C., Daban, S., Sellers, E., Holzner, C., Krausz, G., Carabalona, R., Gramatica, F., and Edlinger, G. (2009a). How many people are able to control a p300-based brain–computer interface (bci)? *Neuroscience letters*, 462(1):94–98.

Guger, C., Daban, S., Sellers, E., Holzner, C., Krausz, G., Carabalona, R., Gramatica, F., and Edlinger, G. (2009b). How many people are able to control a P300-based brain-computer interface (BCI)? *Neuroscience Letters*, 462(1):94–98.

Guy, V., Soriani, M.-H., Bruno, M., Papadopoulo, T., Desnuelle, C., and Clerc, M. (2018). Brain computer interface with the p300 speller: Usability for disabled

people with amyotrophic lateral sclerosis. *Annals of physical and rehabilitation medicine*, 61(1):5–11.

Hämäläinen, M., Hari, R., Ilmoniemi, R. J., Knuutila, J., and Lounasmaa, O. V. (1993). Magnetoencephalography theory, instrumentation, and applications to noninvasive studies of the working human brain. *Reviews of Modern Physics*.

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

Hitziger, S., Clerc, M., Saillet, S., Bénar, C., and Papadopoulo, T. (2017). Adaptive waveform learning: a framework for modeling variability in neurophysiological signals. *IEEE Transactions on Signal Processing*, 65(16):4324–4338.

Hopcroft, J. and Kannan, R. (2014). *Foundations of Data Science*.

Hübner, D., Verhoeven, T., Schmid, K., Müller, K.-R., Tangermann, M., and Kindermans, P.-J. (2017). Learning from label proportions in brain-computer interfaces: Online unsupervised learning with guarantees. *PloS one*, 12(4):e0175856.

Huebner, D., Verhoeven, T., Mueller, K.-R., Kindermans, P.-J., and Tangermann, M. (2018). Unsupervised learning for brain-computer interfaces based on event-related potentials: Review and online comparison [research frontier]. *IEEE Computational Intelligence Magazine*, 13(2):66–67.

Isreal, J. B., Chesney, G. L., Wickens, C. D., and Donchin, E. (1980). P300 and tracking difficulty: Evidence for multiple resources in dual-task performance. *Psychophysiology*, 17(3):259–273.

Iturrate, I., Grizou, J., Omedes, J., Oudeyer, P. Y., Lopes, M., and Montesano, L. (2015). Exploiting task constraints for self-calibrated brain-machine interface control using error-related potentials. *PLoS ONE*, 10(7):1–15.

Jayaram, V., Alamgir, M., Altun, Y., Scholkopf, B., and Grosse-Wentrup, M. (2016). Transfer learning in brain-computer interfaces. *IEEE Computational Intelligence Magazine*, 11(1):20–31.

Jeunet, C., N'Kaoua, B., and Lotte, F. (2016). Advances in user-training for mental-imagery-based bci control: Psychological and cognitive factors and their neural correlates. In *Progress in brain research*, volume 228, pages 3–35. Elsevier.

Jung, T. P., Makeig, S., Westerfield, M., Townsend, J., Courchesne, E., and Sejnowski, T. J. (2001). Analysis and visualization of single-trial event-related potentials. *Human Brain Mapping*, 14(3):166–185.

Kantorovitch, L. (1958). On the translocation of masses. *Management Science*, 5(1):1–4.

Katayama, J. and Polich, J. (1999). Auditory and visual P300 topography from a 3 stimulus paradigm. *Clinical Neurophysiology*, 110(3):463–468.

Kaufmann, T., Schulz, S., Grünzinger, C., and Kübler, A. (2011). Flashing characters with famous faces improves erp-based brain–computer interface performance. *Journal of neural engineering*, 8(5):056016.

Kindermans, P.-J., Schreuder, M., Schrauwen, B., Müller, K.-R., and Tangermann, M. (2014). True zero-training brain-computer interfacing–an online study. *PloS one*, 9(7):e102504.

Kindermans, P.-J., Verschore, H., Verstraeten, D., and Schrauwen, B. (2012a). A P300 BCI for the Masses: Prior Information Enables Instant Unsupervised Spelling. *Advances in Neural Information Processing Systems 25*, pages 719–727.

Kindermans, P.-J., Verstraeten, D., and Schrauwen, B. (2012b). A bayesian model for exploiting application constraints to enable unsupervised training of a p300-based bci. *PloS one*, 7(4):e33758.

Knight, P. A. (2008). The sinkhorn–knopp algorithm: convergence and applications. *SIAM Journal on Matrix Analysis and Applications*, 30(1):261–275.

Kybic, J., Clerc, M., Abboud, T., Faugeras, O., Keriven, R., and Papadopoulo, T. (2005). A common formalism for the Integral formulations of the forward EEG problem. *IEEE Transactions on Medical Imaging*, 24(1):12–28.

Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., and Yger, F. (2018). A review of classification algorithms for eeg-based

brain–computer interfaces: a 10 year update. *Journal of neural engineering*, 15(3):031005.

Lotte, F. and Congedo, M. (2016a). *Brain-computer interfaces. Vol 1, foundations and methods*, chapter EEG Feature Extraction, pages 130–131. London ISTE Ltd Hoboken.

Lotte, F. and Congedo, M. (2016b). Eeg feature extraction. In Clerc, M., Bougrain, L., and Lotte, F., editors, *Brain-Computer Interfaces 1: Methods and Perspectives*, chapter 7, pages 209–232. John Wiley & Sons.

Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., and Arnaldi, B. (2007). A review of classification algorithms for eeg-based brain–computer interfaces. *Journal of neural engineering*, 4(2):R1.

Lotte, F. and Guan, C. (2011). Regularizing common spatial patterns to improve bci designs: unified theory and new algorithms. *IEEE Transactions on biomedical Engineering*, 58(2):355–362.

Lotte, F. and Jeunet, C. (2015). Towards improved bci based on human learning principles. In *Brain-Computer Interface (BCI), 2015 3rd International Winter Conference on*, pages 1–4. IEEE.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

Makeig, S., Debener, S., Onton, J., and Delorme, A. (2004). Mining event-related brain dynamics. *Trends in Cognitive Sciences*, 8(5):204–210.

Mattout, J., Perrin, M., Bertrand, O., and Maby, E. (2015). Improving BCI performance through co-adaptation: Applications to the P300-speller. *Annals of Physical and Rehabilitation Medicine*, 58(1):23–28.

McCarthy, G. and Donchin, E. (1976). The effects of temporal and event uncertainty in determining the waveforms of the auditory event related potential (erp). *Psychophysiology*, 13(6):581–590.

Michalewski, H., Prasher, D., and Starr, A. (1986). Latency Variability And Temporal Interrelationships Of The Auditory Event-Related Potentials (N1, P2, N2, And

P3) In Normal Subjects. *Electroencephalography and Clinical Neurophysiology/ Evoked Potentials*, 64(1):56–71.

Mladenovic, J., Mattout, J., and Lotte, F. (2018). A generic framework for adaptive EEG-based BCI training and operation. In Nam, C. S., Nijholt, A., and Lotte, F., editors, *Brain-Computer Interfaces Handbook: Technological and Theoretical Advances*, chapter 31, pages 595–613. CRC Press, 1 edition.

Niedermeyer, E. et al. (2005). The normal eeg of the waking adult. *Electroencephalography: Basic principles, clinical applications, and related fields*, 167:155–164.

Panicker, R. C., Puthusserypady, S., and Sun, Y. (2010). Adaptation in p300 brain–computer interfaces: A two-classifier cotraining approach. *IEEE Transactions on Biomedical Engineering*, 57(12):2927–2935.

Papageorgakis, C. (2017). *Patient specific conductivity models: characterization of the skull bones.* PhD thesis, Université Côte d'Azur.

Pennec, X. (2009). Statistical computing on manifolds: from riemannian geometry to computational anatomy. In *Emerging Trends in Visual Computing*, pages 347–386. Springer.

Pennec, X., Fillard, P., and Ayache, N. (2006). A riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66.

Perrin, M., Maby, E., Daligault, S., Bertrand, O., and Mattout, J. (2012). Objective and subjective evaluation of online error correction during p300-based spelling. *Advances in Human-Computer Interaction*, 2012:4.

Polich, J. (1997). On the relationship between eeg and p300: individual differences, aging, and ultradian rhythms. *International journal of psychophysiology*, 26(1-3):299–317.

Polich, J. (2009). Updating P300:An Integrative Theory of P3a and P3b. *Clin Neurophysiol*, 118(10):2128–2148.

Polich, J. and Kok, A. (1995). Cognitive and biological determinants of P300: an integrative review. *Biological Psychology*, 41(2):103–146.

Pritchard, W. S. (1981). Psychophysiology of p300. *Psychological bulletin*, 89(3):506.

Rakotomamonjy, A. and Guigue, V. (2008). BCI competition III: Dataset II-ensemble of SVMs for BCI P300 speller. *IEEE Transactions on Biomedical Engineering*, 55(3):1147–1154.

Rivet, B., Cecotti, H., Souloumiac, A., Maby, E., and Mattout, J. (2011). Theoretical analysis of xdawn algorithm: application to an efficient sensor selection in a p300 bci. In *19th European Signal Processing Conference (EUSIPCO 2011)*, pages 1382–1386.

Rivet, B., Souloumiac, A., Attina, V., and Gibert, G. (2009). xdawn algorithm to enhance evoked potentials: application to brain–computer interface. *IEEE Transactions on Biomedical Engineering*, 56(8):2035–2043.

Saavedra, C. and Bougrain, L. (2012). Processing stages of visual stimuli and event-related potentials. In *The NeuroComp/KEOpS'12 workshop*.

Santambrogio, F. (2015). Optimal transport for applied mathematicians. *Birkäuser, NY*, pages 99–102.

Sarvas, J. (1987). Basic mathematical and electromagnetic concepts of the biomagnetic inverse problem. *Physics in Medicine and Biology*.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244.

Solomon, J., De Goes, F., Peyré, G., Cuturi, M., Butscher, A., Nguyen, A., Du, T., and Guibas, L. (2015). Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (TOG)*, 34(4):66.

Souloumiac, A. and Rivet, B. (2013). Improved estimation of eeg evoked potentials by jitter compensation and enhancing spatial filters. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 1222–1226. IEEE.

Sugiyama, M. and Kawanabe, M. (2012). *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press.

Sun, S., Zhang, C., and Zhang, D. (2007). An experimental evaluation of ensemble methods for eeg signal classification. *Pattern Recognition Letters*, 28(15):2157–2163.

Sutton, S., Braren, M., Zubin, J., and John, E. (1965). Evoked-potential correlates of stimulus uncertainty. *Science*, 150(3700):1187–1188.

Thomas, E., Daucé, E., Devlaminck, D., Mahé, L., Carpentier, A., Munos, R., Perrin, M., Maby, E., Mattout, J., Papadopoulo, T., et al. (2014). Coadapt p300 speller: optimized flashing sequences and online learning. In *6th International Brain Computer Interface Conference*.

Thomas, E., Dyson, M., and Clerc, M. (2013). An analysis of performance evaluation for motor-imagery based bci. *Journal of neural engineering*, 10(3):031001.

Townsend, G., LaPallo, B., Boulay, C., Krusienski, D., Frye, G., Hauser, C., Schwartz, N., Vaughan, T., Wolpaw, J. R., and Sellers, E. (2010). A novel p300-based brain–computer interface stimulus presentation paradigm: moving beyond rows and columns. *Clinical Neurophysiology*, 121(7):1109–1120.

Verhoeven, T., Hübner, D., Tangermann, M., Müller, K.-R., Dambre, J., and Kindermans, P.-J. (2017). Improving zero-training brain-computer interfaces by mixing model estimators. *Journal of neural engineering*, 14(3):036021.

Vidal, J. J. (1973). Toward direct brain-computer communication. *Annual review of Biophysics and Bioengineering*, 2(1):157–180.

Villani, C. (2008). *Optimal transport: old and new*, volume 338. Springer Science & Business Media.

Walhovd, K. B. and Fjell, A. M. (2002). One-year test–retest reliability of auditory erps in young and old adults. *International Journal of Psychophysiology*, 46(1):29–40.

Ward, L. M. (2002). *Dynamical cognitive science*. MIT press.

Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big Data*, 3(1):9.

Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., and Vaughan, T. M. (2002). Brain-computer interfaces for communication and control. *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology*, 113(6):767–91.

Wolters, C. H., Grasedyck, L., and Hackbusch, W. (2004). Efficient computation of lead field bases and influence matrix for the FEM-based EEG and MEG inverse problem. *Inverse Problems*.

Yagi, Y., Coburn, K. L., Estes, K. M., and Arruda, J. E. (1999). Effects of aerobic exercise and gender on visual and auditory P300, reaction time, and accuracy. *European Journal of Applied Physiology and Occupational Physiology*, 80(5):402–408.

Zanini, P., Congedo, M., Jutten, C., Said, S., and Berthoumieu, Y. (2018). Transfer learning: a riemannian geometry framework with applications to brain–computer interfaces. *IEEE Transactions on Biomedical Engineering*, 65(5):1107–1116.

# LIST OF PUBLICATIONS

**Journals**

1. Gayraud, N., Clerc, M., Transfer Learning Methods Applied to Brain Computer Interfaces to Handle EEG Variability. *(in submission).*

2. Gayraud, N., Rakotomamonjy, A., Clerc, M., Optimal Transport Applied to Brain Computer Interfaces. *(in preparation).*

3. Gayraud, N., Maksymenko, K., Clerc, M., *Results of chapter 7. (in preparation).*

4. Rimbert, S., Gayraud, N., Bougrain, L., Clerc, M., & Fleck, S., Can a Subjective Questionnaire used as Brain-Computer Interface performance Predictor? Frontiers in Human Neuroscience, Submitted, under review.

5. Rimbert, S., Riff, P., Gayraud, N., Bougrain, L., Median Nerve Stimulation Based BCI: a New Approach to Detect Intraoperative Awareness During General Anesthesia. Submitted, under review.

**Conferences**

1. Lindig-León, C., Gayraud, N., Bougrain, L., & Clerc, M. Comparison of hierarchical and non-hierarchical classification for motor imagery based bci systems. In The Sixth International Brain-Computer Interfaces Meeting. May 2016

2. Gayraud, N., Foy, N., & Clerc, M. A Separability Marker Based on High-Dimensional Statistics for Classification Confidence Assessment. In IEEE International Conference on Systems, Man, and Cybernetics October 9-12. Octobre 2016

3. Gayraud, N., Clerc, M. & Rakotomamonjy, A. Optimal Transport Applied To Transfer Learning For P300 Detection. In the 7th Graz Brain-Computer Interface Conference 2017, September 18th – 22nd, 2017, Graz, Austria

4. Turi, F., Gayraud, & Clerc, M. Zero-Calibration C-Vep Bci Using Word Prediction: A Proof Of Concept. In The Seventh International Brain-Computer Interfaces Meeting. June 2018

5. Rimbert, S., Gayraud, N., Clerc, M., Fleck, S., & Bougrain, L. Can the MIQ-RS questionnaire be used to estimate the performance of a MI-based BCI?. In The Seventh International Brain-Computer Interfaces Meeting. June 2018

6. Gayraud, N. and Clerc, M. Covariate Shift Adaptation using Optimal Transport. In The International Conference on Mathematical Neuroscience, ICMNS, June 2018

7. Gayraud, N., Gallardo, G., Clerc, M. & Wasserman, D., Solving the Cross-Subject Parcel Matrching Problem: Comparing four Methods Using Extrinsic Connectivity, Organization for Human Brain Mapping (OHBM 2018), June 2018, Singapore, Singapore

8. Gallardo, G., Gayraud, N., Deriche, R., Clerc, M., Deslauriers-Gauthier, S., & Wasserman, D., Solving the Cross-Subject Parcel Matching Problem Using Optimal Transport, International Conference On Medical Image Computing & Computer Assisted Intervention - September 16-20 2018, Granada

**Workshops**

1. Gayraud, N., Clerc, M. & Rakotomamonjy, A. “Transport optimal appliqué aux interfaces cerveau machine”. Rencontre C@UCA 2017, 6-8 juin 2017 Fréjus (France).

2. Gayraud, N., Clerc, M. & Rakotomamonjy, A. “Transport optimal appliqué au P300 speller”. Journée Jeunes Chercheurs en Interfaces Cerveau Ordinateur et Neurofeedback (JJC-ICON), 15 juin 2017 Bordeaux (France)

3. Gallardo, G., Gayraud, N., Clerc, M. & Wasserman, D., “Matching parcellations using Optimal Transport: a proof of concept”. Computational Brain

Connectivity Mapping – Winter School Workshop. Novembre 2017, Juan les Pins, (France)

4. Rimbert, S., Gayraud, N., Clerc, M., Fleck, S., & Bougrain, L., “Can the MIQ-RS questionnaire be used to estimate the performance of a MI-based BCI?”. Journée Jeunes Chercheurs en Interfaces Cerveau Ordinateur et Neurofeedback (JJC-ICON), 18 & 19 Avril 2018, ISAE-SUPAERO, Toulouse, (France)

5. Turi, F., Gayraud, N. & Clerc, M., “Zero-Calibration C-Vep Bci Using Word Prediction: A Proof Of Concept”. Journée Jeunes Chercheurs en Interfaces Cerveau Ordinateur et Neurofeedback (JJC-ICON), 18 & 19 Avril 2018, ISAE-SUPAERO, Toulouse, (France)

6. Gayraud, N. & Clerc, M., “Optimal Transport Applied to Motor Imagery Based BCI”. Journée Jeunes Chercheurs en Interfaces Cerveau Ordinateur et Neurofeedback (JJC-ICON), 18 & 19 Avril 2018, ISAE-SUPAERO, Toulouse, (France)

7. Gayraud, N. & Clerc, M., “Covariate shift adaptation using Optimal Transport”. Rencontre C@UCA 2018, 14-15 juin 2018 Fréjus (France).

8. Turi, F., Gayraud, N. & Clerc, M., “Word Spelling using Zero-Calibration C-Vep Brain Computer Interface”. Rencontre C@UCA 2018, 14-15 juin 2018 Fréjus (France).

# APPENDIX
# CONTRIBUTIONS OUTSIDE THE SCOPE OF THIS THESIS

# Covariate Shift Adaptation using Optimal Transport

Nathalie T.H. Gayraud*, nathalie.gayraud@inria.fr Maureen Clerc*, maureen.clerc@inria.fr

## Introduction

Consider a supervised classification problem in which we have an existing dataset $X^e = \{x_i^e\}_{i=1}^{N_e}$, $x_i^e \in \Omega^e \subset \mathbb{R}^d$ and a corresponding set of labels $\{y_i^e\}_{i=1}^{N_e}$, $y_i^e \in \mathbb{R}$, which follow a joint probability distribution $p_e(x^e, y^e)$. Using $X^e$ to train a classifier we obtain a prediction function $f : \mathbb{R}^d \to \mathbb{R}$. One typically seeks to recover the unknown labels $\{y_j^n\}_{j=1}^{N_n}$ of a new dataset $\{x_j^n\}_{j=1}^{N_n}$, $x_j^n \in \Omega^n \subset \mathbb{R}^d$. Under the assumption that $p_e(x^e) = p_n(x^n)$, we would use $f$ to recover the labels of the new dataset. However, due to a phenomenon known as covariate shift, this might not be true. Such variability could be attributed to physiological differences between subjects, changes in the acquisition process, environmental noise, etc. We propose a transfer learning solution to deal with covariate shift by treating it as a domain adaptation problem and use Optimal Transport (OT) [1] to solve it, as in [2]. We proceed by formulating the problem, present our proposed solution, and show an application to Brain Computer Interfaces (BCI).

## Solving covariate shift with Optimal Transport

Assuming that the data has undergone covariate shift, that is, a transformation $\mathbf{T} : \Omega^e \to \Omega^n$, such that $p_e(x^e) = p_e(\mathbf{T}^{-1}(x^n)) \neq p_n(x^n)$ and $p_e(y^e|x^e) = p_e(y^e|\mathbf{T}^{-1}(x^n)) = p_n(y^n|x^n)$, we propose to use discrete regularized OT with class labels [2] to transport the new data onto the domain of the existing ones. First, we compute a transport plan between the probability distributions of the two datasets, which we then use to update the location of the new dataset. Since $p_e(x^e)$ and $p_n(x^n)$ are not known, we use the two corresponding empirical distributions $\mu_e = \sum_{i=1}^{N^e} p_i^e \delta_{x_i^e}$ and $\mu_n = \sum_{j=1}^{N^n} p_j^n \delta_{x_j^n}$ instead, where $p_i^e$ and $p_j^n$ are the probability masses associated to each sample. In this work, supposing that $d$ is high, we assume a uniform probability distribution over all samples, $p_i^e = \frac{1}{N^e}$ and $p_j^n = \frac{1}{N^n}$. We compute the transport plan $\gamma_0$ such that, if $\mathcal{B} = \left\{\gamma \in (\mathbb{R}^+)^{N_e \times N_n} \mid \gamma \mathbf{1}_{N_n} = \frac{1}{N^e}\mathbf{1}_{N_e}, \gamma^{\mathbf{T}}\mathbf{1}_{N_e} = \frac{1}{N^n}\mathbf{1}_{N_n}\right\}$, where $\mathbf{1}_N$ is an $N$-dimensional vector of ones, $\gamma_0 \in \mathcal{B}$ is the output of the following minimization problem.

$$\gamma_0 = \underset{\gamma \in \mathcal{B}}{\arg\min}\ \langle \gamma, C \rangle_F + \lambda \sum_{i,j} \gamma(i,j)\ \log\ \gamma(i,j) + \eta \sum_j \sum_c \|\gamma(\mathcal{I}_c, j)\|_2$$

Matrix $C_{i,j} = \|x_i^e - x_j^n\|_2^2$ represents the cost of moving probability mass from location $x_j^n$ to location $x_i^e$. $\mathcal{I}_c$ encloses the indices of the rows that correspond to the existing samples of class $c$. The first regularization term allows us to solve this optimization problem using the very efficient Sinkhorn-Knopp algorithm. Since we are performing supervised classification, the second regularization term induces a group-sparse penalty on the columns of $\gamma_0$ ensuring that new samples will give mass only to existing samples of the same class [2]. Finally, we compute the location of the new data with barycentric mapping $\hat{\mathbf{X}}^n = \text{diag}(\gamma_0^\top \mathbf{1}_{N_e})^{-1} \gamma_0^\top \mathbf{X}^e$, where $\hat{\mathbf{X}}^n$ and $\mathbf{X}^e$ are matrices whose rows are the vectors of the transported new and existing datasets respectively.

## Application to Brain Computer Interfaces

We perform offline experiments using Database 2a of the BCI competition IV, which includes EEG recordings of 9 healthy subjects performing imagined movements of the left hand and the tongue. We carry out all possible pairs of experiments, in which one subject is used to train the BCI (existing data) and another to test it (new data). The same process was repeated for a state-of-the art classification method in BCI [3], in which the training dataset is also used in the computation of spatial filters that enhance the class separation. In both cases, the features are classified using a Linear Discriminant Analysis classifier. On average, classifying the data after transporting the new dataset achieves a 62.7 % classification accuracy compared to 54.93 % for the state-of-the art method. To assert the statistical significance of the difference, we performed Welch's t-test, which revealed a p-value equal to $p \approx 10^{-6}$. In conclusion, these findings suggest that Optimal Transport is a promising method for solving the issue of covariate shift.

## References

[1] F. Santambrogio. Optimal transport for applied mathematicians. Birkuser, NY, 99-102., 2015.

[2] N. Courty, R. Flamary, D. Tuia, & A. Rakotomamonjy. *Optimal transport for domain adaptation.* IEEE transactions on pattern analysis and machine intelligence, 39 (9), pp 1853-1865, 2017

[3] Y. Wang, S. Gao, & X. Gao. *Common spatial pattern method for channel selection in motor imagery based brain-computer interface.* In Engineering in medicine and biology society, 2005.

*Université Côte d'Azur, Inria Sophia-Antipolis-Méditerranée, France

# Optimal Transport Applied to Motor Imagery based BCI

Nathalie T.H. Gayraud[1] Maureen Clerc[1]

[1]*Université Côte d'Azur, Inria Sophia-Antipolis-Méditerranée, France*

The design of zero-calibration Brain Computer Interfaces (BCI) is a critical research topic. Several transfer learning approaches have been proposed to confront this issue [1]. In particular, for multiclass Motor Imagery-based BCI, the BCI seeks to extract discriminative features from the recorded EEG signal. In subject-to-subject transfer learning strategies, these features are learned from existing labeled sessions recorded with a different subject than the one using the BCI. Nevertheless, due to a phenomenon known as covariate shift [2], the domain of new features can be different from the domain of the ones learned during training. Thus, the resulting classifier may fail to generalize well across subjects.

We solve this issue by using discrete regularized Optimal Transport (OT) with class labels [3,4]. OT theory studies the problem of transporting probability mass between distributions with respect to a cost function. The method learns a transformation which is assumed to have caused the shift between the two domains and applies its inverse to transport new features onto the domain of the existing ones.

We perform offline experiments using Database 2a of BCI competition IV. Considering all possible pairings of subjects where one subject is used to train the BCI and another to test it, we apply OT along with two transfer learning methods. The first consists of learning multiclass CSP features [5], and using them to train an LDA classifier. The second is a Riemannian Geometry-based classifier [6], as they have been effective in countering the subject-to-subject variability [1]. Since the classification task is to separate 4 imagined movements, we use Cohen's kappa value as a performance measure. Our results, displayed in Figure 1, show that OT improves the average results of both techniques. These results demonstrate that OT is a powerful pre-processing tool that can enhance the result of transfer-learning approaches.

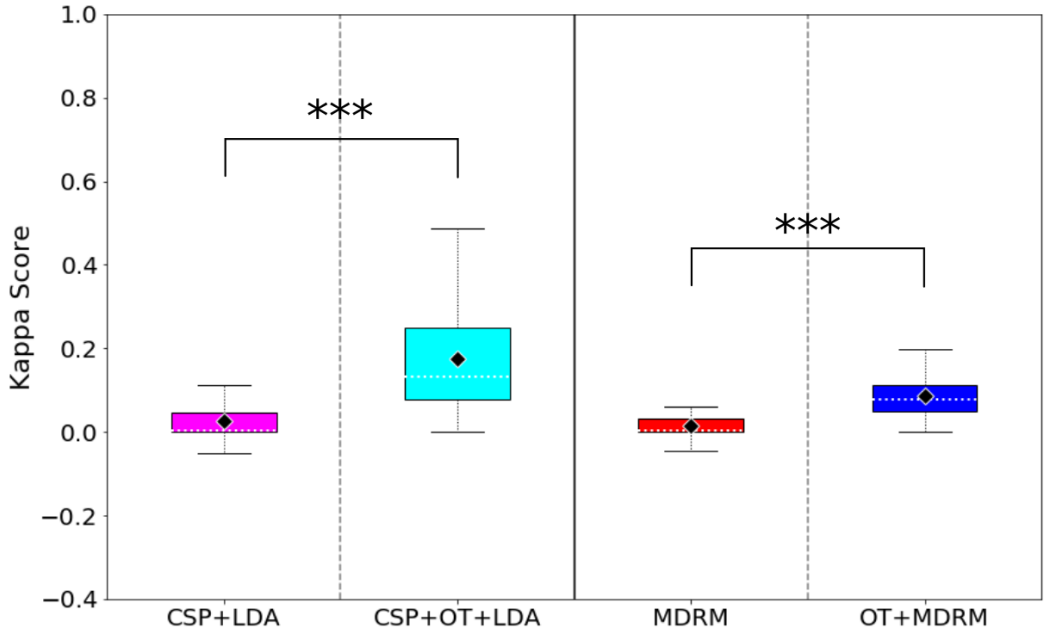



**Figure 1.** Results from subject-to-subject transfer learning offline experiments. Each box shows the result of 72 pairwise experiments. On the left, the first two boxes reflect the performance of a Linear Discriminant Analysis (LDA) classifier, trained with features extracted from 10 multiclass Common Spatial Patterns (CSP) filters [5], without and with OT. Transporting the feature vectors of the test set significantly improves the generalization capacity of the classifier. On the right, a similar experiment, where OT is applied before the Minimum Distance to Riemannian Mean classification algorithm [6]. Once more, the results are greatly improved after the application of our OT method.

## REFERENCES

[1] Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., & Yger, F. (2018). A Review of Classification Algorithms for EEG-based Brain-Computer Interfaces: A 10-year Update. *Journal of neural engineering*.

[2] Clerc, M., Bougrain, L., & Lotte, F. (Eds.). (2016). Adaptive Methods in Machine Learning. In *Brain-Computer Interfaces 1: Methods and Perspectives*. John Wiley & Sons.

[3] Courty, N., Flamary, R., Tuia, D., & Rakotomamonjy, A. (2017). Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, *39*(9), 1853-1865.

[4] Gayraud, N. T., Rakotomamonjy, A., & Clerc, M. (2017, September). Optimal Transport Applied to Transfer Learning For P300 Detection. In *7th Graz Brain-Computer Interface Conference 2017*.

[5] Grosse-Wentrup, M., & Buss, M. (2008). Multiclass common spatial patterns and information theoretic feature extraction. *IEEE transactions on Biomedical Engineering*, *55*(8), 1991-2000.

[6] Barachant, A., Bonnet, S., Congedo, M., & Jutten, C. (2010, September). Riemannian geometry applied to BCI classification. In *International Conference on Latent Variable Analysis and Signal Separation* (pp. 629-636). Springer, Berlin, Heidelberg.

# *ZERO-CALIBRATION C-VEP BCI USING WORD PREDICTION: A PROOF OF CONCEPT*

*Federica Turi - Nathalie Gayraud – Maureen Clerc*

*Inria Sophia Antipolis-Mediterranée, Université Côte d'Azur France*

*Introduction:*

Brain Computer Interfaces (BCIs) based on visual evoked potentials (VEP) [1] allow to spell from a keyboard of flashing characters. Among VEP BCIs, code-modulated visual evoked potentials (c-VEPs) are designed for high-speed communication [2]. In c-VEPs, all characters flash simultaneously. In particular, each character flashes according to a predefined 63-bit binary sequence (m-sequence), circular-shifted by a different time lag. For a given character, the m-sequence evokes a VEP in the electroencephalogram (EEG) of the subject [3], which can be used as a template. This template is obtained during a calibration phase at the beginning of each session. Then, the system outputs the desired character after a predefined number of repetitions by estimating its time lag with respect to the template. Our work avoids the calibration phase, by extracting from the VEP relative lags between successive characters, and predicting the full word using a dictionary.

*Material, Methods and Results:*

Using the time-windowed EEG generated while the user is gazing at the first character, we compute the average response $X_a$ over N repetitions. Since the system has not been calibrated, the first character cannot be displayed. For the second character, we again compute the average response, and shift it by $l \cdot s$ time samples where s is the time lag between two consecutive characters. This produces L shifted averages $X_l$, $l = \{0 , \dots , L-1\}$, where L is the number of characters on the keyboard. Using the lag $l = \text{argmax}_l\{\text{corr}(X_a,X_l)\}$ which produces the maximum correlation to the initial average response, we compute the relative position of this character with respect to the first. Finally, we generate all valid pairs of characters separated by l , and only retain those corresponding to the beginning of valid words within a dictionary. These word beginnings are displayed on the screen as feedback. We repeat this procedure for the following characters, until we are left with a single word (Fig.1). At that moment, we will have recovered the original letter, and the absolute position of $X_a$ can be thereafter used during the computation of the time lag. We conducted offline experiments using the database presented in [3], composed of 9 subjects, 2 sessions per subject, and 640 trials per session. The signals were pre-processed using a Butterworth filter between 1 and 15 Hz. Each experiment consisted of spelling a 3-letter word and was parameterized by the number of repetitions. We repeated the experiment 100 times by simulating the spelling of 3-letter words that we randomly selected among 1014 3-letter English words. We compared our results to a calibrated experiment (Fig.1b and 1c), where we used N repetitions of three characters to compute an average absolute response $X_a$, and performed the same pre-processing as in [3].

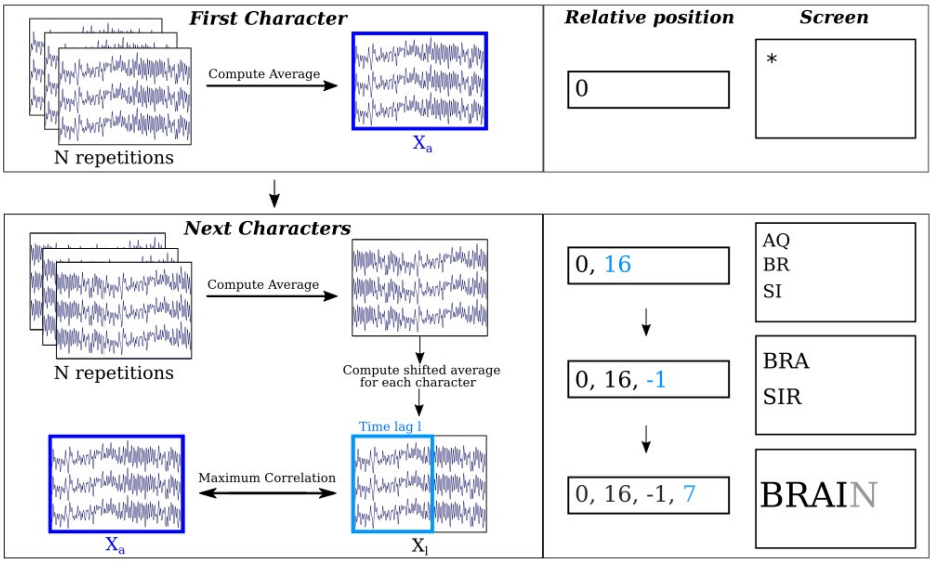
*Discussion:*

Our zero-calibration method achieves good accuracy, even with only 8 repetitions. In comparison, the experiments preceded by calibration reach a good accuracy after 12 repetitions of the m-code flashes. On Fig. 1b we distinguish two groups of subjects: in green, those that perform well, reaching on average an accuracy that exceeds 75% after 12 repetitions; in red, those whose performance does not produce a accuracy higher than 50%. This trend is also seen in the results of [3]. We keep the same color coding on Fig. 1c. While some subjects reach accuracy values equal to 100% after the calibration,

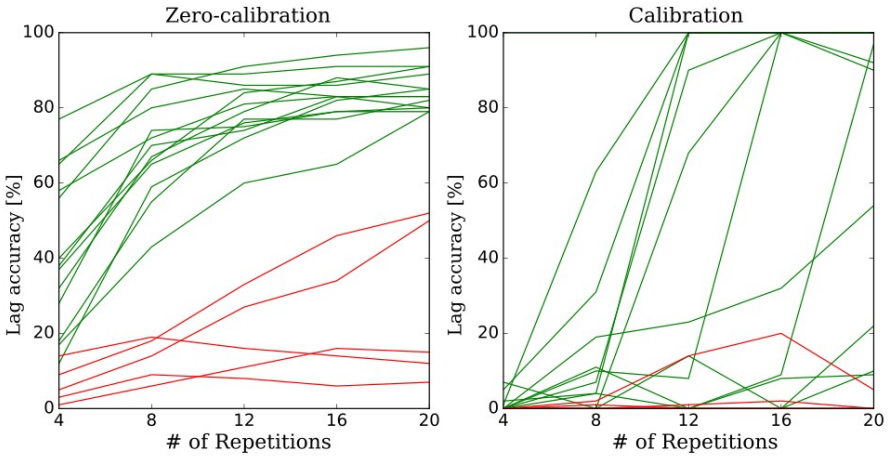
others perform poorly, even compared to the zero-calibration method.

*Significance:*

Zero-calibration BCIs are widely researched as their use is more natural. We have shown that a word prediction-based zero-calibration method in c-VEP BCIs can be efficient. Since this method relies on the correct detection of relative time lags, online experiments will be conducted to further determine the efficiency of our method.




(a)




(b) (c)

Figure 1. (a) Outline of the method: an example of spelling the word BRAIN up to the 4th letter. (b),(c) Average lag accuracy over all experiments for each subject and session, showing how many times each method was able to recover all the correct lags within a single word. The correct lag recovery is crucial to the performance of our method.

*References*

[1] Guangyu Bin, Xiaorong Gao, Yijun Wang, Shangkai Gao *VEP-based brain-computer interfaces: Time, frequency, and code modulations*, IEEE CIM, 2009.

[2] Guangyu Bin, Xiaorong Gao, Yijun Wang, Shangkai Gao *A high-speed BCI based on code modulation VEP*, Journal of Neural Engineering, 2011.

[3] Spüler M., Rosenstiel W., Bogdan M. *Online adaptation of a c-VEP brain-computer interface (BCI) based on error-related potentials and unsupervised learning, PloS one,* 2012.

# Solving the Cross-Subject Parcel Matching Problem using Optimal Transport

Guillermo Gallardo[1⋆], Nathalie T.H. Gayraud[1⋆], Rachid Deriche[1], Maureen Clerc[1], Samuel Deslauriers-Gauthier[1], and Demian Wassermann[1,2]

[1] Inria Sophia Antipolis, Université Côte d'Azur, France
[2] Inria, CEA, Université Paris-Saclay, France

**Abstract.** Matching structural parcels across different subjects is an open problem in neuroscience. Even when produced by the same technique, parcellations tend to differ in the number, shape, and spatial localization of parcels across subjects. In this work, we propose a parcel matching method based on Optimal Transport. We test its performance by matching parcels of the Desikan atlas, parcels based on a functional criteria and structural parcels. We compare our technique against three other ways to match parcels which are based on the Euclidean distance, the cosine similarity, and the Kullback-Leibler divergence. Our results show that our method achieves the highest number of correct matches.

## 1 Introduction

Brain organization displays high variability across individuals and species. Studying brain connectivity therefore faces the challenge of locating homogeneous regions while accounting for this variability. Different techniques have been proposed to parcellate the brain based on its structural connectivity. However, matching the resulting parcels across different subjects is still an open problem in neuroscience. Even when produced by the same technique, parcellations tend to differ in the number, shape, and spatial localization of parcels across subjects [8]. Current theories hold that long-range structural connectivity, namely, extrinsic connectivity, is strongly related to brain function [14]. Therefore, being able to match parcels with similar connectivity across subjects can help to understand brain function while also enabling the comparisons of cortical areas across different species [9].

Most of the current methods to match parcels across subjects are strongly linked to the technique used to create them. For example, Moreno-Dominguez et al. [11] seek correspondences between dendrograms created by means of Hierarchical Clustering. Parisot et al. [13] impose the consistence of parcels across subjects while creating the parcellation. In recent works Mars et al. propose to use the Manhattan distance, cosine similarity [10] or the Kullback–Leibler (KL) divergence [9] to compare and match connectivity fingerprints, successfully identifying common areas across humans and primates.

⋆ Both authors contributed equally in this work.

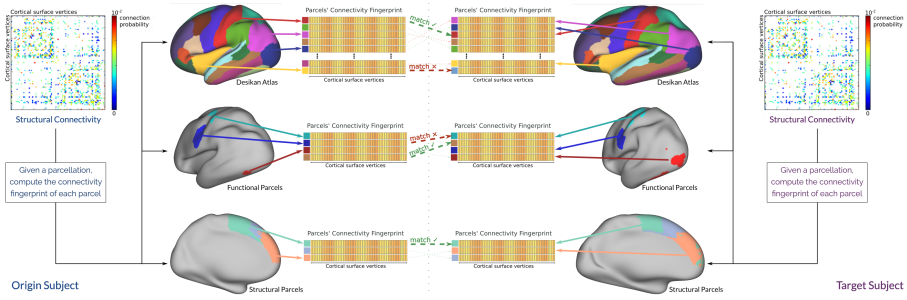



Fig. 1: From the cortico-cortical structural connectivity matrix of a subject, we can estimate the connectivity fingerprints of each parcel in three different types of parcellations. For each parcellation we compute the amount of correct matches (green lines) that each matching technique produces.

In this work, we propose to match parcels based on their extrinsic connectivity fingerprint using Optimal Transportation theory. Optimal Transport (OT) is a technique that seeks the optimal way to transport mass between probability distributions. While KL divergence computes the difference between two distributions, OT computes a matching between them. In particular, our method adopts a discrete regularized version of Optimal Transport (OT), which has been presented in Gayraud et al. [6] and Courty et al. [2] as a solution to the domain adaptation problem.

We validate our method with four different experiments. In the first experiment, we test the feasibility of our method by generating parcels with synthetic connectivity fingerprints and matching them. In the second one, we show that our technique is able to match parcels of the same atlas across subjects. We use the anatomical atlas of Desikan [4] as its parcels have high spatial coherence and consistent connectivity profiles across subjects [16]. Finally, we show the capacity of our method to match parcels generated with the same criteria but have some spatial cross-subject variability. We assess this for two different situations. In the first one, we derive the parcels from functional activations [1]. We use responses to motor and visual stimuli since they have been shown to be strongly related to structural connectivity [12, 15]. In the second one, we divide the Lateral Occipital Gyrus in 3 parcels using a structurally-based parcellation technique [5]. We use the Lateral Occipital Gyrus since it has been shown to have a consistent parcellation across subjects [17, 5]. The outline of the last three experiments can be seen in Figure 1.

In each experiment, we compare our technique against three other ways to match parcels based on the Euclidean distance; the cosine similarity; and the Kullback-Leibler divergence. Our results on real data show that our method based on OT always achieves the highest number of correct matches.

## 2 Methods

Given two subjects with their respective parcellations, we compute their parcel matching by considering one as the origin and the other one as target. More formally, let $X^a = \{x_i^a\}_{i=1}^{N_a}$, $x_i^a \in \Omega^a \subset \mathbb{R}^n$ be an origin dataset where $N_a$ denotes the number of parcels; $x_i^a$ is the extrinsic connectivity fingerprint of parcel $i$; and $n$ denotes its dimension. We wish to recover a matching between $X^a$ and a target dataset $X^b = \{x_i^b\}_{i=1}^{N_b}$, $x_i^b \in \Omega^b \subset \mathbb{R}^n$.

In this section, we start by formulating our regularized discrete OT-based method and proceed by presenting three ways of computing this matching that are based on the Euclidean distance; the cosine similarity; and the KL-divergence.

### 2.1 Discrete Regularized Optimal Transport

Optimal Transport (OT) theory boils down to finding the optimal way to transport or redistribute mass from one probability distribution to another with respect to some cost function. In this work, since the datasets $X^a$ and $X^b$ are discrete datasets, we use their empirical probability distributions and apply the discrete formulation of OT [6, 2] to solve the parcel matching problem. A simplified example of how our method proceeds is presented in Figure 2.

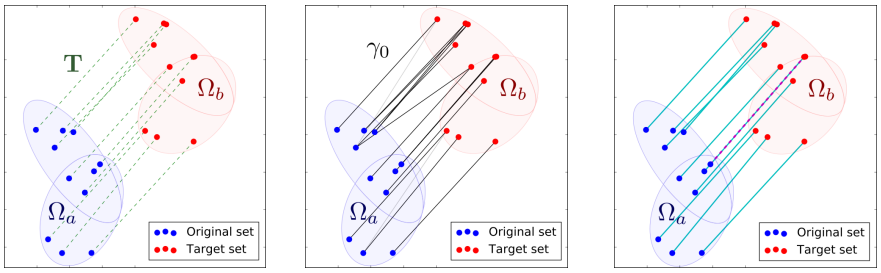
Assume that $X^a$ and $X^b$ follow probability distributions $p_a(x^a)$ and $p_b(x^b)$, respectively. We suppose that $X^a$ has undergone a transformation $\mathbf{T} : \Omega^a \to \Omega^b$, such that $p_b(\mathbf{T}(x^a)) = p_b(x^b)$. We wish to recover $\mathbf{T}$ and use it to match the parcels of $X^a$ and $X^b$. Using discrete regularized OT we compute a transport plan $\gamma_0$ between these two probability distributions. This transport plan is a doubly stochastic matrix which minimizes a certain transportation cost $C$ over the vectors of $X^a$ and $X^b$. In other words, it defines the optimal exchange of mass between the two probability distributions. We use $\gamma_0$ to compute an estimation $\hat{\mathbf{T}}$ by selecting the pairs of vectors, i.e., parcels that exchange the most mass.

Since $p_a(x^a)$ and $p_b(x^b)$ are not known, we use the corresponding empirical distributions $\mu_a = \sum_{i=1}^{N^a} p_i^a \delta_{x_i^a}$ and $\mu_b = \sum_{j=1}^{N^b} p_j^b \delta_{x_j^b}$ instead, where $p_i^a$ and $p_j^b$ are the probability masses associated to each sample. However, given that the dimension of our data depends on the number of vertices in the cortical mesh, the curse of dimensionality makes the estimation of $\mu_a$ and $\mu_b$ intrinsically difficult. We therefore simply assume a uniform probability distribution over all vectors, $p_i^a = \frac{1}{N^a}$ and $p_j^b = \frac{1}{N^b}$. We compute the transport plan $\gamma_0$ such that, if

$$\mathcal{B} = \left\{\gamma \in (\mathbb{R}^+)^{N_a \times N_b} \mid \gamma \mathbf{1}_{N_b} = \frac{1}{N^a}\mathbf{1}_{N_a}, \gamma^{\mathbf{T}}\mathbf{1}_{N_a} = \frac{1}{N^b}\mathbf{1}_{N_b}\right\} \tag{1}$$

denotes the set of all doubly stochastic matrices whose marginals are the probability measures $\mu_a$ and $\mu_b$, where $\mathbf{1}_N$ is an $N$-dimensional vector of ones, then $\gamma_0 \in \mathcal{B}$ is the output of the following minimization problem.

$$\gamma_0 = \underset{\gamma \in \mathcal{B}}{\arg\min}\ \langle \gamma, C \rangle_F + \lambda \sum_{i,j} \gamma(i,j)\ \log\ \gamma(i,j) \tag{2}$$




(a) Original & target datasets (b) Computed transport plan (c) Matching

Fig. 2: A 2-d example of using OT to compute the matching between two different datasets. On the left we show the original and target datasets. The real matchings are displayed as green dashed edges. In the middle, the edge densities represent the values of the computed coupling $\gamma_0$, which denote the amount of mass that is exchanged between vectors $x_i^a$ and $x_j^b$. On the right, we see the recovered matching. The blue edges represent the correct matchings, while the red dotted edges represent the incorrect ones.

The matrix $C$, where $C(i,j) = \|x_i^a - x_j^b\|_2^2$, represents the cost of moving probability mass from location $x_j^a$ to location $x_i^b$, in terms of their squared Euclidean distance. The rightmost term is a regularization term based on the negative entropy of $\gamma$ allows us to solve this optimization problem using the Sinkhorn-Knopp algorithm [3] which improves the computation time.

Matrix $\gamma_0$ contains information about the exchange of probability mass between the vectors of $X^a$ and $X^b$. By construction, this exchange depends on the selected cost function. The choice of the squared euclidean distance is motivated both by the fact that it renders the optimization problem convex and because it will allow the parcels to be matched according to the vicinity of their feature vectors. Hence, the origin feature vectors will distribute their corresponding probability mass to the target feature vectors that are closest to them. Consequently, we define $\hat{\mathbf{T}} : \Omega^a \to \Omega^b$ as $\hat{\mathbf{T}}(x_i^a) = x_{\hat{j}}^b$ where $\hat{j} = \arg\max_j \gamma_0(i,j)$. Therefore, $i$ will be matched to the parcel $\hat{j}$ that it sent the most mass to.

### 2.2 Matching Parcels Based on Dissimilarity Between Features

Let $d(x_i^a, x_j^b)$ be some dissimilarity measure between the elements of $X^a$ and $X^b$. Then, we say that parcel $i$ matches parcel $j$ if $\arg\min_k d(x_i^a, x_k^b) = j$. We compare three dissimilarity measures against our method. First, we use the Euclidean distance, which can be interpreted as matching the parcel $i$ to the parcel $j$ whose feature vector $x_j^b$ is the closest to $x_i^a$. Then, we use the cosine similarity, which is minimized when two feature vectors are colinear. Lastly, we use the Kullback-Leibler divergence, which measures the difference between two probability distributions in terms of their relative entropy. Note that we need to convert our vectors into probability vectors in order to evaluate $d_{KL}$.

## 3 Experiments and Results

### 3.1 Data and Preprocessing

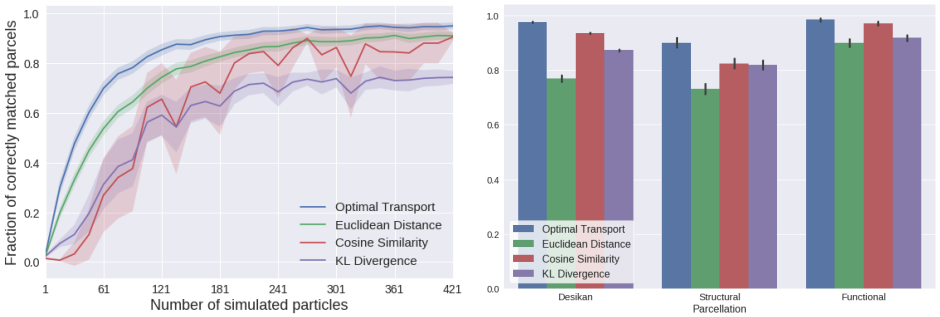
For this work we randomly selected 20 subjects from the S500 group of the Human Connectome Project (HCP), all preprocessed with the HCP minimum pipeline [7]. Fiber orientation distributions functions where computed using spherical constrained deconvolution with a spherical harmonic order of 8. Probabilistic tractography was then performed using 1000 seeds per vertex of the cortical mesh provided with the HCP data. For each subject, we computed a connectivity matrix by counting the number of streamlines that connect each pair of vertices of the cortical mesh. Each row in the matrix is a vertex connectivity vector, representing the probability that a connection exists between a surface vertex and the rest of the surface's vertices.

Given a whole brain cortical parcellation, we compute the connectivity fingerprint of each parcel by averaging the connectivity fingerprint of its vertices. Because the mesh's vertices are coregistered across subjects [7], we are able to compare the connectivity fingerprints across subjects. The criterion to compute the parcel matching between two subjects is the similarity between connectivity fingerprints. That is, we match two parcels if they are connected to the rest of the brain in a similar manner. Due to the distance bias that occurs in tractography, a parcel tends to be highly connected to the vertices that compose it. To prevent the matching to be influenced by this bias, we disconnect each parcel from its own vertices.

### 3.2 Matching Parcels

In this section we evaluate the performance of our method by comparing it to the methods presented in Section 2.2. For each experiment we compute parcel matchings between all possible pairs of connectivity matrices. To quantify the result of each technique, we compute the accuracy in terms of percentage of correctly matched parcels per pairwise matching.

**Matching parcels with synthetic fingerprints.** In this first experiment, we test the feasibility of our method by generating parcels with synthetic connectivity fingerprints and matching them. We start by generating a connectivity matrix $M$ using probabilistic Constrained Spherical Deconvolution based tractography to use as ground truth. Our ground truth matrix is a square matrix that represents the connectivity between the 64 parcels of the Desikan atlas in one subject of the HCP dataset. Each coefficient $M(i,j) = \theta_{ij}$ is the parameter of a random variable that follows a Bernoulli distribution $X_{ij}$ $B(\theta_{ij})$. This variable $X_{ij}$ represents the probability of a connection existing between the parcels $i$ and $j$. Using $M$, we generate 20 synthetic matrices in such a way that the coefficients of each synthetic connectivity matrix are random variables that follow a binomial distribution $X(i,j) \sim B(p = M(i,j), n)$. By doing this we simulate doing tractography for various values of the number $n$ of particles. Figure 3a shows the performance of each method as a function of $n$.




(a) Synthetic data (b) Real data

Fig. 3: Proportion of parcels correctly matched by each method (see section 2.2) when matching: (a) synthetic connectivity fingerprints and (b) connectivity fingerprints of a cortical parcellation, for three different parcellations (as described in section 3.2). OT always performs significantly better.

**Matching parcels of the Desikan Atlas.** For each subject, we compute the connectivity fingerprint of each parcel in their Desikan atlas as explained in Section 3.1. When matching parcels across subjects, Figure 3b shows that on average OT achieves an accuracy of 98%±2%, followed by cosine similarity (94%±3%), KL divergence (87%±4%), and finally Euclidean distance (77%±11%).

**Matching parcels created using functional criteria.** Each subject in the HCP dataset possesses z-score maps representing responses to different stimuli obtained with functional MRI (fMRI) [1]. We derive parcels for each subject from the responses to motor (hand, foot and tongue movement) and visual stimuli (faces vs shape recognition). We do so by keeping only the vertices whose z-score is in the top 35%. Figure 3b shows that OT performs best with an average of 98%±6%. The cosine similarity, KL divergence, and Euclidean distance achieve average accuracies of 97%±6%, 92%±10%, and 90%±13% respectively.

**Matching parcels created using structural criteria.** For each subject, we first mask their Lateral Occipital Gyrus using the Desikan atlas. Then, we divide it into 3 parcels using the structural based parcellation technique of Gallardo et al. [5]. Once more, we can see on Figure 3b that optimal transport has the highest average accuracy, equal to 92%±16%. It is followed by the cosine similarity, the KL divergence, and the Euclidean distance, whose average accuracies equal 85%±17%, 84%±17%, and 75%±17%

## 4 Discussion

In this work we proposed a method to match parcels across subjects based on the connectivity fingerprint of a parcel.

We tested our method with four different experiments. In the first experiment our technique correctly matched connectivity fingerprints created in a synthetic way. Specifically, each entry in a fingerprint was sampled from a Binomial distribution, whose parameter was chosen as the corresponding value of a ground truth connectivity matrix. This can be thought as a simulation of the process of tracking in tractography with different number of streamlines.

Our second experiment shows that we can correctly match parcels of the Desikan atlas across subjects with a 98% of correct matches. The parcels of the Desikan atlas are known to have high spatial coherence and consistent connectivity profiles across subjects [16]. We therefore use this experiment as a reference point to benchmark our technique. The last two experiments show that our technique can match parcels generated with a same criteria, even when they have some spatial variability across-subjects. The first experiment uses parcels created from the functional response to specific motor and visual stimuli, known to be strongly linked to functional connectivity [12, 15]. The second one, parcels created from the structural parcellation of the Lateral Occipital Gyrus, a structure documented to have a consistent structural division [17, 5].

It's important to notice that our technique achieved more than a 90% of correct matches in every experiment with real data. Given that we used 20 subjects, this represents a total of 20x19=380 cross-subject matches. In the case of the Desikan atlas, which possesses 64 parcels, this translates into a total of 24320 matches, from which 98% where correctly matched. Furthermore, when tested with a paired t-test to compare the number of correct matches, our method always performs significantly better than the other three ($p < 10^{-256}$).

## 5 Conclusion

Matching structural parcels across different subjects is an open problem in neuroscience. In this work, we proposed a novel parcel matching method based on Optimal Transport. We tested its performance with four different experiments, always obtaining the highest number of correctly matched parcels, which is an improvement over the results of the currently used techniques. Our technique could have major implications in the study of brain connectivity and its relationship with brain function, allowing for the location of parcels with similar connectivity but not high spatial coherence. Also, it could help to understand the link between different brain atlases, and improve the comparisons of cortical areas between higher primates.

## Acknowledgements

This work has received funding from the European Research Council (ERC) under the Horizon 2020 research and innovation program (ERC Advanced Grant agreement No 694665 : CoBCoM), and from the ANR NeuroRef

## References


1. Barch, D.M., Burgess, G.C., Harms, M.P., et al.: Function in the human connectome: Task-fMRI and individual differences in behavior. Neuroimage 80, 169–189 (oct 2013)
2. Courty, N., Flamary, R., Tuia, D., Rakotomamonjy, A.: Optimal transport for domain adaptation. IEEE transactions on pattern analysis and machine intelligence 39(9), 1853–1865 (2017)
3. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. In: Advances in neural information processing systems. pp. 2292–2300 (2013)
4. Desikan, R.S., Ségonne, F., Fischl, B., et al.: An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. Neuroimage 31(3), 968–980 (jul 2006)
5. Gallardo, G., Wells, W., Deriche, R., Wassermann, D.: Groupwise structural parcellation of the whole cortex: A logistic random effects model based approach. Neuroimage (January), 1–14 (feb 2017)
6. Gayraud, N.T., Rakotomamonjy, A., Clerc, M.: Optimal transport applied to transfer learning for p300 detection. In: 7th Graz Brain-Computer Interface Conference 2017 (2017)
7. Glasser, M.F., Sotiropoulos, S.N., Wilson, S., Webster, M., Polimeni, J.R., Van Essen, D.C., Jenkinson, M.: The minimal preprocessing pipelines for the Human Connectome Project. Neuroimage 80, 105–124 (oct 2013)
8. Jbabdi, S., Behrens, T.E.: Long-range connectomics. Ann. N. Y. Acad. Sci. 1305(1), 83–93 (dec 2013)
9. Mars, R.B., Sotiropoulos, S.N., Passingham, R.E.: Whole brain comparative anatomy using connectivity blueprints. bioRxiv p. 245209 (2018)
10. Mars, R.B., Verhagen, L., Gladwin, T.E., Neubert, F.X., Sallet, J., Rushworth, M.F.S.: Comparing brains by matching connectivity profiles. Neurosci. Biobehav. Rev. 60, 90–97 (2016)
11. Moreno-Dominguez, D., Anwander, A., Knösche, T.R.: A hierarchical method for whole-brain connectivity-based parcellation. Hum. Brain Mapp. 35(10), 5000–5025 (oct 2014)
12. Osher, D.E., Saxe, R.R., Koldewyn, K., Gabrieli, J.D.E., Kanwisher, N., Saygin, Z.M.: Structural Connectivity Fingerprints Predict Cortical Selectivity for Multiple Visual Categories across Cortex. Cereb. Cortex 26(4), 1668–1683 (2016)
13. Parisot, S., Arslan, S., Passerat-Palmbach, J., Wells, W.M., Rueckert, D.: Tractography-Driven Groupwise Multi-scale Parcellation of the Cortex. Inf. Process. Med. Imaging 24, 600–12 (2015)
14. Passingham, R.E., Stephan, K.E., Kötter, R.: The anatomical basis of functional localization in the cortex. Nat. Rev. Neurosci. 3(8), 606–616 (aug 2002)
15. Penfield, W., Jasper, H.: Epilepsy and the Functional Anatomy of the Human Brain. Boston (1954)
16. de Reus, M.A., van den Heuvel, M.P.: The parcellation-based connectome: Limitations and extensions. Neuroimage 80, 397–404 (2013)
17. Thiebaut de Schotten, M., Urbanski, M., Batrancourt, B., Levy, R., Dubois, B., Cerliani, L., Volle, E.: Rostro-caudal Architecture of the Frontal Lobes in Humans. Cereb. Cortex pp. 1–15 (2016)