Exploring invariances of multivariate time series via Riemannian geometry: validation on EEG data

Pedro Luiz Coelho Rodrigues

Communauté UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Spécialité : SIGNAL IMAGE PAROLE TELECOMS

Arrêté ministériel : 25 mai 2016

Présentée par

Pedro Luiz COELHO RODRIGUES

Thèse dirigée par **Christian JUTTEN**, Professeur, Communauté Université Grenoble Alpes
et codirigée par **Marco CONGEDO**, Chargé de recherche, CNRS

préparée au sein du **Laboratoire Grenoble Images Parole Signal Automatique**
dans **l'École Doctorale Electronique, Electrotechnique, Automatique, Traitement du Signal (EEATS)**

Exploration des invariances de séries temporelles multivariées via la géométrie Riemannienne : validation sur des données EEG

Exploring invariances of multivariate time series via Riemannian geometry: validation on EEG data

Thèse soutenue publiquement le **16 octobre 2019**,
devant le jury composé de :

Monsieur CHRISTIAN JUTTEN
PROFESSEUR, UNIVERSITE GRENOBLE ALPES, Directeur de thèse
Monsieur ALEXANDRE GRAMFORT
DIRECTEUR DE RECHERCHE, INRIA CENTRE SACLAY-ÎLE-DE-FRANCE, Rapporteur
Monsieur YANNICK BERTHOUMIEU
PROFESSEUR, IMS - BORDEAUX, Rapporteur
Monsieur FABIEN LOTTE
DIRECTEUR DE RECHERCHE, INRIA CENTRE BORDEAUX - SUD - OUEST, Examinateur
Monsieur STEPHANE CANU
PROFESSEUR, NORMANDIE UNIVERSITE, Examinateur
Monsieur PATRICK FLANDRIN
DIRECTEUR DE RECHERCHE, CNRS DELEGATION RHONE AUVERGNE, Président
Monsieur MARCO CONGEDO
CHARGE DE RECHERCHE, CNRS DELEGATION ALPES, Co-directeur de thèse

Exploring invariances of multivariate time series via Riemannian geometry: validation on electroencephalographic data

Pedro Luiz Coelho Rodrigues

Abstract

Multivariate time series are the standard tool for describing and analysing measurements from multiple sensors during an experiment. In this work, we discuss different aspects of such representations that are invariant to transformations occurring in practical situations. The main source of inspiration for our investigations are experiments with neural signals from electroencephalography (EEG), but the ideas that we present are amenable to other kinds of time series.

The first invariance that we consider concerns the dimensionality of the multivariate time series. Very often, signals recorded from neighbouring sensors present strong statistical dependency between them. We present techniques for disposing of the redundancy of these correlated signals and obtaining new multivariate time series that represent the same phenomenon but in a smaller dimension.

The second invariance that we treat is related to time series describing the same phenomena but recorded under different experimental conditions. For instance, signals recorded with the same experimental apparatus but on different days of the week, different test subjects, etc. In such cases, despite an underlying variability, the multivariate time series share certain commonalities that can be exploited for joint analysis. Moreover, reusing information already available from other datasets is a very appealing idea and allows for "data-efficient" machine learning methods. We present an original transfer learning procedure that transforms these time series so that their statistical distributions become aligned and can be pooled together for further statistical analysis.

Finally, we extend the previous case to when the time series are obtained from different experimental conditions and also different experimental setups. A practical example is having EEG recordings from subjects executing the same cognitive task but with the electrodes positioned differently. We present an original method that transforms these multivariate time series so that they become compatible in terms of dimensionality and also in terms of statistical distributions.

We illustrate the techniques described above on EEG epochs recorded during brain-computer interface (BCI) experiments. We show examples where the reduction of the multivariate time series does not affect the performance of statistical classifiers used to distinguish their classes, as well as instances where our transfer learning

and dimension-matching proposals provide remarkable results on classification in cross-session and cross-subject settings.

For exploring the invariances presented above, we rely on a framework that parametrizes the statistics of the multivariate time series via Hermitian positive definite (HPD) matrices. We manipulate these matrices by considering them in a Riemannian manifold in which an adequate metric is chosen. We use concepts from Riemannian geometry to define notions such as geodesic distance, center of mass, and statistical classifiers for time series. This approach is rooted on fundamental results of differential geometry for Hermitian positive definite matrices and has links with other well established areas in applied mathematics, such as information geometry and signal processing.

Résumé

L'utilisation de séries temporelles multi-variées est une procédure standard pour décrire et analyser des mesures enregistrées par plusieurs capteurs au cours d'une expérience. Dans ce travail, nous discutons certains aspects de ces représentations temporelles, invariants aux transformations qui peuvent se produire en situations pratiques. Nos recherches s'inspirent en grande partie d'expériences neurophysiologiques reposant sur l'enregistrement de l'activité cérébrale au moyen de l'électroencéphalographie (EEG), mais les idées que nous présentons ne sont pas restreintes à ce cas particulier et peuvent s'étendre à d'autres types de séries temporelles.

La première invariance sur laquelle nous portons notre attention est celle de la dimensionalité des séries temporelles multi-variées. Bien souvent, les signaux enregistrés par des capteurs voisins présentent une forte dépendance statistique entre eux. Nous introduisons donc l'utilisation de techniques permettant d'éliminer la redondance des signaux corrélés et d'obtenir de nouvelles représentations du même phénomène en dimension réduite.

La deuxième invariance que nous traitons est liée à des séries temporelles qui décrivent le même phénomène mais sont enregistrées dans des conditions expérimentales différentes. Par exemple, des signaux enregistrés avec le même appareil expérimental, mais à différents jours de la semaine ou sur différents sujets, etc. Dans de tels cas, malgré une variabilité sous-jacente, les séries temporelles multi-variées partagent certains points communs qui peuvent être exploités par une analyse conjointe. En outre, la réutilisation des informations déjà disponibles à partir d'autres jeux de données est une idée très séduisante et permet l'utilisation de méthodes d'apprentissage automatiques dites «data-efficient». Nous présentons une procédure originale d'apprentissage par transfert qui transforme les séries temporelles de telle sorte que leurs distributions statistiques soient alignées et puissent être regroupées pour une analyse statistique plus poussée.

Enfin, nous étendons le cas précédent au contexte où les séries temporelles sont obtenues à partir de différentes conditions expérimentales et de différentes configurations d'enregistrement de données. Nous présentons une méthode originale qui transforme ces séries temporelles multi-variées afin qu'elles deviennent compatibles en termes de dimensionalité et de distributions statistiques.

Nous illustrons les techniques citées ci-dessus en les appliquant à des signaux EEG enregistrés dans le cadre d'expériences d'interface cerveau-ordinateur (BCI). Nous montrons sur plusieurs exemples, avec des simulations et des données réelles, que la réduction de dimension – judicieusement choisie – de la série temporelle multi-variée

n'affecte pas les performances de classifieurs statistiques utilisés pour déterminer la classe des signaux, et que notre méthode de transfert d'apprentissage et de compatibilité de dimensionalité apporte des améliorations remarquables en matière de classification inter-sessions et inter-sujets.

Pour explorer les invariances présentées ci-dessus, nous nous appuyons sur l'utilisation de matrices Hermitiennes définies positives (HPD) afin de décrire les statistiques des séries temporelles multi-variées. Nous manipulons ces matrices en considérant qu'elles reposent dans une variété Riemannienne pour laquelle une métrique adéquate est choisie. Nous utilisons des concepts issus de la géométrie Riemannienne pour définir des notions telles que la distance géodésique, le centre de masse ou encore les classifieurs statistiques de séries temporelles. Cette approche repose sur les résultats fondamentaux de la géométrie différentielle pour les matrices Hermitiennes définies positives et est liée à d'autres domaines bien établis en mathématiques appliquées, tels que la géométrie de l'information et le traitement du signal.

Acknowledgement

Dear reader, the following $\simeq$170 pages describe the works that I've developed during my three years of doctoral studies at the GIPSA-lab, in Grenoble. The text has been written in English for the sake of ensuring a wider audience, but, in the next few lines, I will indulge myself to write in French, English, and Portuguese to thank the different people that accompanied me during this long and winding road.

Tout d'abord, j'aimerais remercier mes encadrants de thèse: Christian Jutten et Marco Congedo. Je vous remercie d'avoir cru en moi depuis le tout début. Vous m'avez donné à la fois la liberté totale d'explorer des thématiques qui m'inspiraient et l'orientation pour approfondir les problèmes qui vous paraissaient intéressants et pertinents. Je vous remercie d'avoir accepté d'encadrer un étudiant brésilien sorti un peu de nulle part et avoir fait l'effort de trouver un financement qui m'a permis de venir en France pour travailler avec vous. Vous avez été les encadrants de thèse parfaits, toujours disponibles et extrêmement généreux avec le temps que vous m'avez accordé, vos connaissances et votre passion pour la recherche. Sachez que, si un jour j'avais le plaisir d'encadrer d'autres étudiants, je m'inspirerais de vous et de toutes les expériences que j'ai vécues pendant ces années de thèse.

Ensuite, j'aimerais remercier les membres de mon jury de thèse. Merci à Alexandre Gramfort et Yannick Berthoumieu d'avoir accepté le rôle de rapporteurs de mes travaux de recherche, et à Stéphane Canu et Fabien Lotte pour avoir participé, en tant qu'examinateurs, à mon jury. Merci à Patrick Flandrin, membre de l'Académie de Sciences, et actuellement vice-président, d'avoir présidé le jury. Ce fut un très grand honneur et un plaisir d'avoir soutenu ma thèse devant un jury aussi prestigieux.

Je tiens aussi à remercier tous ceux qui m'ont suivi de près ou de loin pendant ces trois années de thèse à Grenoble. J'ai eu le plaisir de rencontrer des nombreuses personnes dont certains sont devenus des grands amis. J'ai fait de mon mieux pour mentionner tout le monde (ou presque) dans les prochains paragraphes. Voyons ce que ça donne.

Je commence par ceux que j'ai rencontrés au GIPSA-lab. Les amitiés qui se créent pendant une thèse sont assez particulières. La frontière entre la vie perso et la vie pro devient souvent floue et on s'attache très vite à nos collègues de travail. Heureusement pour moi, j'ai eu le plaisir d'être entouré par des gens incroyables

avec qui j'ai passé des très bon moments dans le labo et en dehors aussi. Je pourrais écrire des phrases, des paragraphes, ou même des poèmes pour remercier chacun de vous. Cependant, faute d'espace (et de temps), je vais faire simple. Merci, donc, à Miguel, Taia, Lucas, Victor, Marielle, Paolo, Raph, Pierre Maho, Kévin, Quentin, Alex Marquet, Pierre Narvor, Aziliz, Emmanuelle, Jeanne, Louis Deschamps, Louis Korczowski, Florent, Grégoire, Robin, Cosme, Violette, Imane, Geoff, Edurne, María, Marc, Marion Revolle, Fabrice, Gaël, Ludovic, Julien et Fanny. I would also like to thank those colleagues who are less comfortable with Molière's language. Thank you, Tien, Dawood, Maria, Karina, Luisa, and Omar.

Eu também gostaria de agrader a meus amigos brasileiros erradicados na França que compartilharam comigo a saudade de casa e o gosto por essa vida no país da baguette e das randonnées. Obrigado, Juice and Yumi, João Gabriel, Rafael, Olcyr e Victor Hugo et Catarina.

Je garde un paragraphe tout spécial pour remercier la famille Vilaca (+Carco) pour m'avoir accueilli déjà dans la dernière ligne droite de la thèse. Merci, Cam, pour ton aide dans la préparation de ce pot de thèse qui va rester dans les annales (merci surtout pour les crêpes !). Merci, Myriam, pour l'accueil à Dolomieu et les agréables journées de woofing dans ton jardin. Et merci, Marion, pour toute ton aide, ton attention et ta complicité. Ce n'était pas simple de te lancer dans cette aventure d'accompagner un doctorant en fin de thèse... merci de l'avoir fait quand même !

Por fim, mas não menos importante, obrigado à minha família, por todo o amor e todo o encorajamento desde o primeiro minuto. Sem vocês, eu não seria nada.

Contents

1 Introduction **1**

2 Theoretical background **9**
2.1 Introduction . . . 11
2.2 Multivariate time series analysis . . . 11
2.3 Riemannian geometry of the HPD manifold . . . 18
2.4 Riemannian geometry for EEG signals . . . 28
2.5 Numerical illustrations . . . 36
2.6 Conclusion . . . 44

3 Dimensionality reduction **45**
3.1 Introduction . . . 47
3.2 Linear methods . . . 49
3.3 Non-linear methods . . . 60
3.4 Conclusion . . . 72

4 Transfer learning **75**
4.1 Introduction . . . 77
4.2 Literature review . . . 79
4.3 Riemannian Procrustes analysis . . . 85
4.4 Numerical illustrations: simulated data . . . 98
4.5 Numerical illustrations: real data . . . 101
4.6 Conclusion . . . 114

5 Transcending dimensions **117**
5.1 Introduction . . . 119
5.2 Literature review . . . 121
5.3 Dimensionality transcending . . . 124
5.4 Application to BCI datasets . . . 131
5.5 Numerical illustrations . . . 134
5.6 Conclusion . . . 144

6 Conclusion **149**

Bibliography **153**

1 Introduction

We are surrounded by sources of activity that fluctuate in a more or less irregular manner: temperatures changing in the course of a year, stock market prices oscillating during a day, brain activity varying in the scale of milliseconds. The goal of a scientist is to quantify, understand, model, and predict the time evolution of such phenomena. For this purpose, we use the concept of *multivariate time series*, which represents measurements obtained from a set of sensors as a collection of vectors indexed by time. For example, Figure 1.1 illustrates electroencephalographic (EEG) signals recorded on three different electrodes placed over a person's scalp. These signals can be conveniently studied as a three-dimensional time series, where each dimension represents the signal recorded on each electrode.

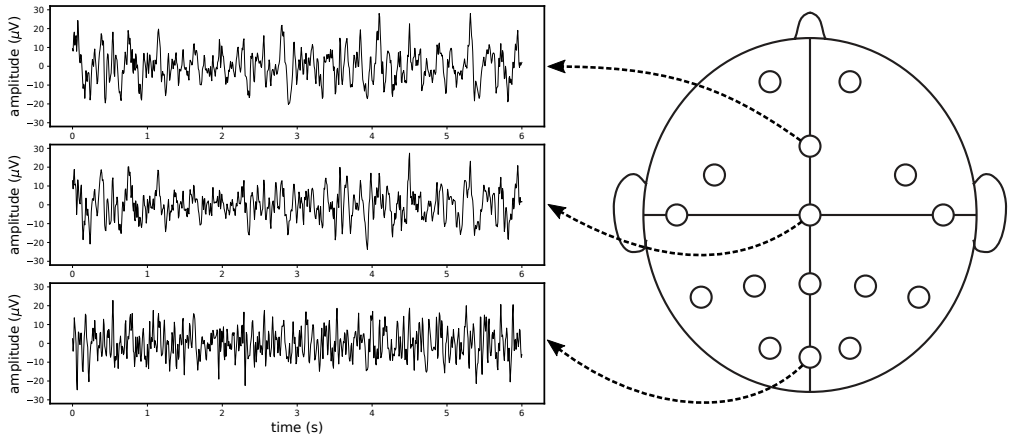

Fig. 1.1: EEG signals recorded on electrodes `Fz`, `Cz`, and `Oz`, on a subject's scalp during a resting-state experiment. Data is from the `ALPHA.EEG.2017-GIPSA` database [Cat+18].

A common way for analysing multivariate time series is to estimate a set of parameters that describes its statistical behavior, such as its mean vector, its auto-covariance matrices, or its cross-spectral density matrices [Pri83]. This assumes that the time series are stationary and that their statistical behavior can be exhaustively described by their second-order moments, i.e., that they are multivariate Gaussian processes. Two multivariate time series may then be compared by defining a distance between the sets of parameters describing their statistics. A principled way for doing so is to study the intrinsic geometry of the space where the parameters are defined and use the geodesic distance between them as a measure of similarity. Such approach is based on concepts borrowed from Riemannian geometry (RG) and allows us to manipulate multivariate time series as points in a metric space. This abstraction

is inspired by what is done in information geometry, where statistical distributions are seen as points in a statistical manifold and then compared using the geodesic distance induced by a metric defined in it [Ama16]. A convenient outcome of this approach is that it allows the development of new algorithms inspired by intuitive geometric arguments, as well as a new understanding of classical algorithms that were firstly developed in a purely analytical form and that can be reinterpreted under the RG framework. Figure 1.2 gives a visual intuition of the RG framework applied to multivariate time series.

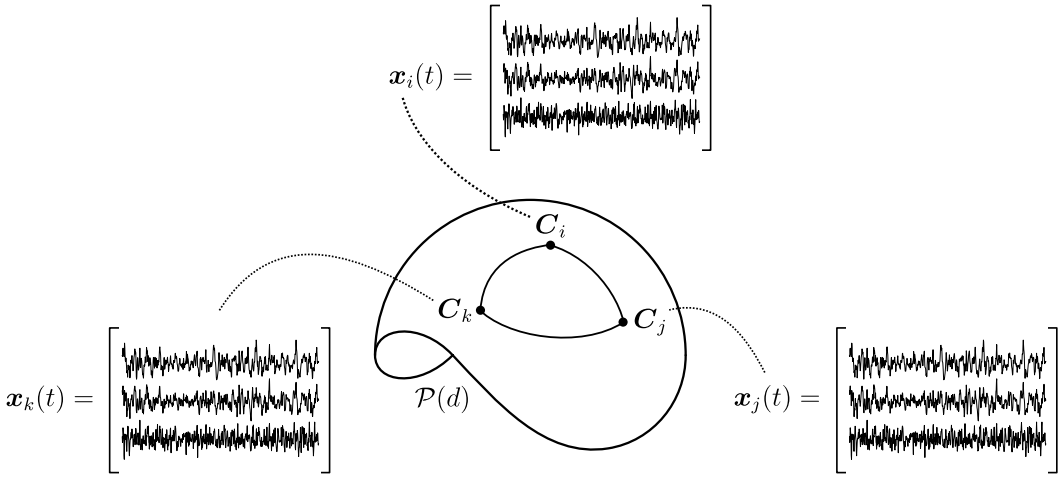

Fig. 1.2: Visual representation of the RG framework. The statistics of each d-dimensional multivariate time series ($d = 3$ in the figure) is described by its spatial covariance matrix, with $\boldsymbol{x}_i(t)$, $\boldsymbol{x}_j(t)$, and $\boldsymbol{x}_k(t)$, associated to $\boldsymbol{C}_i$, $\boldsymbol{C}_j$, and $\boldsymbol{C}_k$, respectively. These matrices are symmetric positive definite (SPD) and have dimensions $d \times d$. They live in a manifold, the SPD manifold, denoted by $\mathcal{P}(d)$, and we use tools from Riemannian geometry to manipulate them. $\mathcal{P}(d)$ has non-positive curvature, which makes its geometry different from that of a flat Euclidean space. We have, for instance, that the sum of the angles of a triangle in $\mathcal{P}(d)$ is not 180°. Also, the distance between two points in $\mathcal{P}(d)$ is given by the length of the geodesic path connecting them, which is not necessarily a straight path.

Geometry-aware algorithms have gained increasing attention in the last few years. It has been shown in a number of occasions that studying and understanding the intrinsic geometry of a set of features or data points gives considerable insight, allowing for the development of new and more efficient algorithms. Some examples are the recent surge of deep learning algorithms crafted for handling data defined in a manifold [Bro+17] or the reinterpretation of classical methods for text classification using concepts of Riemannian geometry [Leb05]. In the context of multivariate time series, the Riemannian geometric framework has lead to major improvements in the field of brain-computer interfaces (BCI) based on EEG signals, where classification methods were traditionally known to have rather weak generalization properties [Lot+18]. In RG methods, the statistics of the EEG signals are parametrized via their spatial covariance matrices, which are symmetric positive definite (SPD) matrices. Such

matrices are defined in a Riemannian manifold with a well known intrinsic geometry [Bha09] and that can be used to define methods for manipulating the multivariate time series via their statistical descriptors. A particularly interesting feature of the SPD manifold is that we may define a geodesic distance which is invariant to affine transformations. Consequently, the distance between two multivariate time series parametrized by SPD matrices is not affected by the action of a linear transformation applied to them. This is a very attractive property, since linear transformations can be used to model different practical situations, such as the effect of slightly moving the positions of electrodes on a subject's scalp or the effects caused by the mixture of different sources of activity in a person's brain. The RG approach for classifying EEG signals has, therefore, lead to new classification algorithms that have demonstrated excellent results in practice and have become one of the state-of-the-art methods in the BCI research community [Con+17; Yge+17; Lot+18].

Objective of the thesis

This thesis uses Riemannian geometry concepts to investigate *invariances* in multivariate time series. An invariance is "a property that remains unchanged regardless of changes in the conditions of measurements". This is a very powerful property of a system, which reflects a notion of stability that is intrinsic to the phenomenom under study and that allows for a profound interpretation of its behavior. Invariances are at the core of many scientific fields, such as in classical mechanics, where the laws of motion are the same in all inertial frames (also known as Galilean invariance), and electromagnetism, where invariances and symmetries are commonly used to determine expressions describing electric and magnetic fields. In image processing, the use of invariant features for classification is a very active topic of research. For instance, the scale-invariant feature transform (SIFT) [Low04] leads to an algorithm that detects features which are invariant to changes in scale, illumination, and noise. Such features are robust descriptors of images, and classification algorithms based on them have demonstrated good generalization properties in practice. Invariances in images are also at the core of convolutional neural networks, which are built to exploit the fact that the output for a classifier to a given image should always be the same regardless small deformations that it may undertake, such as translation, rotation, and stretching.

In the context of multivariate time series, invariances may be related to different aspects of the phenomena they represent. For instance, the statistical distribution of samples gathered from different experimental sessions are usually different, hindering their joint analysis with classical statistical methods. However, if the experiments portray the same phenomena, it is reasonable to assume that the samples of each session share invariant features. A concrete example is in EEG-based

BCI, where the data from two subjects carrying out the same cognitive tasks, i.e. the same BCI paradigm, may have very different statistical distributions, even if latent information is clearly shared. Similarly, multivariate time series may be recorded using different sensor setups – different positions of electrodes, different number of sensors, etc. In this case, the dimensionality mismatch between time series makes them, in principle, incompatible for joint statistical analysis. However, if they portray the same phenomena, we expect that they share invariances that could be exploited. In the context of EEG-based BCI, this is related to datasets that are recorded in different laboratories using different experimental setups, but under the same BCI paradigm. In Figure 1.3, we use a non-linear dimensionality reduction technique called *diffusion maps* [CL06] to illustrate the differences between the descriptors of multivariate time series associated to two subjects performing a BCI experiment. We observe a clear difference in the distributions of data points, although the subjects were asked to perform the same set of cognitive tasks. This mismatch explains why a classifier trained on the dataset from one subject has poor performance when applied to a dataset from another subject.

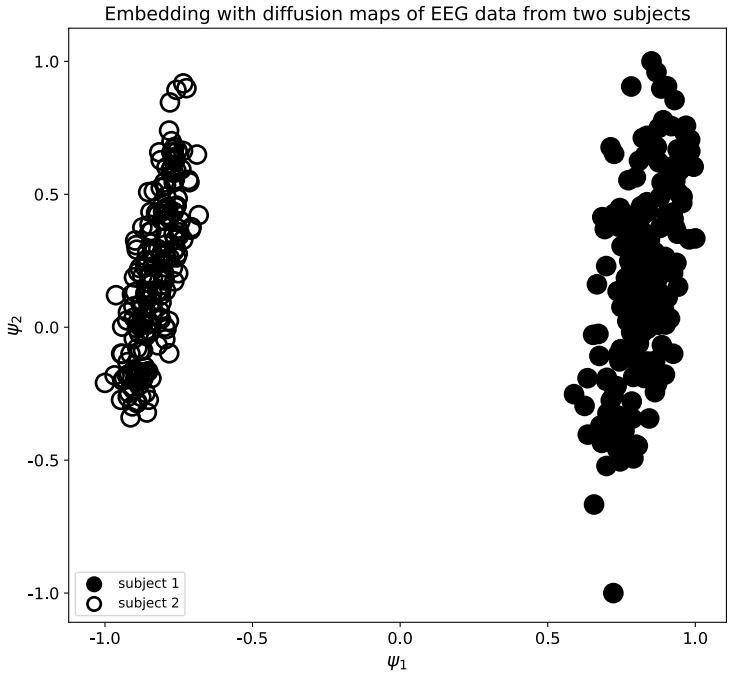

Fig. 1.3: Two-dimensional representation of the embedding obtained via the diffusion maps algorithm applied to the recordings of two subjects in the `Cho2017` database [Cho+17]; the axis ψ_1 and ψ_2 are eigenvectors of the Laplacian matrix estimated from the data points with the diffusion maps algorithm [CL06]. Each point corresponds to the EEG signal of an experimental trial and the distances between the data points were calculated using the geodesic distance of the SPD manifold.

Questions related to invariances in multivariate time series are very general and applicable to several contexts. In this thesis, we give particular attention to examples

with EEG data. We consider datasets obtained from different experimental paradigms, such as brain-computer interfaces, sleep recordings, and resting-state experiments. On the one hand, working with this kind of data is rather challenging because it often has low signal-to-noise ratio and presents considerable variability between recording sessions. On the other hand, EEG signals are very rich and carry physiological information in its spectral content and waveforms, which can be used to cope with the intrinsic limitations related to how it is recorded. Our results demonstrate that the exploration of invariances in EEG time series is very fruitful and allows for the design of new and better methods for their analysis and classification.

Organization of the manuscript

The text that follows is composed of five chapters.

In Chapter 2, we present the theoretical foundations on which all contributions of this thesis rely. We begin by presenting the traditional approach for multivariate time series analysis and show how it can be studied under a Riemannian geometric framework. Then, we give a brief overview of the geometric properties of the manifold where the statistical parameters of the multivariate time series are defined. We conclude by showing how to apply the RG framework to EEG data and illustrating its use on a few practical examples.

In Chapter 3, we consider problems related to the dimensionality reduction (DR) of multivariate time series. For this, we investigate how DR techniques can be used to exploit redundancies in multivariate time series and obtain more compact representations for them. We first consider linear techniques and use an extension of the classical principal component analysis to a context where the data points live in a space which is non-Euclidean. We also consider non-linear DR techniques and focus on the method of *diffusion maps* (DM). We apply DM to datasets containing multivariate time series and show how this procedure can be used for unsupervised analysis of such kind of data. The scope of our contributions in this chapter is rather limited as compared to the following chapters, but it concerns an important practical problem that deserves to be discussed. Furthermore, it illustrates an invariant property of multivariate time series related to how two different representations of the same phenomenon can convey the same information.

In Chapter 4, we consider the problem of *transfer learning* applied to multivariate time series. Transfer learning is a very relevant topic in machine learning and concerns the ability of a system to extract knowledge obtained from different sources of information. This goes in line with discussions regarding the 'data-efficiency' of classification algorithms, which ponders on how information from different datasets can be reused to avoid the need for generating new samples and, without

loosing classification performance, avoid the need of redoing energy consuming experiments, storing new data samples, etc. We present an original contribution that uses the RG framework to adapt the statistics of mismatched datasets and makes their joint analysis possible. Our method is an extension of the classical Procrustes analysis [Ken89], which applies rigid transformations to data points (i.e., translation, stretching and rotation) from two datasets in order to match their statistical distributions. These transformations are carried out on points defined in a Riemannian manifold, therefore, we call our method the *Riemannian Procrustes analysis* (RPA). The works on this chapter have generated the publication:

> P. L. C. Rodrigues, C. Jutten, and M. Congedo, "Riemannian procrustes analysis: transfer learning for brain-computer interfaces", IEEE Transactions on Biomedical Engineering, pp. 1–1, 2018.

In the numerical illustrations of Chapter 4, we apply RPA to data from EEG-based BCI experiments and show that it yields very good results in cross-subject classification, i.e., when the data from one subject is classified using a classifier trained with the data from another subject. These results pave the way to new BCI systems able to reduce (or even bypass) the calibration phase.

In Chapter 5, we extend the context of the preceding chapter and consider the case of datasets containing multivariate time series of different dimensionalities and/or registered with different sensor positions. This kind of situation represents the common problem of trying to match datasets coming from different experimental setups but representing the same phenomena. We present an original contribution based on concepts from RG to match datasets obtained under this context. Our proposal uses a two-step procedure that transforms the parameters describing the statistics of multivariate time series so that they become matched in terms of dimensionality and statistics. In the dimensionality matching step, we use isometric transformations to map the features of each dataset into a common space without changing their internal geometric structures. Then, the statistical matching of the dimensionality-matched data points is done using RPA. We have named this procedure *dimensionality transcending* (DT) and we have submitted a paper describing this proposal:

> P. L. C. Rodrigues, C. Jutten, and M. Congedo, "Dimensionality transcending: a method for working with datasets defined in different SPD manifolds", submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence.

In the numerical illustrations of Chapter 5, we apply DT to BCI datasets recorded using different experimental setups (for instance, different number and placement of electrodes) but under the same paradigm. Our results demonstrate that it is

indeed possible to extract common latent information from mismatched data and use it to perform cross-subject classification even when the data come from subjects associated to different databases. Our results bring the idea of 'data-efficiency' in the BCI field to a new level, making it possible to reuse information from previously incompatible sources of data.

In Chapter 6, we present our concluding remarks concerning the investigations carried out during this thesis and discuss on future perspectives for what we have developed. We split the perspectives into 'short-term' and 'long-term' goals and give an overview of what are the most interesting paths of research that this thesis opens.

In order to foster reproducible research, `Python` code for all methods discussed in this thesis are available online on the public repository:

`https://github.com/plcrodrigues/PhD-Code`

Moreover, all numerical illustrations have been carried out on publicly available datasets, some of which were developed at the GIPSA-lab.

Theoretical background

2

Contents

2.1 Introduction **11**
2.2 Multivariate time series analysis **11**
2.2.1 Basic definitions and notation 12
2.2.2 Statistical assumptions 12
2.2.3 Distance between multivariate time series 15
2.3 Riemannian geometry of the HPD manifold **18**
2.3.1 Basic definitions and notation 18
2.3.2 Statistics in the HPD manifold 23
2.3.3 Classification in the HPD manifold 25
2.4 Riemannian geometry for EEG signals **28**
2.4.1 Basic concepts about EEG 28
2.4.2 The Riemannian geometric framework 32
2.4.3 An application: BCI classification 34
2.5 Numerical illustrations **36**
2.5.1 Example 1: BCI classification 36
2.5.2 Example 2: Sleep-stage classification 40
2.6 Conclusion **44**

List of acronyms and notations of the chapter

EEG	electroencephalography
BCI	brain-computer interface
HPD	Hermitian positive definite
SPD	symmetric positive definite
RG	Riemannian geometry
AIRM	affine-invariant Riemannian metric
MDM	minimum distance to mean classifier
ERP	event-related potential
MI	motor imagery
ROC	receiver operating characteristic
AUC	area under the ROC curve
DTFT	discrete-time Fourier transform
$\mathbb{Z}$	set of integer numbers
$\mathbb{R}^d$	set of d-dimensional real vectors
$\boldsymbol{x}$	multivariate time series
$\boldsymbol{C}$	spatial covariance matrix
$\boldsymbol{S}$	cross-spectral density matrix
δ_E	Frobenius distance between two matrices
δ_R	AIRM-induced distance between two HPD matrices
$\mathcal{H}(d)$	set of d-dimensional Hermitian matrices
$\mathcal{P}(d)$	manifold of d-dimensional HPD matrices
$T_{\boldsymbol{C}}\mathcal{P}(d)$	tangent space to $\mathcal{P}(d)$ at point $\boldsymbol{C}$
$\boldsymbol{M}^{\mathcal{X}}$	geometric mean of the HPD matrices in a set $\mathcal{X}$

2.1 Introduction

This chapter presents the theoretical foundations on which all contributions of this thesis rely. Section 2.2 introduces statistical tools for the analysis of multivariate time series and discuss two fundamental assumptions that are typically done regarding their statistics: stationarity and Gaussianity. These assumptions allow the description of the full statistical behavior of a real multivariate time series via a set of Hermitian positive definite (HPD) matrices. One may then compare two time series by comparing the HPD matrices used to parametrize them.

The set of HPD matrices is known to have a particular intrinsic geometry and in Section 2.3 we give an overview of its properties. Most importantly, we present a distance between HPD matrices that is invariant to affine-invariant transformations (e.g. the action of a matrix), a very useful property when parametrizing multivariate time series. We also show how to model the statistics of a dataset containing HPD data points and how to use statistical classifiers for discriminating between different classes of HPD matrices. The combination of all these concepts is what we call the Riemannian geometric (RG) framework for multivariate time series.

Section 2.4 describes how to apply the RG framework to recordings of electroencephalographic (EEG) signals. We introduce basic concepts related to the electrical activity in the brain (how it is generated, measured, and processed) as well as some important markers that are often used to classify EEG signals. Then, we show how to parametrize EEG signals via HPD matrices and give an overview of recent brain-computer interface (BCI) applications that use the Riemannian geometric framework.

Section 5.5 closes the chapter with numerical illustrations on EEG data of two kinds: BCI and sleep recordings. All examples with BCI classification were carried out using the `MOABB` framework [JB18], which is a Python library based mostly on three other libraries: `scikit-learn` [Ped+11], `MNE-python` [Gra13] and `pyRiemann`[1]. The scripts generating some of the figures in this chapter are available in the `GitHub` repository for this thesis:

`https://github.com/plcrodrigues/PhD-Code`

2.2 Multivariate time series analysis

In this section, we define what is a multivariate time series and present statistical tools for analysing it. We discuss some common assumptions regarding their statistics and define a notion of distance between two multivariate time series.

[1] `http://pyriemann.readthedocs.io/`

2.2.1 Basic definitions and notation

We define a multivariate time series as a collection of d-dimensional vectors

$$\boldsymbol{x}(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_d(t) \end{bmatrix} \tag{2.1}$$

indexed by $t \in \mathbb{Z}$, with $t_i - t_{i-1} = T_s$ its sampling period. Each dimension in $\boldsymbol{x}(t)$ represents a different quantity that depends of the context where the time series is defined. For instance, in stock market prediction, each dimension describes the time evolution of a certain stock [Lut07], whereas for experiments with audio, $x_i(t)$ is associated to what is recorded at microphone i [OS94]. In EEG recordings, each time series in $\boldsymbol{x}(t)$ is related to the neural activity registered by one electrode placed on a subject's scalp [SC07].

The standard approach for studying multivariate time series is to consider each sample $\boldsymbol{x}(t)$ as a random vector in $\mathbb{R}^d$ generated by some statistical law whose probability density function is $\pi_{\boldsymbol{x}(t)}$. One can then define basic statistical quantities, such as the mean value of the time series at each time instant t,

$$\boldsymbol{\mu}(t) = \mathbb{E}\left[\boldsymbol{x}(t)\right] = \int_{\mathbb{R}^d} \boldsymbol{y}\, \pi_{\boldsymbol{x}(t)}(\boldsymbol{y}) \mathrm{d}\boldsymbol{y}\ , \tag{2.2}$$

and its autocovariance between two time instants t and s,

$$\boldsymbol{R}(t,s) = \mathbb{E}\Big[\big(\boldsymbol{x}(t) - \boldsymbol{\mu}(t)\big)\big(\boldsymbol{x}(s) - \boldsymbol{\mu}(s)\big)^H\Big] \tag{2.3}$$

$$= \int_{\mathbb{R}^d \times \mathbb{R}^d} \big(\boldsymbol{y} - \boldsymbol{\mu}(t)\big)\big(\boldsymbol{z} - \boldsymbol{\mu}(s)\big)^H\, \pi_{[\boldsymbol{x}(t),\boldsymbol{x}(s)]}(\boldsymbol{y},\boldsymbol{z}) \mathrm{d}\boldsymbol{y}\mathrm{d}\boldsymbol{z}\ , \tag{2.4}$$

where $\pi_{[\boldsymbol{x}(t),\boldsymbol{x}(s)]}$ is the joint probability density function for $\boldsymbol{x}(t)$ and $\boldsymbol{x}(s)$, and $\boldsymbol{x}^H$ denotes the conjugate transpose of $\boldsymbol{x}$. Other statistical quantities may also be defined, such as higher-order moments (kurtosis, skewness, etc.) [NM93] or the entropy of the time series [BV00], but we will not consider them in this thesis.

2.2.2 Statistical assumptions

It is common to make assumptions regarding the statistics of the samples of a multivariate time series $\boldsymbol{x}(t)$. When these assumptions are verified, one can obtain better estimators for describing the statistical law of the samples (less bias and smaller variance), as well as clearer interpretations about the underlying stochastic process that generated them [Lut07].

Stationarity. One of such assumptions is regarding how the statistics of $\boldsymbol{x}(t)$ evolves in time. A common hypothesis is that of *wide-sense stationarity* (WSS), which assumes that the mean of the multivariate time series is constant for all time samples,

$$\boldsymbol{\mu}(t) = \boldsymbol{\mu} \, , \tag{2.5}$$

and that the autocovariance matrix for two time instants t and s depends only or their lag difference $\tau = t - s$,

$$\boldsymbol{R}(t, s) = \boldsymbol{R}(t - s, 0) = \boldsymbol{R}(\tau) \, . \tag{2.6}$$

Under the WSS hypothesis, one can also define the notion of cross-power spectral density of a multivariate time series, which is the discrete-time Fourier transform (DTFT) of the sequence of auto-covariance matrices [Pri83]

$$\boldsymbol{S}(f) = \sum_{k=-\infty}^{+\infty} \boldsymbol{R}(k) e^{-j2\pi fk} \, , \tag{2.7}$$

where $f \in [0, 1]$ is a normalized frequency and j is the imaginary unit[2]. The quantity $\boldsymbol{S}(f)$ is a Hermitian positive definite matrix whose diagonal values describe how the power (or variance) of each time series in $\boldsymbol{x}(t)$ is distributed along the frequency domain; the out-of-diagonal values portray the statistical correlation between the time series in each pair of dimensions in the frequency domain. For simplicity, in the rest of this thesis, we will use interchangeably the terms 'stationarity' and 'wide-sense stationarity', although 'stationarity' is often defined as a stronger property than 'wide-sense stationarity'; in fact, wide-sense stationarity means stationarity up to the second order, and hence is equivalent to stationarity for Gaussian time series. See [Pri83] and [PW93] for more details.

Stationarity ensures interesting statistical properties on the time series, but it might not always be adequate to assume it is true. In fact, there are many applications where the goal is to identify changes in the statistics of the samples, such as detecting changes in the behavior of financial time series [Tuc95; Lun+03], changes in neural connectivity [Ast+08; RB15], changes in seismic activity [JK93; Mal+18], or, more broadly, changes in the statistics of a dynamical system generating samples of a time series [Bas88]. In this context, assuming WSS for the whole time series would be contradictory. Nevertheless, a common approach is to assume that the changes in the statistics are relatively smooth, so samples from small time intervals around a given time sample t have approximately the same statistics. In this approach, one uses a small sliding window containing a certain number of samples and considers that their statistical behavior can be described by the same mean vector and autocovariance

[2]Note that, in practice, we have only access to a finite number of samples of a time series. Consequently, the sum in Eq. (2.7) has always a finite number of terms and, therefore, is always convergent.

matrices. Then, the evolution of $\boldsymbol{x}(t)$'s statistics is described by how its mean and auto-covariance matrices evolve from one window to the next. Note that the choice of 'how small' the sliding window should be, depends on the sampling frequency and the time-scale that one wants to consider in a given application; choosing a window that is too small yields poor statistical estimators, whereas larger windows may blur the dynamics that one is trying to reveal. There have been many works in the literature for optimal segmentation of time series into windows. For instance, [Hal+17] proposes a model-based clustering method for detecting samples whose statistics may be described by the same covariance matrix, [Das+98] defines a motif discovery algorithm based on pairwise distances between short windows, and [BB83] segments statistically homogeneous strands of data based on distances between autoregressive models estimated in each window. See [Lov+14] and [Keo+93] for surveys on this topic.

Gaussianity. Another usual assumption concerns how the statistics of $\boldsymbol{x}(t)$ should be characterized. Our approach in this thesis, and in most of the literature of time series analysis [Lut07] and signal processing [Mar87], is to assume that $\pi_{\boldsymbol{x}(t)}$ can be approximated by a multivariate Gaussian distribution. Under this hypothesis, the mean vector and sequence of autocovariance matrices (or, equivalently, the cross-spectral density matrices) describe the full statistical behavior of $\boldsymbol{x}(t)$ [Pri83]. This can be used for defining a notion of distance between two time series, as discussed in Section 2.2.3. Equivalently, we say that the statistics of $\boldsymbol{x}(t)$ can be exhaustively described via its second-order moments.

Note, however, that the Gaussian assumption is not always justified. For instance, the statistics of rare events are better described by Poisson distributions, as is the case for cosmic rays detection [Gib40] or the emission of particles in PET scans [LQ00]. In general, the use of non-Gaussian distributions to model data is appropriate when one has some knowledege about the physical phenomena that generates its samples. In most cases, though, such model does not exist, so one has to resort to the most conservative assumption about the statistics of the data: the Gaussian assumption. In fact, if several independent factors play a role in the generation of the data, one may use the central limit theorem to argue that the sum of all their contributions yields a statistical behavior that can be well described by a Gaussian distribution [PP02]. Another selling argument for the Gaussian assumption is a numerical one: parameter estimation under the Gaussian model yields convex optimization problems that have analytic solutions. Furthermore, it has been shown that the Gaussian distribution leads to the largest Cramer-Rao bound (CRB) in a large class of parameter estimation problems [SB11]. This means that algorithms that estimate parameters using a variance minimization procedure based on a Gaussian model are in fact min-max optimal, i.e., they attain the best CRB-related performance in the worst case scenario. See [SB11] and [Par+13] for more details.

Parameter estimation. Assuming that $\boldsymbol{x}(t)$ is wide-sense stationary over T samples, $\{\boldsymbol{x}(0), \ldots, \boldsymbol{x}(T-1)\}$, we can write the estimators for (2.2) and (2.3) as

$$\hat{\boldsymbol{\mu}} = \frac{1}{T} \sum_{t=0}^{T-1} \boldsymbol{x}(t) \tag{2.8}$$

and

$$\hat{\boldsymbol{R}}(\tau) = \frac{1}{T-|\tau|} \sum_{t=0}^{T-1-|\tau|} \Big(\boldsymbol{x}(t+|\tau|) - \hat{\boldsymbol{\mu}}\Big)\Big(\boldsymbol{x}(t) - \hat{\boldsymbol{\mu}}\Big)^T . \tag{2.9}$$

The cross-spectral density matrices can be directly obtained from the DTFT of $\hat{\boldsymbol{R}}$ or via spectral estimation methods such as the periodogram or Welch's method [PW93].

2.2.3 Distance between multivariate time series

A consequence of the assumptions above is that we can compare two time series, $\boldsymbol{x}_i(t)$ and $\boldsymbol{x}_j(t)$, via the parameters used to describe their statistics. This is more appropriate than directly comparing their samples on a given realization and leads to superior results in classification and clustering tasks (see Section 2.5 for an example). Without loss of generality, we will consider that all time series are zero-mean, so that their parametrization may be done using just their cross-spectral density matrices. Furthermore, we will assume that the time series have been bandpass filtered and so their spectral content is supported on a set of frequencies denoted by $\mathcal{F}$.

Cross-spectrum distance. We define the cross-spectrum distance between two time series $\boldsymbol{x}_i(t)$ and $\boldsymbol{x}_j(t)$ as

$$d_S(\boldsymbol{x}_i(t), \boldsymbol{x}_j(t))^2 = \int_{\mathcal{F}} \delta^2(\boldsymbol{S}_i(f), \boldsymbol{S}_j(f)) \mathrm{d}f \ , \tag{2.10}$$

where $\boldsymbol{S}_i(f)$ and $\boldsymbol{S}_j(f)$ are the cross-spectral density matrices of $\boldsymbol{x}_i(t)$ and $\boldsymbol{x}_j(t)$, respectively, and δ is some distance between matrices. The usual choice for δ is the Frobenius distance

$$\delta_E^2(\boldsymbol{A}, \boldsymbol{B}) = \|\boldsymbol{A} - \boldsymbol{B}\|_F^2 = \sum_{k=1}^{d} \lambda_k^2 \ , \tag{2.11}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ are d-dimensional matrices and λ_k are the eigenvalues of $\boldsymbol{A} - \boldsymbol{B}$. However, one can show that cross-spectral density matrices are Hermitian positive definite matrices and, as such, they are usually treated in a Riemannian manifold with an intrinsic geometry [Bha09]. Therefore, it is more natural to compare cross-spectral densities using the geodesic distance of the HPD manifold, given by [Bha09]

$$\delta_R^2(\boldsymbol{A}, \boldsymbol{B}) = \left\| \log(\boldsymbol{A}^{-1/2} \boldsymbol{B} \boldsymbol{A}^{-1/2}) \right\|_F^2 = \sum_{k=1}^{d} \log^2(\lambda_k) \ , \tag{2.12}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ are d-dimensional HPD matrices and λ_k are the eigenvalues of $\boldsymbol{A}^{-1/2}\boldsymbol{B}\boldsymbol{A}^{-1/2}$ or, equivalently, $\boldsymbol{A}^{-1}\boldsymbol{B}$ (the logarithm of a HPD matrix is defined in Section 2.3.1). In Section 2.3, we give more details about the geometric features of the HPD manifold, as well as a justification for expression (2.12).

Note that when $\delta = \delta_R$ in (2.10), we have for any invertible matrix $\boldsymbol{A} \in \mathbb{R}^{d\times d}$,

$$d_S(\boldsymbol{A}\boldsymbol{x}_i(t), \boldsymbol{A}\boldsymbol{x}_j(t))^2 = \int_{\mathcal{F}} \delta_R^2(\boldsymbol{A}\boldsymbol{S}_i(f)\boldsymbol{A}^T, \boldsymbol{A}\boldsymbol{S}_j(f)\boldsymbol{A}^T)\mathrm{d}f \;, \tag{2.13}$$

$$= \int_{\mathcal{F}} \sum_{k=1}^{d} \log^2(\mu_k(f))\mathrm{d}f \;, \tag{2.14}$$

where $\mu_k(f)$ are the eigenvalues of matrix $(\boldsymbol{A}\boldsymbol{S}_i(f)\boldsymbol{A}^T)^{-1}(\boldsymbol{A}\boldsymbol{S}_j(f)\boldsymbol{A}^T)$. But,

$$(\boldsymbol{A}\boldsymbol{S}_i(f)\boldsymbol{A}^T)^{-1}(\boldsymbol{A}\boldsymbol{S}_j(f)\boldsymbol{A}^T) = \boldsymbol{A}^{-T}\boldsymbol{S}_i^{-1}(f)\boldsymbol{S}_j(f)\boldsymbol{A}^T \;, \tag{2.15}$$

which, by similarity, has the same eigenvalues of $\boldsymbol{S}_i^{-1}(f)\boldsymbol{S}_j(f)$. Therefore,

$$d_S(\boldsymbol{A}\boldsymbol{x}_i(t), \boldsymbol{A}\boldsymbol{x}_j(t))^2 = \int_{\mathcal{F}} \delta_R^2(\boldsymbol{S}_i(f), \boldsymbol{S}_j(f))\mathrm{d}f = d_S(\boldsymbol{x}_i(t), \boldsymbol{x}_j(t))^2 \;. \tag{2.16}$$

This shows that distance (2.10) is invariant to affine transformations of time series, a property that is very useful in practice. For instance, it is invariant to the choice of measurement scale, so the distance between two time series recorded in mV or μV is the same. Moreover, it is not unusual to observe mixing effects when working with data related to physical phenomena, such as the volume conduction in EEG [Con13] or the crosstalk in audio signal processing [Vin+06]. When such mixings may be approximated as the action of a linear operator, distance (2.10) is invariant to their effects as well.

The idea of comparing two time series based on their cross-spectral densities is not new and works have been developed in different research communities, such as in speech processing [GM76], radars [Bar08], and EEG analysis [Li+12]. In [Bas89], the author presents several distances between statistical distributions and shows how to use them in the context of time series.

Covariance distance. In some applications, one may not have enough time samples to obtain a good estimate of the cross-spectral density matrix. In such cases, it is more judicious to condense the information contained in the spectrum into a single parameter and then compare the corresponding parameters for each time series. One can do this by noticing that the inverse DTFT applied to the cross-spectral density matrices of a zero-mean $\mathcal{F}$-bandpass filtered time series $\boldsymbol{x}(t)$ gives

$$\int_{\mathcal{F}} \boldsymbol{S}(f)\mathrm{d}f = \boldsymbol{R}(0) = \mathbb{E}[\boldsymbol{x}(t)\boldsymbol{x}(t)^T] = \boldsymbol{C} \;, \tag{2.17}$$

which is the covariance matrix of $\boldsymbol{x}(t)$ and can be calculated without having to estimate its spectrum [Con13]. We may then define the covariance distance between time series $\boldsymbol{x}_i(t)$ and $\boldsymbol{x}_j(t)$ as

$$d_C(\boldsymbol{x}_i(t), \boldsymbol{x}_j(t))^2 = \delta_R^2(\boldsymbol{C}_i, \boldsymbol{C}_j) \ , \tag{2.18}$$

where $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$ are the covariance matrices of $\boldsymbol{x}_i(t)$ and $\boldsymbol{x}_j(t)$, respectively.

An application that has demonstrated good results using (2.18) as distance is EEG-based Brain-Computer Interfaces (BCI). In this kind of system, the realizations of the time series are usually quite short (in the order of one or two seconds), so the number of available samples is not large enough for ensuring good spectral estimation [Con13]. Using only covariance matrices as descriptors for EEG signals, [Bar+12] proposed a new framework for BCI classification and obtained state-of-the-art performance. In Section 2.4.2, we discuss with more details the use of distance (2.18) for comparing time series in EEG-related applications.

Hermitian positive definite matrices may be used to describe the statistics of other kinds of data not necessarily related to multivariate time series. In fact, distance (2.18) has been used to classify textures in images [Tuz+06] and movements in videos [Tuz+08], as well as detect structures in images [May06]. In [Pen06], the authors comment on the advantages of using distance (2.18) for manipulating data from diffusion tensor imaging (DTI), showing that δ_R avoids the 'swelling effect' typically present with δ_E – 'when considering Euclidean geometry to interpolate between two diffusion tensors, the determinant of the intermediate matrices may become strictly larger than the determinants of both original matrices, which from a physics point of view, is unacceptable.' [Har+18].

It is worth noting that in some applications one may not have enough available samples to expect a good estimate of the covariance matrix that describes the statistics of the data. For instance, if $\boldsymbol{x}(t) \in \mathbb{R}^{64}$ then its covariance matrix $\boldsymbol{C}$ has dimensions 64×64. Therefore, if we estimate $\boldsymbol{C}$ using the estimator in (2.9) with $\tau = 0$, there should be at least 4096 samples available for the estimate $\hat{\boldsymbol{C}}$ to have a chance of not being rank-deficient [Pri83]. A common approach for alleviating such problem is to use a regularization term that adds a weighted Identity matrix to $\hat{\boldsymbol{C}}$, with the optimal weight being determined from the data. This technique is often called 'shrinkage' in the literature and many methods have been proposed for determining the weight to assign to the regularization term [Che+10]. More recently, tools from random matrix theory have been applied to understand the statistical distribution of the eigenvalues of high-dimensional covariance matrices and improve algorithms based on them. See [CM14] for an overview of this topic and [Tio+19] for a work that proposes a way of improving the estimation of distance (2.12)

between two high-dimensional covariance matrices estimated from a limited number of samples.

2.3 Riemannian geometry of the HPD manifold

In this section, we introduce the geometry of the manifold of Hermitian positive definite matrices (HPD). We define some basic notions, such as geodesic distance and center of mass, as well as more sophisticated ones, such as parallel transport of tangent vectors and statistical descriptions of data points defined in the HPD manifold. We also discuss how to perform classification tasks when a dataset is composed of HPD data points.

2.3.1 Basic definitions and notation

Let $\mathcal{P}(d)$ be the set of $d \times d$ Hermitian positive definite (HPD) matrices defined as

$$\mathcal{P}(d) = \left\{ \boldsymbol{C} \in \mathbb{C}^{d \times d} \mid \boldsymbol{C}^H = \boldsymbol{C},\ \boldsymbol{x}^H \boldsymbol{C} \boldsymbol{x} > 0,\ \forall \boldsymbol{x} \in \mathbb{C}^d,\ \boldsymbol{x} \neq \boldsymbol{0} \right\}, \tag{2.19}$$

where $\boldsymbol{C}^H$ is the conjugate-transpose version of $\boldsymbol{C}$. It is known from basic linear algebra that every matrix $\boldsymbol{C} \in \mathcal{P}(d)$ can be decomposed as

$$\boldsymbol{C} = \boldsymbol{Q} \boldsymbol{\Lambda} \boldsymbol{Q}^H , \tag{2.20}$$

with $\boldsymbol{\Lambda}$ a diagonal matrix,

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_d \end{bmatrix} \quad \text{and} \quad \lambda_i > 0 , \tag{2.21}$$

and $\boldsymbol{Q}^H \boldsymbol{Q} = \boldsymbol{I}_d$.

The application of an analytic function $f : \mathbb{R} \to \mathbb{R}$ to $\boldsymbol{C} \in \mathcal{P}(d)$ is defined as

$$f(\boldsymbol{C}) = \boldsymbol{Q} f(\boldsymbol{\Lambda}) \boldsymbol{Q}^H , \tag{2.22}$$

where f is applied to each eigenvalue of $\boldsymbol{C}$. For instance, the square root of $\boldsymbol{C}$ is

$$\boldsymbol{C}^{1/2} = \boldsymbol{Q} \boldsymbol{\Lambda}^{1/2} \boldsymbol{Q}^H , \tag{2.23}$$

and its logarithm is

$$\log(\boldsymbol{C}) = \boldsymbol{Q} \log(\boldsymbol{\Lambda}) \boldsymbol{Q}^H . \tag{2.24}$$

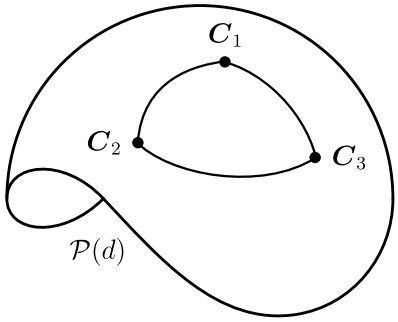

Fig. 2.1: The manifold $\mathcal{P}(d)$ is portrayed as a surface with non-positive curvature. The drawn lines are the shortest paths between each pair of points in the HPD manifold, also known as geodesics. Note that, in this space, the sum of angles in a triangle is not 180 degrees.

HPD manifold. Matrices in $\mathcal{P}(d)$ lie in a manifold [Bha09], a set of points with the property that the neighborhood of each $\boldsymbol{C} \in \mathcal{P}(d)$ can be bijectively mapped onto an Euclidean space, also known as its tangent space $T_{\boldsymbol{C}}\mathcal{P}(d)$. Intuitively, we say that the neighbourhood of every point in the manifold is flat, but the whole manifold has a non-positive curvature [Moa05], as portrayed in Figure 2.1. Because $\mathcal{P}(d)$ is an open subspace of the set $\mathcal{H}(d)$ of hermitian matrices in $\mathbb{C}^{d\times d}$, we can identify its tangent space as simply being $\mathcal{H}(d)$ [Abs+09]. If we endow every tangent space of a manifold with a metric that changes smoothly along its elements, we say that we have a Riemannian manifold. In this case, fundamental geometric notions are naturally defined, such as geodesic (shortest curve joining two points), distance between two points (length of the geodesic connecting them), the center of mass of a set of points, etc.

Affine-invariant Riemannian metric. There are several possible choices of metric for $\mathcal{P}(d)$ and each one induces a different geometry that can be more or less adequate according to the applications that we are interested in. A metric that is particularly relevant is the one defined for tangent vectors $\boldsymbol{\eta}, \boldsymbol{\xi} \in T_{\boldsymbol{C}}\mathcal{P}(d)$ as

$$\langle \boldsymbol{\eta}, \boldsymbol{\xi} \rangle_{\boldsymbol{C}} = \operatorname{tr}(\boldsymbol{C}^{-1}\boldsymbol{\eta}\boldsymbol{C}^{-1}\boldsymbol{\xi}) \;, \tag{2.25}$$

where $\boldsymbol{C} \in \mathcal{P}(d)$ and $\operatorname{tr}(\cdot)$ denotes the trace operator. Note that, for any invertible matrix $\boldsymbol{M} \in \mathbb{R}^{d\times d}$, we have

$$\begin{aligned}
\langle \boldsymbol{M}\boldsymbol{\eta}\boldsymbol{M}^T, \boldsymbol{M}\boldsymbol{\xi}\boldsymbol{M}^T \rangle_{\boldsymbol{M}\boldsymbol{C}\boldsymbol{M}^T} &= \operatorname{tr}((\boldsymbol{M}\boldsymbol{C}\boldsymbol{M}^T)^{-1}(\boldsymbol{M}\boldsymbol{\eta}\boldsymbol{M}^T)(\boldsymbol{M}\boldsymbol{C}\boldsymbol{M}^T)^{-1}(\boldsymbol{M}\boldsymbol{\xi}\boldsymbol{M}^T)) \;, \\
&= \operatorname{tr}(\boldsymbol{M}^{-T}\boldsymbol{C}^{-1}\boldsymbol{M}^{-1}\boldsymbol{M}\boldsymbol{\eta}\boldsymbol{M}^T\boldsymbol{M}^{-T}\boldsymbol{C}^{-1}\boldsymbol{M}^{-1}\boldsymbol{M}\boldsymbol{\xi}\boldsymbol{M}^T)) \;, \\
&= \operatorname{tr}(\boldsymbol{C}^{-1}\boldsymbol{\eta}\boldsymbol{C}^{-1}\boldsymbol{\xi})) \;, \\
&= \langle \boldsymbol{\eta}, \boldsymbol{\xi} \rangle_{\boldsymbol{C}} \;.
\end{aligned} \tag{2.26}$$

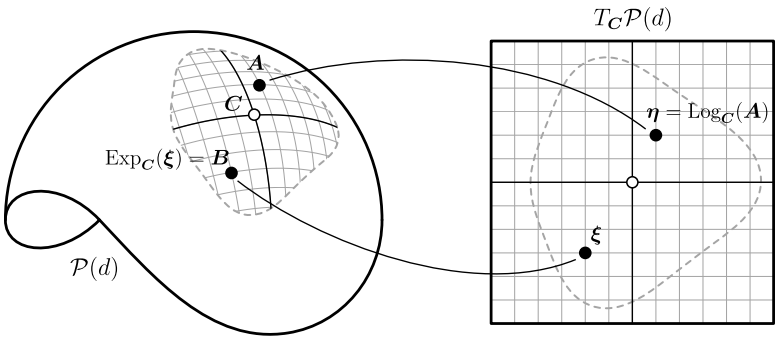

Fig. 2.2: Depiction of the relation between the HPD manifold and a tangent space with reference at point $\boldsymbol{C}$.

Because of this property, metric (2.25) is named the Affine-Invariant Riemannian metric (AIRM) and is known as the 'natural' Riemannian metric for the HPD manifold [Bha09; Pen06; Moa05]. Another reason for its relevance is the connection of the AIRM to the Fisher-Rao metric when considering zero-mean multivariate Gaussian distributions [Ama16].

Tangent space. The map that transforms matrices in $\mathcal{P}(d)$ into tangent vectors in $T_{\boldsymbol{C}}\mathcal{P}(d)$ is called the logarithmic map and, when AIRM is used as metric, it is given by [Bha09]

$$\begin{aligned} \mathrm{Log}_{\boldsymbol{C}} \;:\; \mathcal{P}(d) &\rightarrow T_{\boldsymbol{C}}\mathcal{P}(d) \\ \boldsymbol{A} &\mapsto \boldsymbol{C}^{1/2}\log(\boldsymbol{C}^{-1/2}\boldsymbol{A}\boldsymbol{C}^{-1/2})\boldsymbol{C}^{1/2}\,. \end{aligned} \tag{2.27}$$

Conversely, the map that transforms a tangent vector in $T_{\boldsymbol{C}}\mathcal{P}(d)$ into a matrix in $\mathcal{P}(d)$ is called the exponential map, and is given by

$$\begin{aligned} \mathrm{Exp}_{\boldsymbol{C}} \;:\; T_{\boldsymbol{C}}\mathcal{P}(d) &\rightarrow \mathcal{P}(d) \\ \xi &\mapsto \boldsymbol{C}^{1/2}\exp(\boldsymbol{C}^{-1/2}\xi\boldsymbol{C}^{-1/2})\boldsymbol{C}^{1/2}\,. \end{aligned}$$

Figure 2.2 illustrates the concepts defined above.

Geodesic distance. Thus far, we have only considered the local geometry of the HPD manifold. We will now extend our discussion to the whole geometry of $\mathcal{P}(d)$, starting with the distance between two points in the manifold.

Let $\gamma : [0,1] \rightarrow \mathcal{P}(d)$ be a differentiable curve defined in $\mathcal{P}(d)$. The length of γ is given by [Bha09]

$$L(\gamma) = \int_0^1 \|\gamma^{-1/2}(t)\dot{\gamma}(t)\gamma^{-1/2}(t)\|_2 \,\mathrm{d}t\,, \tag{2.28}$$

where $\dot{\gamma}(t)$ is the instantaneous speed vector of $\gamma(t)$. The *geodesic distance* between two points in $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{P}(d)$ is defined as

$$\delta_R(\boldsymbol{A}, \boldsymbol{B}) = \inf \left\{ L(\gamma) \text{ such that } \gamma(0) = \boldsymbol{A} \text{ and } \gamma(1) = \boldsymbol{B} \right\} , \tag{2.29}$$

where the curve γ that attains the infimum is called the geodesic path between points $\boldsymbol{A}$ and $\boldsymbol{B}$. The explicit expression for (2.29) when AIRM is chosen as metric for $\mathcal{P}(d)$ is

$$\delta_R(\boldsymbol{A}, \boldsymbol{B}) = \left\| \log \left(\boldsymbol{A}^{-1/2} \boldsymbol{B} \boldsymbol{A}^{-1/2} \right) \right\|_F \tag{2.30}$$

and the geodesic path γ linking $\boldsymbol{A}$ and $\boldsymbol{B}$ is

$$\gamma(t) = \boldsymbol{A}^{1/2} \left(\boldsymbol{A}^{-1/2} \boldsymbol{B} \boldsymbol{A}^{-1/2} \right)^t \boldsymbol{A}^{1/2} . \tag{2.31}$$

The reader is referred to [Bha09] for a demonstration of these results.

Distance δ_R has many interesting properties. For instance, for every invertible matrix $\boldsymbol{M} \in R^{d \times d}$, we have that

$$\delta_R(\boldsymbol{M} \boldsymbol{A} \boldsymbol{M}^T, \boldsymbol{M} \boldsymbol{B} \boldsymbol{M}^T) = \delta_R(\boldsymbol{A}, \boldsymbol{B}) , \tag{2.32}$$

which is a consequence of the affine-invariance of the AIRM. It is also invariant to inversion, so that

$$\delta_R(\boldsymbol{A}^{-1}, \boldsymbol{B}^{-1}) = \delta_R(\boldsymbol{A}, \boldsymbol{B}) . \tag{2.33}$$

Because of these and other properties (see [Bha09] for more of them), the AIRM-induced distance has found great popularity in geometry-aware algorithms for processing HPD matrices [MS00; WV05; Tuz+06; May06; Bar08; Li+09; Bar+12].

In this work, whenever we refer to $\mathcal{P}(d)$, we will be implicitly assuming that it has been equipped with the AIRM. However, it is possible to define other distances in the HPD manifold, which may have certain properties that justify their use instead of the AIRM distance in some contexts. See [Ars+07] and [Bha+18] for two examples.

Center of mass. Once we have the expression for the distance between any two points $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{P}(d)$, it is natural to ask what is the HPD matrix that is equidistant to $\boldsymbol{A}$ and $\boldsymbol{B}$ in terms of (2.11). We denote such matrix $\boldsymbol{A} \# \boldsymbol{B}$ and, from (2.28) for $t = 1/2$, we obtain

$$\boldsymbol{A} \# \boldsymbol{B} = \boldsymbol{A}^{1/2} \left(\boldsymbol{A}^{-1/2} \boldsymbol{B} \boldsymbol{A}^{-1/2} \right)^{1/2} \boldsymbol{A}^{1/2} , \tag{2.34}$$

which is the mid-way point in the geodesic linking $\boldsymbol{A}$ and $\boldsymbol{B}$. By definition, matrix $\boldsymbol{A} \# \boldsymbol{B}$ satisfies [Bha09]

$$\boldsymbol{A} \# \boldsymbol{B} = \underset{\boldsymbol{M} \in \mathcal{P}(d)}{\operatorname{argmin}} \left(\delta_R^2(\boldsymbol{A}, \boldsymbol{M}) + \delta_R^2(\boldsymbol{B}, \boldsymbol{M}) \right) , \tag{2.35}$$

which, in words, means that it is the point in $\mathcal{P}(d)$ that minimizes the dispersion of $\boldsymbol{A}$ and $\boldsymbol{B}$. This is why $\boldsymbol{A}\#\boldsymbol{B}$ is also named the center of mass of points $\boldsymbol{A}$ and $\boldsymbol{B}$. Note, also, that when $\boldsymbol{A}$ and $\boldsymbol{B}$ are strictly positive scalars, $\boldsymbol{A}\#\boldsymbol{B}$ is their geometric mean $\sqrt{\boldsymbol{A}\boldsymbol{B}}$. This explains why many researchers [Bha09; Con13; Arn+13; Mas+18] adopt the term 'geometric mean' to refer to the center of mass of a set of HPD matrices

We can extend the above definition to a set of K matrices,

$$\mathcal{X} = \{\boldsymbol{C}_1, \ldots, \boldsymbol{C}_K\} \subset \mathcal{P}(d) \;, \tag{2.36}$$

so that

$$\boldsymbol{M}^{\mathcal{X}} = \underset{\boldsymbol{M}\in\mathcal{P}(d)}{\operatorname{argmin}} \sum_{i=1}^{K} \delta_R^2(\boldsymbol{M}, \boldsymbol{C}_i) \;. \tag{2.37}$$

In the literature, matrix $\boldsymbol{M}^{\mathcal{X}}$ is sometimes called the center of mass of $\mathcal{X}$, its geometric means, its Fréchet mean, or also its Karcher mean [Bha09]. When $K \geq 3$, there is no closed form solution for (2.37) in general, however, due to the non-positive curvature of the HPD manifold, it is possible to show that there always exists a solution for its optimization problem [Kar77]. With this in mind, many researchers have proposed procedures for calculating the center of mass of a set of HPD matrices iteratively. In [Bar08], the author uses an algorithm based on back-and-forth projections between the HPD manifold and its tangent space in order to converge to the solution of (2.37). More recently, [Con+17] proposed a fixed-point algorithm for calculating the geometric mean as well as a whole family of other means of HPD matrices called power means. See [Con+19] for applications of the power means algorithm to BCI classification tasks.

Parallel transport. For a comparison between two tangent vectors to make sense, they have to be defined in the same tangent space. When this is not the case, one has to use the notion of parallel transport [Abs+09], which transforms tangent vectors in a given tangent space into tangent vectors of another tangent space, without changing the inner product of the transformed vectors.

The parallel transport taking tangent vectors from $T_{\boldsymbol{A}}\mathcal{P}(d)$ to $T_{\boldsymbol{B}}\mathcal{P}(d)$ is given by

$$\begin{array}{rccc} \boldsymbol{P}_{\boldsymbol{A}\to\boldsymbol{B}} & : & T_{\boldsymbol{A}}\mathcal{P}(d) & \to & T_{\boldsymbol{B}}\mathcal{P}(d) \\ & & \boldsymbol{\eta} & \mapsto & (\boldsymbol{A}\#\boldsymbol{B})\boldsymbol{A}^{-1}\boldsymbol{\eta}\boldsymbol{A}^{-1}(\boldsymbol{A}\#\boldsymbol{B}) \end{array} \tag{2.38}$$

and we have that, for any $\boldsymbol{\eta}, \boldsymbol{\xi} \in T_{\boldsymbol{A}}\mathcal{P}(d)$,

$$\langle \boldsymbol{\eta}, \boldsymbol{\xi} \rangle_A = \left\langle \boldsymbol{P}_{\boldsymbol{A}\to\boldsymbol{B}}(\boldsymbol{\eta}), \boldsymbol{P}_{\boldsymbol{A}\to\boldsymbol{B}}(\boldsymbol{\xi}) \right\rangle_B \;. \tag{2.39}$$

We refer the interested reader to [Yai+19] for a demonstration of this expression.

2.3.2 Statistics in the HPD manifold

When using HPD matrices as features to describe real data, it might be useful to model the variability of such matrices by assuming that there is some statistical law that generated the data points. A first option would be to model the HPD matrices as coming from a Wishart distribution [Wis28], as it is traditionally done in the statistics literature [LEE+94; LS97; HSJ10]. However, such distribution does not consider all aspects of the intrinsic geometry of the HPD manifold.

Riemannian Gaussian. Recently, [Sai+17] has proposed the Riemannian Gaussian distribution, which generalizes the notion of Gaussian distributions in Euclidean space to the HPD manifold. In the same way as for its Euclidean counterpart, Riemannian Gaussians are parametrized using two parameters: a HPD matrix $\boldsymbol{M} \in \mathcal{P}(d)$ describing the centrality of the distribution and a strictly positive scalar ε describing its dispersion around the center. The expression for its probability density function is given by

$$p(\boldsymbol{C}) = \frac{1}{\zeta(\varepsilon)} \exp\left(-\frac{\delta_R^2(\boldsymbol{C}, \boldsymbol{M})}{2\varepsilon^2}\right) , \tag{2.40}$$

where $\zeta(\varepsilon)$ is a normalization factor that depends on ε. Building on the work from [Pen06], the authors from [Sai+17] determined expressions for the maximum likelihood estimators (MLE) of the parameters of a Riemannian Gaussian distribution. We have that for a set of matrices generated by (2.40),

$$\mathcal{X} = \{\boldsymbol{C}_1, \ldots, \boldsymbol{C}_K\} \subset \mathcal{P}(d) , \tag{2.41}$$

the MLE for $\boldsymbol{M}$ is the geometric mean of the set of matrices

$$\hat{\boldsymbol{M}}_K = \boldsymbol{M}^{\mathcal{X}} = \underset{\boldsymbol{M} \in \mathcal{P}(d)}{\operatorname{argmin}} \sum_{k=1}^{K} \delta_R^2(\boldsymbol{M}, \boldsymbol{C}_k) , \tag{2.42}$$

and the MLE for ε is

$$\hat{\varepsilon}_K = \Phi\left(\frac{1}{K} \sum_{k=1}^{K} \delta_R^2(\hat{\boldsymbol{M}}_K, \boldsymbol{C}_k)\right) , \tag{2.43}$$

where Φ is a strictly increasing (and, therefore, bijective) function detailed in [Sai+17].

Ref. [Sai+17] has also presented the concept of mixtures of Riemannian Gaussians distributions, which allows for more flexible models of the statistics of a set containing HPD data points.

Parametrization. A simple assumption that one can make regarding the statistics of the data points from a dataset $\mathcal{X} \subset \mathcal{P}(d)$ is that they are generated by a mixture

of Riemannian Gaussian distributions, where each mixture is related to one of the classes of the dataset and the dispersion around the class mean for all the mixtures is assumed the same. More precisely, consider dataset

$$\mathcal{X} = \left\{ (\boldsymbol{C}_i, \ell_i) \text{ for } i = 1, \ldots, K \right\} , \tag{2.44}$$

with data points $\boldsymbol{C}_i \in \mathcal{P}(d)$ and class labels $\ell_i \in \{1, \ldots, L\}$. We assume that the statistical distribution of $\mathcal{X}$ can be sufficiently well described by a set of a few parameters and denote such description by

$$\Theta_{\mathcal{X}} \sim \left\{ \boldsymbol{M}^{\mathcal{X}}, \boldsymbol{M}_1^{\mathcal{X}}, \ldots, \boldsymbol{M}_L^{\mathcal{X}}, \sigma^{\mathcal{X}} \right\} , \tag{2.45}$$

where $\boldsymbol{M}^{\mathcal{X}}$ is the geometric mean of all the data points in $\mathcal{X}$, $\boldsymbol{M}_1^{\mathcal{X}}, \ldots, \boldsymbol{M}_L^{\mathcal{X}}$ are the geometric means of the points belonging to each class, and $\sigma^{\mathcal{X}}$ is the dispersion of the points in $\mathcal{X}$ around $\boldsymbol{M}^{\mathcal{X}}$, defined as

$$(\sigma^{\mathcal{X}})^2 = \frac{1}{K} \sum_{k=1}^{K} \delta_R^2(\boldsymbol{C}_i, \boldsymbol{M}^{\mathcal{X}}) . \tag{2.46}$$

In the rest of this thesis, we use this parametrization to describe the statistics of a dataset containing HPD data points.

Comparing statistics. The problem of comparing the statistical distributions of two datasets, $\mathcal{X}$ and $\mathcal{Y}$, has attracted much attention in the statistical literature for a long time [KL51; AS66; Bas89; Ama16]. There are mainly two approaches: parametric and non-parametric.

Non-parametric distances do not make any modelling assumptions regarding distributions $\Theta_{\mathcal{X}}$ and $\Theta_{\mathcal{Y}}$ and are mostly based on the pairwise distances between all elements of datasets $\mathcal{X}$ and $\mathcal{Y}$. Many distances have been proposed in this context, such as the Kullback-Leiber divergence [KL51] (which is not really a distance), the maximum-mean discrepancy [Bor+06] and the Wasserstein distance [PC19]. These distances have found recent popularity in the neural networks community, being used in generative adversarial networks (GAN), where the cost function to be optimized is one that is based on the distance between the statistics of two datasets [Goo+14; Arj+17].

Parametric distances are based on the distance between the parameters that describe the statistics of the dataset. In the context of HPD matrices, description (2.45) may be used to define a notion of distance between $\Theta_{\mathcal{X}}$ and $\Theta_{\mathcal{Y}}$ as

$$\mathcal{W}^2(\Theta_{\mathcal{X}}, \Theta_{\mathcal{Y}}) = \delta_R^2(\boldsymbol{M}^{\mathcal{X}}, \boldsymbol{M}^{\mathcal{Y}}) + \sum_{c=1}^{L} \delta_R^2(\boldsymbol{M}_c^{\mathcal{X}}, \boldsymbol{M}_c^{\mathcal{Y}}) + \log^2 \left(\frac{\sigma^{\mathcal{X}}}{\sigma^{\mathcal{Y}}} \right) , \tag{2.47}$$

which is zero if, and only if, the statistical distributions of $\mu_{\mathcal{X}}$ and $\mu_{\mathcal{Y}}$ are described by the same set of parameters.

2.3.3 Classification in the HPD manifold

Classification is the action of assigning a class to an object. Some examples are: deciding whether an image portrays a dog, a cat, or a sheep, if an EEG recording corresponds to light sleep or deep sleep, to which category a given text extract should be assigned to, etc. In mathematical terms, we say that there exists a mapping c that relates elements from a set of objects $\mathcal{O}$ to labels from a set $\mathcal{L}$ and that our goal is to find the function $h : \mathcal{O} \to \mathcal{L}$ from a class of hypothesis $\mathcal{H}$ that ressembles the most to c in some sense to be defined. We call the elements of $\mathcal{H}$ 'classifiers' and say that a good candidate is one that maps the objects of $\mathcal{O}$ to $\mathcal{L}$ the same way as c.

We measure how well $h \in \mathcal{H}$ approximates c via the probability of the two functions assigning different classes to the same $\boldsymbol{x} \in \mathcal{O}$. This is called the generalization error (or risk), defined as

$$R(h) = \text{Prob}\{h(\boldsymbol{x}) \neq c(\boldsymbol{x})\} = \mathbb{E}\left[\mathbf{1}_{\{h(\boldsymbol{x})\neq c(\boldsymbol{x})\}}\right] , \tag{2.48}$$

where the expectation is taken with respect to the statistical distribution of data points $\boldsymbol{x} \in \mathcal{O}$ and $\mathbf{1}_{\mathcal{B}}$ is the indicator function for set $\mathcal{B}$. The goal, then, is to determine which hypothesis in $\mathcal{H}$ has the smallest generalization error. To do so, we choose a method $\mathcal{M}$ that searches for a classifier

$$h_{\mathcal{M}} = \underset{h\in\mathcal{H}}{\text{argmin}}\ R(h) . \tag{2.49}$$

More concretely, choosing a method $\mathcal{M}$ involves choosing how the statistics of the data points should be modelled, what is the class of hypothesis functions to be considered, and which optimization algorithm should be used for solving (2.49). Some examples of methods are logistic regression, support vector machines, and neural networks [Bis07].

Note that, in practice, the generalization error of a hypothesis is never accessible, since both the distribution of $\boldsymbol{x}$ and the map c are unkown. Still, if one has access to a set of labeled examples,

$$\mathcal{X} = \Big\{(\boldsymbol{x}_k, \ell_k) \text{ for } k = 1, \ldots, K\Big\} \subset \mathcal{O} \times \mathcal{L} , \tag{2.50}$$

where $\ell_k = c(\boldsymbol{x}_k)$, a proxy for the generalization error may be defined: the empirical error, given by

$$R^{\mathcal{X}}(h) = \frac{1}{K}\sum_{k=1}^{K} \mathbf{1}_{\{h(\boldsymbol{x}_k)\neq c(\boldsymbol{x}_k)\}} . \tag{2.51}$$

Then, the classifier that approximates c when using method $\mathcal{M}$ to minimize the empirical error on dataset $\mathcal{X}$ is

$$h_{\mathcal{M}}^{\mathcal{X}} = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \, R^{\mathcal{X}}(h) \, . \tag{2.52}$$

It is important to note that the classifier obtained from (2.52) does not minimize the generalization error, but the empirical error, in a procedure that is called empirical risk minimization under a supervised setting (because we consider having labeled examples in $\mathcal{X}$) [SSBD14]. Consequently, the real goal of minimizing the generalization error is never attainable in practice; we can only minimize the empirical error and hope that it does not differ too much from the generalization error.

There has been many studies investigating the relation between the generalization error of $h_{\mathcal{M}}$ and $h_{\mathcal{M}}^{\mathcal{X}}$. From [SSBD14], we have that, with probability $1 - \delta$,

$$R(h_{\mathcal{M}}^{\mathcal{X}}) \leq R(h_{\mathcal{M}}) + 2 \sqrt{\frac{\log(|\mathcal{H}|) + \log\left(\frac{2}{\delta}\right)}{2K}} \, , \tag{2.53}$$

where the family of hypothesis $\mathcal{H}$ is assumed to be of finite size and $|\mathcal{H}|$ is how many elements it has; K is the number of samples in $\mathcal{X}$. A consequence of (2.53) is that the larger the set of hypothesis $\mathcal{H}$ is, the looser the upper bound for $R(h_{\mathcal{M}}^{\mathcal{X}})$ is, and, therefore, it is harder to know whether $R(h_{\mathcal{M}}^{\mathcal{X}})$ is close to $R(h_{\mathcal{M}})$ or not. In other words, it is harder to control the empirical error of the classifier when the family of hypothesis is 'too rich' [SSBD14]. Evidently, one might argue that, in reality, the family of classifiers $\mathcal{H}$ is always infinite, since the parameters of most models used in practice are continuous. However, it is possible to define a notion of 'richness' of a family of classifiers which is infinite and use it to bound the difference between empirical error and generalization error in a similar way to (2.53). See [SSBD14] for more details.

Cross-validation. In practice, the generalization error of a classifier $h_{\mathcal{M}}^{\mathcal{X}}$ is never accessible, but one may get an estimate of $R(h_{\mathcal{M}}^{\mathcal{X}})$ by evaluating its empirical error on a set of labeled data points that were not considered during the minimization procedure leading to its estimation. Based on this idea, one may assess how good the classifiers proposed by a method $\mathcal{M}$ are for a certain dataset $\mathcal{X}$ using a cross-validation procedure [Bis07]:

- Partition $\mathcal{X}$ into F subsets containing (approximately) the same number of elements and the same number of examples from each class in $\mathcal{L}$. We have

$$\mathcal{X} = \mathcal{X}_1 \cup \cdots \cup \mathcal{X}_F \, . \tag{2.54}$$

- Define the train and test folds,

$$\mathcal{X}_{\text{train}}^{(f)} = \mathcal{X} \setminus \mathcal{X}_f \quad \text{and} \quad \mathcal{X}_{\text{test}}^{(f)} = \mathcal{X}_f \,, \tag{2.55}$$

  and the empirical error calculated on each test fold,

$$R_{\mathcal{M}}^{(f)} = R^{\mathcal{X}_{\text{test}}^{(f)}} \left(h_{\mathcal{M}}^{\mathcal{X}_{\text{train}}^{(f)}} \right) \,. \tag{2.56}$$

- Define the average performance of $\mathcal{M}$ on dataset $\mathcal{X}$ by

$$R_{\mathcal{M}} = \frac{1}{F} \sum_{f=1}^{F} R_{\mathcal{M}}^{(f)} \,, \tag{2.57}$$

  which is the average empirical error of the classifiers proposed by $\mathcal{M}$ on each test fold; the expected value of $R_{\mathcal{M}}$ is the generalization error of $h_{\mathcal{M}}^{\mathcal{X}}$ [SSBD14].

MDM classifier. When $\mathcal{O} = \mathcal{P}(d)$, there are mainly two approaches for obtaining a classifier that predicts well the labels from the data. The first one is to consider classifiers that take into account the intrinsic geometry of the HPD manifold and work directly with the HPD matrices. An example is the k-nearest neighbors classifier, which assigns to each data point $\boldsymbol{x}_i \in \mathcal{P}(d)$ the prevalent class among the k nearest points to $\boldsymbol{x}_i$ in the labeled dataset $\mathcal{X}$. To respect the intrinsic geometry of the data space, one may use the AIRM-induced distance to determine what are the k closest points to $\boldsymbol{x}_i$. This kind of classifier has been used with HPD data in [Tuz+06] and [Har+18] for classifying textures in images and in [Li+09] for classifying sleep stages from EEG recordings.

A more robust classifier is one that estimates the center of mass for each class of elements in $\mathcal{X}$ and assigns to an unlabeled data point the class of the closest class mean. Traditionally, such method is called the nearest-centroid classifier [Bis07], but it has also been named the minimum distance to mean (MDM) classifier by [Bar+12] when data is defined in the HPD manifold. This type of classifier has been used in [Bar+12] for BCI classification tasks and in [Bar08] for radar applications. It is worth mentioning that the statistical modelling presented in Section 2.3.2 fits well the implicit assumptions behind the MDM classifier, since it assumes that the dataset can be sufficiently well described by its geometric mean, dispersion, and class means.

Tangent space classifier. Alternatively, one may project the data points from $\mathcal{X}$ into the tangent space of the HPD manifold at some reference point (usually the geometric mean of the dataset) and then classify the tangent vectors. A clear advantage of this approach is that the tangent vectors define a linear vector space that can be operated using simple linear algebra. This, in turn, makes it possible to use classifiers from the traditional pattern recognition literature defined in Euclidean space [Bis07]. On the

other hand, the dimension of the tangent vectors for $P(d)$ is $d \times (d+1)/2$, which, in practice, may lead to situations where one has tangent vectors that are bigger than the number of samples and may cause problems for some classification algorithms. Classification based on tangent vectors has been used in [Har+18] for classifying textures in images, in [Tuz+08] for detecting pedestrians in videos, and in [Bar+12] for BCI classification.

2.4 Riemannian geometry for EEG signals

In this section, we show how the Riemannian geometric framework presented in previous sections can be applied to the analysis and classification of electroencephalographic recordings. We begin with a brief presentation of concepts related to the electrical activity in the brain (how it is generated, measured, and processed) as well as some important markers that are often used to classify EEG signals. Then, we show how to parametrize EEG signals via HPD matrices and give an overview of recent BCI applications that use the Riemannian geometric framework.

2.4.1 Basic concepts about EEG

Generation. Our brain is composed of billions of neurons that communicate with each other mainly via synapses. This communication is based on the exchange of chemical substances between the neurons and has the effect of producing electrical activity at their membranes (difference of electric potential between the outer and inner membrane). When neurons at a certain region activate together for some particular reason (cognitive load, homeostasis, etc.), their electric activity tend to synchronize and become measurable at a macroscopic scale. Equipements that measure this electric activity are called electroencephalograms and the signals they record are called electroencephalographic signals [Kan+91].

Measurement. In a typical EEG recording, the experimenter puts several electrodes on different parts of the scalp of a subject and asks him or her to perform a sequence of cognitive tasks. The recorded signals are then amplified (the typical amplitude of the EEG activity of an adult is around 10 to 50 μV), filtered, downsampled, and stored in digital form. To facilitate comparisons between experiments, it is common practice to put the electrodes on standard positions. See Figure 2.3 for an example.

A fundamental assumption of experiments using EEG is that the activity recorded by sensors at certain positions may serve as a sign of brain activity at that given location. Based on this hypothesis, one may try to infer which cognitive task a subject is executing just from the information coming from the EEG signals. For instance, it is known that when a human closes his eyes, the occipital region (region related to the

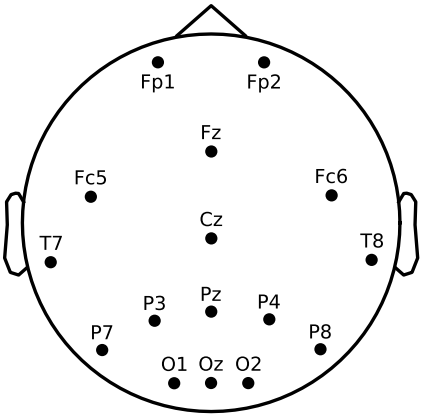

Fig. 2.3: Position of 16 electrodes used in an experiment at the GIPSA-lab (dataset `ALPHA.EEG.2017-GIPSA`).

vision apparatus of the brain, located at the back of the head) generates EEG signals oscillating at approximately 12 Hz, also known as alpha waves. Therefore, in an experiment where a person closes his eyes during a few seconds and then opens it, one might expect to detect relevant activity from signals recorded at electrodes `O1`, `O2` and `Oz` (see Figure 2.3). Unfortunately, though, EEG is known for its poor spatial resolution: measuring electric activity in a given electrode does not necessarily mean that the region of the brain located some centimeters underneath the electrode is active. This happens because cortical current must go through several layers of brain tissue with different conductivity before attaining the scalp. As a consequence, at every spatial scalp position, the recorded activity is a mixture of the underlying brain sources. This phenomenon is called volume conduction effect [NS05]. On the other hand, EEG has very good temporal resolution, allowing the detection of changes in brain activity in the order of milliseconds. Also, EEG signals are very popular in experimental settings because of its relatively low price as compared to other options (few hundred euros for an EEG setup versus several thousand euros for a fMRI or MEG machine). Furthermore, there has been many works in the literature investigating ways of inverting the volume conduction effect and recovering the activity at the brain level with spatial precision [DPm99].

Processing. Once the EEG recordings are stored, one may use signal processing tools to analyze the data. A first important step is to filter artefacts, otherwise one may make conclusions about the activity of the brain based on elements that are not physiologically relevant. Two artefacts that are commonly removed are: the spectral peak at 50 Hz, due to the power line frequency, and perturbations, affecting especially the frontal electrodes (eg. `Fp1` and `Fp2` in Figure 2.3), due to eye-movements [SC07]. Then, the signals are bandpass filtered to some frequency interval carrying physiological information relevant for the analysis being done. If the EEG experiment consists of tasks that are repeated several times, the signals are

cut into several epochs (also called trials), which can then be combined, averaged, compared, etc.

Oscillations. There has been a great number of experiments where the EEG activity of a subject was recorded while performing different tasks. Thanks to these studies, certain features from EEG signals have been established as relevant for inferring qualitative aspects of the brain activity that generated them. One of them is related to how the EEG signals oscillate, i.e., their spectral content. It has been observed that, for different cognitive tasks, the EEG in different parts of the brain oscillate differently [Buz06]. Some examples of oscillations are:

- **Delta waves** are oscillations in the 0.5-4 Hz band, usually associated to a deep state of sleep.
- **Alpha waves** are oscillations in the 8-12 Hz band, usually associated to a relaxed state of mind (e.g. being with the eyes closed). These waves appear mostly in the occipital region of the brain (electrodes `O1`, `O2`, and `Oz` in Figure 2.3).
- **Mu waves** are also oscillations in the 8-12 Hz band, but mostly linked to voluntary motor activity. They appear in the motor cortex region (electrode `Cz` in Figure 2.3).
- **Beta waves** are oscillations in the 12-30 Hz band and are associated to a normal state of consciousness.

Figure 2.4 shows an example of the power spectral density of an EEG recording obtained when a subject was asked to alternate between keeping his eyes open or closed during a few seconds. We note a clear difference in the spectra around 10 Hz for each state, due to the alpha rhythms that appear when the subject closes his eyes.

Other type of brain waves are the steady state visually evoked potentials (SSVEP) [Mor+96]. These signals are produced when a subject is visually stimulated by oscillations at certain frequency values and engenders a synchonisation of the waves produced at the visual cortex to the same frequency.

Event-related potentials. Another important pattern are event-related potentials (ERP), perturbations appearing on the EEG signal triggered by external stimuli of different types (visual, auditory, sensory, etc.). The traditional way of analysing an ERP is to take its average along several trials [Luc14]. Then, the inspection of some of its components at different latencies post-stimulus may be used to infer the state of the subject's brain. For instance, a very relevant marker is the P300 component, which is the value of the positive (hence P) peak of the ERP 300 ms after a stimulus [CB64]. If the amplitude of the P300 marker is large and positive, it means that the subject was concentrated on a target cue and was surprised by something

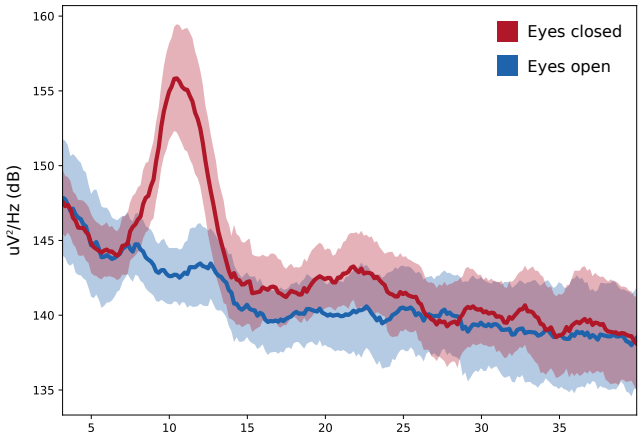

Fig. 2.4: Power spectral density at electrode `Oz` of a subject executing the experiment described in the text. The curves represent the average of the spectra along five trials for each state; the lighter areas represent the confidence interval with plus/minus one standard deviation. Data from the `ALPHA.EEG.2017-GIPSA` dataset [Cat+18].

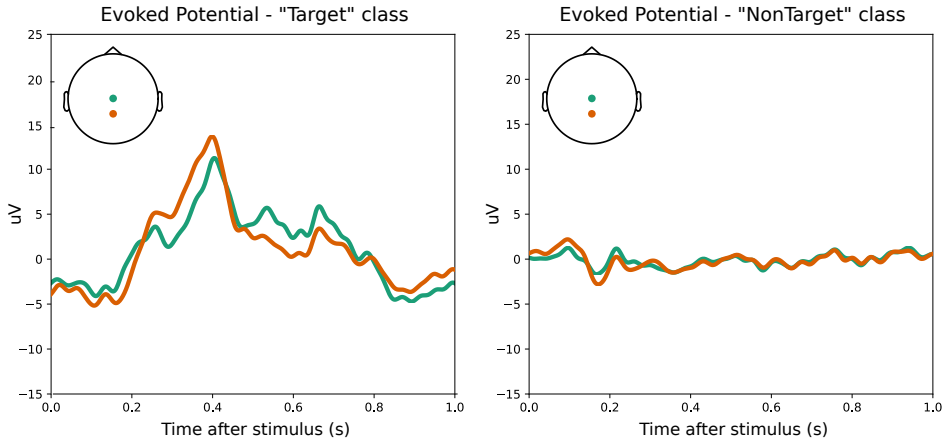

Fig. 2.5: The figures show the averaged ERPs for each condition on electrodes `Cz` and `Pz`. Note that the amplitude of the variations are quite small in both conditions. Data from the `BI.EEG.2013-GIPSA` dataset.

that changed on that cue (for instance, a visual cue that flashes). Figure 2.5 shows examples of averaged ERPs obtained in a Brain Invaders experiment [Con+11]. In this experiment, a subject is presented to a screen displaying a 6-by-6 matrix composed of pictograms of aliens. The subject is then asked to concentrate on a target alien proposed by the interface, while all the aliens in the matrix flash randomly. The EEG trials corresponding to when the target alien is flashed are labeled 'Target'. All other trials are labeled 'Non target'. Note that in the figure the P300 marker for the ERPs on the 'Target' condition are clearly higher as compared to the 'Non target' condition, as it was expected from such an experiment.

2.4.2 The Riemannian geometric framework

When working with EEG signals recorded on d electrodes, each dimension of a multivariate time series

$$\boldsymbol{x}(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_d(t) \end{bmatrix}, \tag{2.58}$$

represents the electric activity measured by one sensor. These are filtered to a frequency interval containing physiological information that is most relevant to the analysis being done. Then, if the data consists of K trials with T time samples of duration, the time series is epoched into several $d \times T$ matrices denoted by $\boldsymbol{X}_k$, where $k \in \{1, \ldots, K\}$. For a trial k starting at time sample t_k, we have

$$\boldsymbol{X}_k = \begin{bmatrix} \boldsymbol{x}(t_k) & \boldsymbol{x}(t_k + 1) & \cdots & \boldsymbol{x}(t_k + T - 1) \end{bmatrix}. \tag{2.59}$$

Parametrization. Assuming the EEG epochs are (approximately) stationary and that their statistics may be (sufficiently well) described by a Gaussian law, we can parametrize the statistics of each $\boldsymbol{X}_k$ via its cross-spectral density matrices $\boldsymbol{S}_k$. Note that although frequency f in (2.7) is a real number, in practice we have only a finite number F of frequencies in which the cross-spectral density matrix is evaluated (determined by the spectral estimation algorithm used to calculate them). Therefore, $\boldsymbol{S}_k$ may be seen as a block diagonal matrix containing F matrices with dimensions $d \times d$, where the block element associated to frequency f is denoted $\boldsymbol{S}_k(f)$,

$$\boldsymbol{S}_k = \begin{bmatrix} \boldsymbol{S}_k(1) & & 0 \\ & \ddots & \\ 0 & & \boldsymbol{S}_k(F) \end{bmatrix}. \tag{2.60}$$

To compare two epochs $\boldsymbol{X}_k$ and $\boldsymbol{X}_\ell$, we use the AIRM distance between HPD matrices (2.18) as in

$$D_S^2(\boldsymbol{X}_k, \boldsymbol{X}_\ell) = \delta_R^2(\boldsymbol{S}_k, \boldsymbol{S}_\ell) = \sum_{f=1}^{F} \delta_R^2(\boldsymbol{S}_k(f), \boldsymbol{S}_\ell(f)). \tag{2.61}$$

Extensions. Although parametrizing time series via their cross-spectral density matrices captures their full statistical information, it might not be feasible to estimate them when the number of samples in the epochs is small. This is often the case when working with EEG signals related to BCI tasks, since each trial corresponds to a cognitive task that lasts just a few seconds. In this kind of situation, it is customary to follow the approach presented in Section 2.2.3 and condense the spectral information

of trial $\boldsymbol{X}_k$ into a single parameter, its covariance matrix $\boldsymbol{C}_k$. The distance between two epochs, $\boldsymbol{X}_k$ and $\boldsymbol{X}_\ell$, is then calculated using Equation (2.18),

$$D_C^2(\boldsymbol{X}_k, \boldsymbol{X}_\ell) = \delta_R^2(\boldsymbol{C}_k, \boldsymbol{C}_\ell) \ . \tag{2.62}$$

An important downside of condensing spectral information into a single parameter is that one loses all the fine-grain information that would be relevant for discriminating two time series with, for example, the same covariace matrix but peaks of power in different frequencies. One way of keeping at least part of the spectral information accessible is to apply a bank of N band-pass filters to $\boldsymbol{x}(n)$ containing non-overlapping supports in the frequency domain. By doing so, one obtains N new time series, whose spectral information can be condensed separately to form N covariance matrices that parametrize $\boldsymbol{x}(n)$. For each epoch $\boldsymbol{X}_k$ we get N new bandpass filtered epochs $\boldsymbol{X}_k^1, \ldots, \boldsymbol{X}_k^N$, and estimate their covariance matrices $\boldsymbol{C}_k^1, \ldots, \boldsymbol{C}_k^N$. We then form a block diagonal matrix with the N covariance matrices, denoted $\boldsymbol{C}_k^{(N)}$. The distance between two epochs $\boldsymbol{X}_k$ and $\boldsymbol{X}_\ell$ is then defined as

$$D_{S_N}^2(\boldsymbol{X}_k, \boldsymbol{X}_\ell) = \delta_R^2(\boldsymbol{C}_k^{(N)}, \boldsymbol{C}_\ell^{(N)}) = \sum_{n=1}^{N} \delta_R^2(\boldsymbol{C}_k^n, \boldsymbol{C}_\ell^n) \ . \tag{2.63}$$

Event-related potentials are, by definition, non-stationary. Therefore, parametrizing them using second order statistics is not well justified by the theory presented in Section 2.2. Nevertheless, [BC14] has proposed to parametrize recordings from P300 experiments via HPD matrices by simply concatening each epoch with a prototype signal related to the P300 pattern. Mathematically, we have for each $\boldsymbol{X}_k$ an extended version given by

$$\tilde{\boldsymbol{X}}_k = \begin{bmatrix} \boldsymbol{X}_k \\ \boldsymbol{P} \end{bmatrix} \ , \tag{2.64}$$

where $\boldsymbol{P}$ is a $d \times T$ matrix obtained from averaging all epochs related to the 'Target' class from the P300 experiment. Then, the HPD matrix that describes the epoch is the covariance matrix $\tilde{\boldsymbol{C}}_k$ estimated from $\tilde{\boldsymbol{X}}_k$. Note that if d electrodes are available, the parametrization is done via $2d$-dimensional HPD matrices. The distance between two P300 epochs is then defined as

$$D_{\text{P300}}^2(\boldsymbol{X}_k, \boldsymbol{X}_\ell) = \delta_R^2(\tilde{\boldsymbol{C}}_k, \tilde{\boldsymbol{C}}_\ell) \ . \tag{2.65}$$

Classification. Note that all distances defined above use the AIRM-induced distance to compare two HPD matrices parametrizing two epochs. As such, one can directly apply the Riemannian classifiers defined in Section 2.3.3 to classify EEG signals.

2.4.3 An application: BCI classification

A Brain-Computer Interface (BCI) is a system that allows a person to interact with a machine without any physical interaction. It works by extracting features from neuro-physiological signals (e.g., the power spectral densities on certain frequency bands) and assigning them to different classes. These classes may be associated to cognitive states, sensory responses, etc., and the features are chosen so that they are discriminative for each class. We call a *paradigm* the set of cognitive tasks that a subject is asked to perform when using a BCI system; different paradigms activate different brain mechanisms and yield different signal features that may be used later as features for classification. The three most relevant paradigms in the BCI literature are:

- **Motor Imagery**: in this paradigm, a subject is asked to imagine movement, e.g. lifting his hands, feet or tongue when a visual cue is displayed on a screen. The fact of voluntary imagining such movement produces Mu-waves in the motor cortex that may then be identified by a classifier algorithm [PN01]. The laterality of the imagined movement (e.g., lifting the left-hand or right-hand) is reflected in the laterality of the production of Mu-waves, with different EEG spatial patterns being observed for each class of imagined movement. BCI systems using the motor imagery (MI) paradigm can be traced back to [Wol+91] and [Kal+96] in the 90s and are still often used in practice. Since the discriminant markers of the recorded EEG are related to oscillations in the Mu-band, most classifiers in the literature use the power spectral density of the signals in each electrode as features [Lot+18]. See [Wie+18] for a comprehensive review of classification methods used in the motor imagery paradigm.
- **SSVEP**: in this paradigm, the subject is exposed to different visual stimuli, each oscillating at a given frequency. The frequency of the SSVEP induced on the subject's brain is then used to determine which visual stimulus he was observing. As for the MI paradigm, the features of interest in an SSVEP are oscillatory-based, so classifiers are fed with information gathered from the power spectral density of the signals. See [Zhu+10] for a in-depth review of BCI systems based on the SSVEP paradigm.
- **P300**: in this paradigm, the subject is presented to a screen with multiple visual cues flashing in an apparently random manner and asked to fix his attention on one of the cues (the flashing is perceived as random by the subject, but they are controlled by the experimenters' computer). The moment the 'target' cue flashes, an ERP containing a preponderant P300 component is detected at the EEG recording. In this way, if each visual cue is associated to a semantic class (e.g. names of persons, letters of the alphabet, etc.), the P300 marker on the subject's ERPs may be used as a feature for deciding at which class the

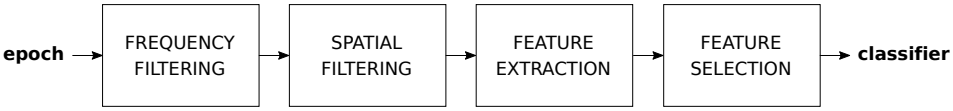

Fig. 2.6: A block diagram with the sequence of transformations applied to an input EEG epoch in a typical BCI system. Note that the order of the steps may be inverted in some cases as well as done jointly by some algorithms (e.g. CSP does spatial filtering and feature selection at the same time).

subject had his attention fixed. One of the first uses of the P300 paradigm in BCI systems was proposed in [FD88], but not much attention was given to it until the beginning of the years 2000. Nowadays, the P300 paradigm is widely used in the BCI community due to the replicability of the physiological phenomena responsible for its generation. Furthermore, the P300 paradigm demands only a certain degree of attention from the subject, whereas motor imagery usually involves training the subject and expecting a real cognitive effort from him in generating the imagined movements. See [FR+12] for a review of current trends around the P300 paradigm for BCI systems.

The standard way of operating a BCI system is to first callibrate it during an offline training phase and, then, use it online for translating EEG patterns into semantic classes. During the training phase, a classifier is optimized to discriminate between the classes of the EEG epochs in the training dataset. This step is crucial and, usually, the more data one gathers, the better are the results on the online phase (we say 'usually' because certain drifts in the statistics of the data may have a negative influence over the training phase; we will discuss further this concept in later chapters). It is also important to choose with care which signal features are to be used by the classifier and how they should be combined. For instance, in the MI paradigm, it is very common to apply a spatial filtering step called common spatial patterns (CSP [Ram+00]) before feeding the EEG signals into a classifier; this method reduces the dimensionality of the data by combining the signals from different electrodes in a way that separates their classes the most. Similarly, in the P300 paradigm, xDawn spatial filtering [Riv+09] is widely used for improving the signal-to-noise ratio of the features fed into the classifier. Traditionally, the classifiers used in BCI systems are based on linear methods, such as linear discriminant analysis, logistic regression, and support vector machines [Lot+07], and the feature vectors are composed of the power of the EEG signals on different frequency bands. Recently, there has been much interest in applying deep neural nets to the classification of EEG patterns [CG11; MG15; Din+15], but the limited number of training data points in BCI is a challenge for such systems. See [Lot+18] for a comprehensive review of classification methods in BCI. Figure 2.6 summarizes the processing steps of a BCI system.

A major difficulty in BCI systems is that very often the features used for classifying the signals do not generalize well between different subjects or even different recording sessions of the same subject. Consequently, calibration phases are usually rather long and cumbersome. However, it has been observed in practice [Con+17] that when the Riemannian geometric framework is used for describing and classifying EEG signals, the BCI system is usually more robust to changes in the statistics of the dataset. This is mostly explained by the affine-invariance of the distance between matrices in the HPD manifold, as discussed in Section 2.2.3.

The use of the Riemannian geometric framework for BCI classification has been first advocated in [Bar+12], based on previous successful applications of such approach on the classification of sleep stages in EEG [Li+12] and diffusion tensors in biomedical image processing [Pen06]. Since then, many works have been proposed using this type of classifier for BCI applications. See [Con+17] and [Yge+17] for two comprehensive reviews. Additionally, the first author of [Bar+12] has been able to demonstrate the power of the Riemannian geometric framework by winning several competitions involving EEG data (see references in [Con+17]).

2.5 Numerical illustrations

This section shows the application of the Riemannian geometric framework to two kinds of EEG data. In the first example, we consider data from BCI experiments and compare several classification pipelines using EEG epochs as input feature. The second example considers data from a sleep experiment and we show how the Riemannian geometric framework can be used for classifying epochs belonging to different sleep states. All datasets considered in this section are publicly available.

2.5.1 Example 1: BCI classification

As discussed in Section 2.4, the basic data point in a BCI experiment is an EEG epoch, which we denote $\boldsymbol{X}_k$ for the k-th experimental trial; $\boldsymbol{X}_k$ is a $d \times T$ matrix, where d is the number of electrodes and T is the number of time samples in an epoch. We consider BCI classification on two kinds of experimental paradigm: motor imagery and P300.

Motor Imagery. In this example, we compare the performance in terms of area under the ROC curve (AUC) [Bis07] of nine classification pipelines using the cross-validation scheme explained in Section 2.3.3:

(1) **epo+mdm-euc**: MDM classifier with an epoch (`epo`) $\boldsymbol{X}_k$ as input feature; Euclidean (`euc`) distance to compare data points

Tab. 2.1: Main features describing the Motor Imagery datasets used in this section.

dataset	subjects	electrodes	reference
`Weibo2014`	10	60	[Yi+14]
`Zhou2016`	4	16	[Zho+16]
`BNCI2014004`	9	6	[Lee+07]
`BNCI2014002`	14	15	[Ste+16]
`BNCI2015001`	12	13	[Fal+12]
`Alex MI`	8	16	[Bar12]

(2) **epo+knn-euc**: k-nearest neighbours (`knn`) classifier ($k = 5$) with $\boldsymbol{X}_k$ as input feature; Euclidean distance to compare data points

(3) **cov+dia+lda**: linear discriminant analysis (LDA) classifier with the diagonal (`dia`) of the covariance (`cov`) of $\boldsymbol{X}_k$ as input feature

(4) **epo+csp+lda**: reduce dimension of $\boldsymbol{X}_k$ via CSP [Ram+00] and classify with LDA

(5) **cov+knn-euc**: estimate covariance of $\boldsymbol{X}_k$ and use k-nearest neighbours classifier ($k = 5$); Euclidean distance to compare data points

(6) **cov+knn-rie**: estimate covariance of $\boldsymbol{X}_k$ and use k-nearest neighbours classifier ($k = 5$); Riemannian (`rie`) distance to compare data points

(7) **cov+mdm-euc**: estimate covariance of $\boldsymbol{X}_k$ and use MDM classifier; Euclidean distance to compare data points

(8) **cov+mdm-rie**: estimate covariance of $\boldsymbol{X}_k$ and use MDM classifier; Riemannian distance to compare data points

(9) **cov+tgs+lda**: estimate covariance of $\boldsymbol{X}_k$ and project to the tangent space (`tgs`) with reference at the geometric mean of the dataset; use LDA to classify the tangent vectors

We apply these pipelines to 6 diffferent MI datasets, all available in the `MOABB` framework [JB18]. See Table 2.2 for information on each dataset. Following the usual approach in MI paradigm, we filter every epoch in the 8-35 Hz band and use covariance matrices to parametrize their statistics. The results in the left column of Figure 2.8 elucidate several interesting facts:

- As mentioned in Section 2.2, it is not a good idea to use directly the time series epochs as features for a classifier based on distances between data points. This is reflected in the poor performance of classifiers (1) and (2).
- In general, the k-NN classifier yields a rather poor performance. This can be explained by the high-dimension of the features (d^2 dimensions when using SPD matrices and dT for epochs) and the effects of the curse of dimensionality,

which states that points in a very-high dimensional are never close to each other [Has+09].

- Using a Riemannian distance to compare SPD data points is consistently better than employing the Euclidean distance, as seen by the superior performance of pipeline (8) over pipeline (7) on all datasets.
- Classification in the tangent-space – pipeline (9) – yields the best performance on most datasets, along with another classic approach in BCI: the CSP+LDA pipeline. Note, however, that CSP applies a supervised dimensionality reduction to data points before classification. One could conjecture, then, that adding an equivalent dimensionality reduction step to pipeline (9) might increase its performance; we explore this idea in Chapter 3.

P300. We consider nine classification pipelines for data from P300 experiments. The performance is assessed via cross-validation and the scores are in terms of AUC:

(1) **erpcov+mdm-euc**: estimate extended covariance matrix of $\boldsymbol{X}_k$ with (2.64) (`erpcov`) and classify with MDM; Euclidean distance (`euc`) to compare data points

(2) **erpcov+mdm-log**: estimate extended covariance matrix of $\boldsymbol{X}_k$ and classify with MDM; log-Euclidean (`log`) distance to compare data points

(3) **erpcov+mdm-rie**: estimate extended covariance matrix of $\boldsymbol{X}_k$ and classify with MDM; Riemannian (`rie`) distance to compare data points

(4) **xdwcov+mdm-euc**: reduce dimension of epoch $\boldsymbol{X}_k$ with xDawn [Riv+09] (`xdwcov`), estimate extended covariance matrix, and classify with MDM; Euclidean distance to compare data points

(5) **xdwcov+mdm-log**: reduce dimension of epoch $\boldsymbol{X}_k$ with xDawn, estimate extended covariance matrix, and classify with MDM; log-Euclidean distance to compare data points

(6) **xdwcov+mdm-rie**: reduce dimension of epoch $\boldsymbol{X}_k$ with xDawn, estimate extended covariance matrix, and classify with MDM; Riemannian distance to compare data points

(7) **erpcov+tgs+lda**: estimate extended covariance matrix of $\boldsymbol{X}_k$, project to the tangent space (`tgs`) with reference at the geometric mean of the dataset, and classify with LDA (`lda`).

(8) **xdwcov+tgs+lda**: reduce dimension of $\boldsymbol{X}_k$ with xDawn, estimate extended covariance matrix, project to the tangent space with reference at the geometric mean of the dataset, and classify with LDA.

(9) **epo+xdw+lda**: reduce dimensionality of epoch (`epo`) $\boldsymbol{X}_k$ with xDawn and classify with LDA.

Fig. 2.7: Scores in terms of AUC for all 9 pipelines considered in the MI paradigm. The colors are intended to help discerning between the bars and the numbers on the left correspond to the index of each pipeline described in the text.

Tab. 2.2: Main features describing the P300 datasets. All experiments consisted on randomly flashing several visual cues and considering only one as 'Target' flash. The ratio of 'Target' flashes was of 1 every 6 flashes for all datasets.

dataset	subjects	electrodes	reference
`BI.EEG.2014a`	65	16	[Kor+19]
`BI.EEG.2013`	24	16	[Vai+18]
`BI.EEG.2012`	25	17	[VV+19]
`BNCI2015003`	10	8	[Gug+09]
`BNCI2014009`	10	16	[Ari+14]
`BNCI2014008`	8	8	[Ric+13]

We use six publicly available datasets (also available in the `MOABB` framework) to illustrate the performance of the pipelines. The epochs in all datasets were filtered between 1 Hz and 24 Hz and they last between 0.8 and 1.0 seconds. From the results in the right column of Figure 2.8 we can conclude that:

- Once more, using the Euclidean distance to compare between SPD data points yields inferior results as compared to both log-Euclidean and Riemannian distances.
- Reducing the dimension of epochs with the xDawn algorithm before estimating their covariance matrices yields consistently better results than not doing so, as seen by the performance of pipelines (4)-(6) as compared to pipelines (1)-(3).
- Pipelines based on tangent space classification – pipelines (7) and (8) – are again those yielding the best performance. Pipeline (9) is frequently used in the literature and has inferior results to those obtained with the Riemannian framework.

2.5.2 Example 2: Sleep-stage classification

In this example, we use data available at the Physionet database [Ter+01; Gol+00]. It contains recordings from 9 EEG electrodes of a subject sleeping for approximately 8 hours. The original sampling frequency was 512 Hz but we downsampled it to 128 Hz after band-filtering the EEG signals between 8 Hz and 35 Hz. A specialist in sleep data analysis was responsible for cutting the recordings into several 30-sec clips and then classifying them according to which sleep stage (S1, S2, S3, and REM) they belonged. We ended up with a dataset containing $K = 564$ trials of $T = 3840$ samples each. We only considered conditions S2, S3, and REM in the rest of this section, because the epochs in the S1 class have a rather high variability that is not well captured by the Riemannian geometric framework (in fact, even for the specialist it is hard to define what epochs should be in the S1 class).

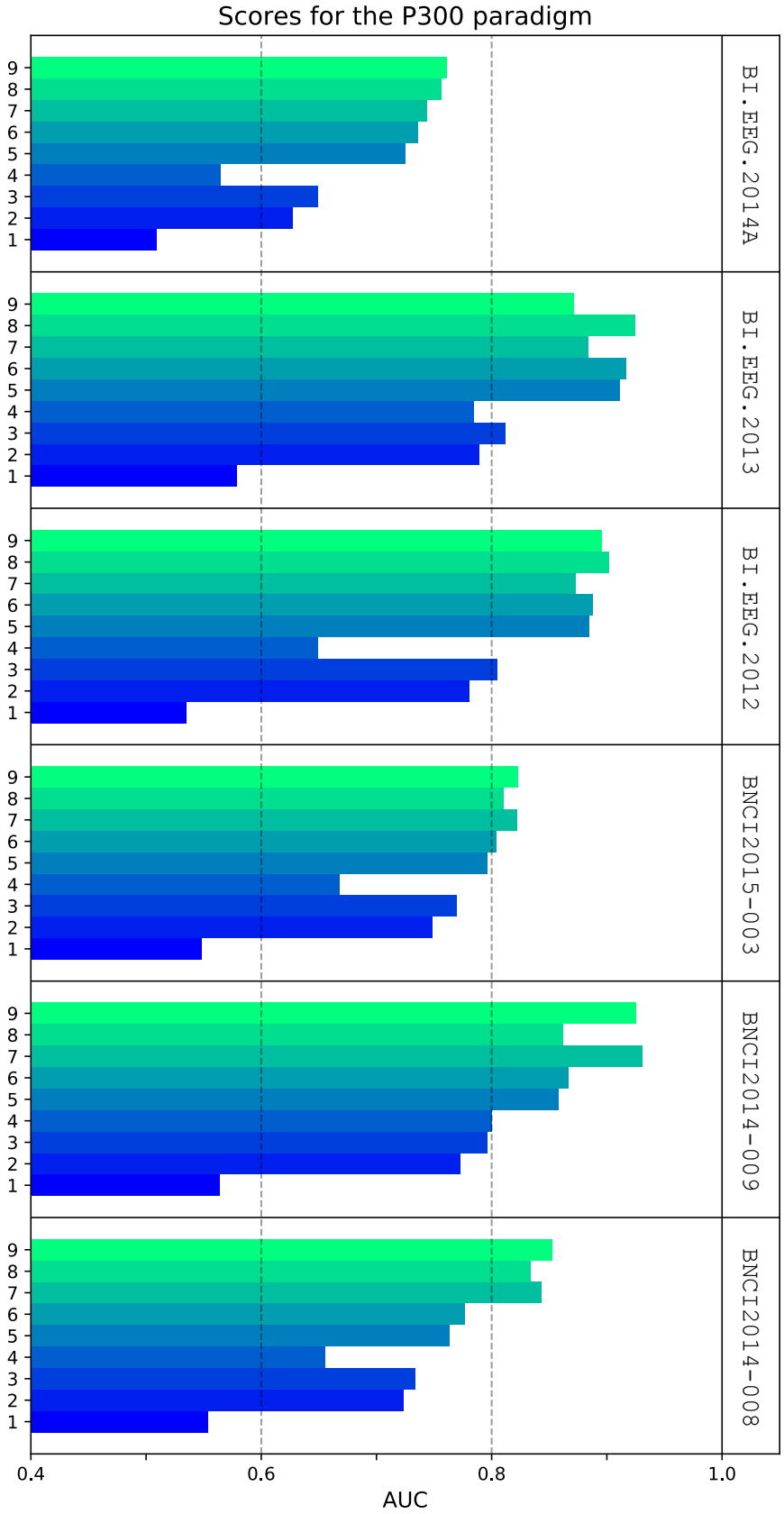

Fig. 2.8: Scores in terms of AUC for all 9 pipelines considered in the P300 paradigm. The colors are intended to help discerning between the bars and the numbers on the left correspond to the index of each pipeline described in the text.

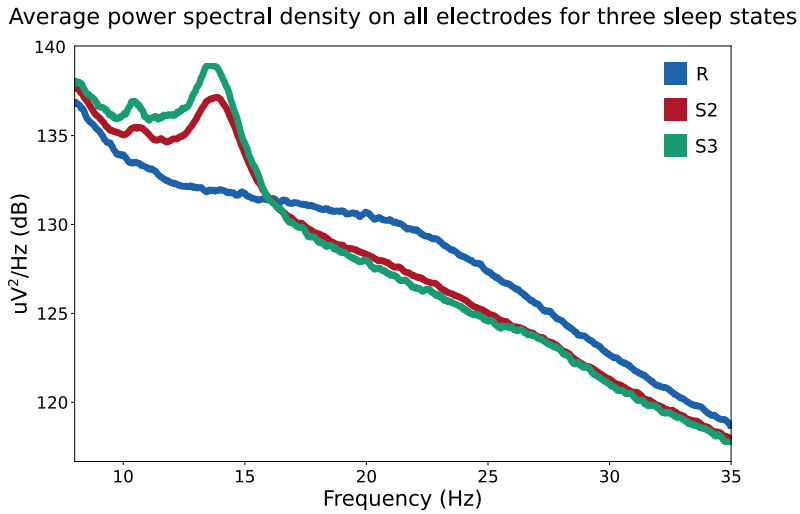

Fig. 2.9: Power spectral density estimated using Welch's method. The plotted curves are the average spectra among all epochs of each condition averaged over all nine electrodes. Remember that the epochs were filtered between 8 and 35 Hz.

Sleep states tend to have different oscillatory patterns [SC07] and it is partly based on this information that specialists are capable of assigning EEG epochs to different classes. Our first manipulation was to perform a spectral analysis of the activity recorded on all 9 electrodes, which is portrayed in Figure 2.9. We see that the REM state (labeled 'R' in the figure) has a very different spectral pattern as compared to S2 and S3, whereas the spectral content for S2 and S3 are similar but with an apparent shift in frequency (S2 is slighty to the right of S3) and distinct intensity levels.

We analysed the performance of the MDM classifier to classify EEG epochs in two cases: when the multivariate signals are parametrized via their cross-spectral density matrices and when they are parametrized simply by their covariance matrix. In each case, the distance used for comparing data points was chosen following the discussion in Section 2.4.2. Figure 2.10 shows the confusion matrices for the MDM classifier, where the parameter 'nfreqs' indicates the number of frequencies F in (2.60). We observe that:

- As expected, the classification performance is very good for epochs in the 'R' class, since its spectral content is very different to that of the two other states
- The results when epochs are parametrized via their cross-spectral density matrices are, as expected, superior to when only covariance matrices are used
- In general, larger values of 'nfreqs' lead to better classification performance, since one has more finesse in the description of the spectral content of the time series

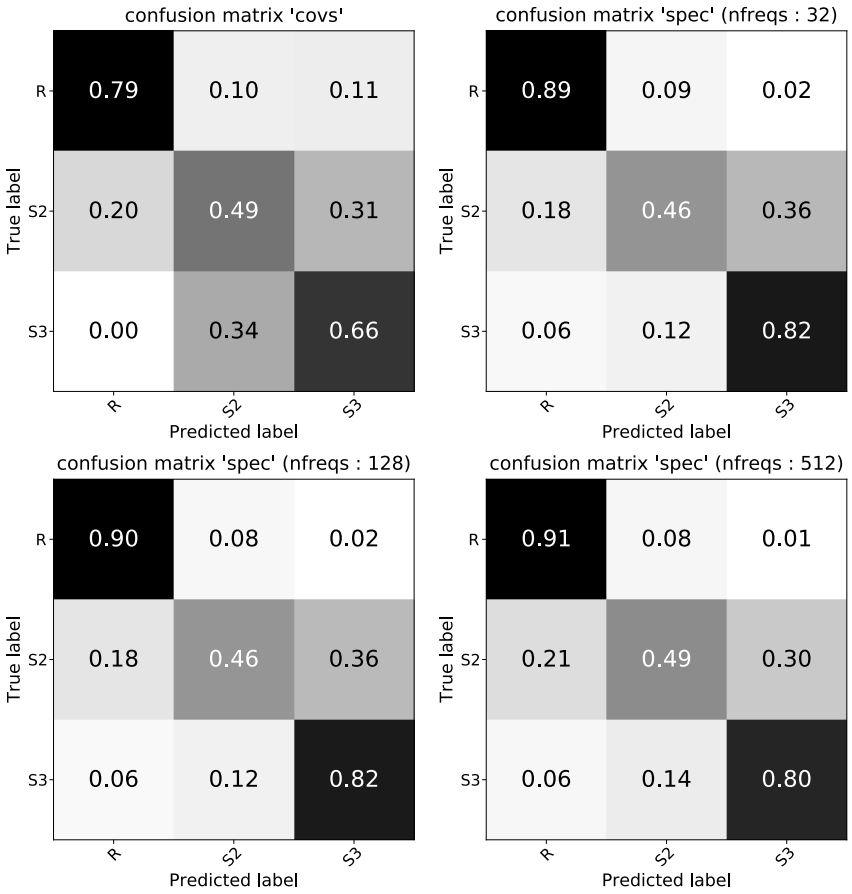

Fig. 2.10: Confusion matrices for the MDM classifier on different cases: 'covs' is when the epochs are parametrized just by their covariance matrices and 'spec' is when we use the cross-spectral density matrices. The different values of 'nfreqs' indicate how many points were used to discretize the spectra.

2.6 Conclusion

This chapter has given an overview of concepts that serve as theoretical foundation for all the contributions presented in the following chapters. Most importantly, we have motivated and presented the Riemannian geometric framework for multivariate time series, as well as illustrated its use on EEG data. A fundamental concept to retain are the effects that choosing the AIRM-induced distance in the HPD manifold have on the analysis of time series: for two time series, $\boldsymbol{x}_i(t)$ and $\boldsymbol{x}_j(t)$, band-filtered in the frequency interval $\mathcal{F}$, and whose statistics are parametrized via their cross-spectral density matrices, $\boldsymbol{S}_i(f)$ and $\boldsymbol{S}_j(f)$, we have that

$$\delta_S^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = \int_{\mathcal{F}} \delta_R^2(\boldsymbol{S}_i(f), \boldsymbol{S}_j(f)) \mathrm{d}f \ , \tag{2.66}$$

where

$$\delta_R^2(\boldsymbol{A}, \boldsymbol{B}) = \| \log(\boldsymbol{A}^{-1/2} \boldsymbol{B} \boldsymbol{A}^{-1/2}) \|_F^2 \ . \tag{2.67}$$

The affine-invariance of (2.67) makes the comparison of time series invariant to, for example, the choice of measurement scale (the distance between two time series recorded in mV or μV is the same) and also to mixing effects when they may be approximated as the action of a linear operator, such as some simplified models for the volume conduction in EEG [Con13] or some cases of crosstalk in audio signal processing [Vin+06].

In the following chapters, other invariant aspects of multivariate time series will be discussed, such as invariances related to changes in the dimension of the data (e.g., using more or less electrodes to record the same phenomenon) and in time (e.g., how to model the drift in the statistics of the data from one recording session to the next one).

3 Dimensionality reduction

Contents

3.1 Introduction **47**
3.1.1 Contributions 48
3.2 Linear methods **49**
3.2.1 Literature review 50
3.2.2 Geometry-aware methods 52
3.2.3 Numerical illustrations 54
3.3 Non-linear methods **60**
3.3.1 Literature review 60
3.3.2 Manifold learning for multivariate time series 66
3.3.3 Numerical illustrations 67
3.4 Conclusion **72**

List of notations and acronyms of the chapter

EEG	electroencephalography
BCI	brain-computer interface
HPD	Hermitian positive definite
SPD	symmetric positive definite
RG	Riemannian geometry
AIRM	affine-invariant Riemannian metric
MDM	minimum distance to mean classifier
DR	dimensionality reduction
DM	diffusion maps
PCA	principal component analysis
MI	motor imagery
ROC	receiver operating characteristic
AUC	area under the ROC curve
$\mathbb{R}^d$	set of d-dimensional real vectors
$\boldsymbol{x}$	multivariate time series
$\boldsymbol{x}^{\downarrow}$	reduced-dimension multivariate time series
$\boldsymbol{C}$	spatial covariance matrix
$\boldsymbol{C}^{\downarrow}$	reduced-dimension spatial covariance matrix
δ_E	Frobenius distance between two matrices
δ_R	AIRM-induced distance between two HPD matrices
$\mathcal{P}(d)$	manifold of d-dimensional HPD matrices
$\mathcal{O}^{d\times p}$	set of $d \times p$ orthogonal matrices
$\boldsymbol{M}^{\mathcal{X}}$	geometric mean of the HPD matrices in a set $\mathcal{X}$
K	number of data points
d	dimensionality of original data points
p	dimensionality of reduced data points

3.1 Introduction

The age of big data has changed the way of doing experimental science, making it easier and cheaper than ever to record physical phenomena simultaneously from several different sensors. Data points gathered under these circumstances live in very high-dimensional spaces and methods developed to process them are susceptible to a number of challenging problems. In fact, the issues related to high-dimensionality of data are termed the 'curse of dimensionality' [Don00] and may appear in different contexts and forms, such as:

- Approximating functions in high-dimensional spaces with grid-based methods requires a large number of samples, which quickly becomes prohibitive in practice.
- Non-parametric statistical methods based on density estimations become impractical due to the excessive number of samples that they require.
- Norms in $\mathbb{R}^d$, with d very large, are not numerically equivalent and so the same function may have different degrees of smoothness under different norms.
- Algorithms become very slow for processing the data.

From these observations, one might have the tendency to say that, in fact, having more dimensions to describe a physical phenomena might not be such a good idea after all. However, in practice, the dimensions of high-dimensional data points are not completely independent between each other. One clear example is the case of electroencephalographic (EEG) recordings, where sensors located at close positions of a subject's scalp tend to record time series which are very correlated to one another. These correlations imply that, in fact, a data point living in a high-dimensional space does not have necessarily as many degrees of freedom as available sensors. In other words, the true dimensionality of the data point is often much smaller than that of its ambient space.

Assuming that the samples of a dataset have an intrinsic low dimensionality has inspired many works in different research communities. The main goal in these approaches is to determine a transformation of the data points in such a way that their new representation is a more compact description of the phenomena that they describe. This is usually called *dimensionality reduction* (DR). In information theory, DR is related to compression and coding problems. In statistics, it is also called latent variable analysis. In the field of pattern recognition, it is known as feature extraction. Put in precise terms, DR is the problem of determining a mapping of points from a set where the data points were originally described, into a new set where their description is simpler. The difference in the proposals found in the literature resides in what information one is ready to lose [CG15]. Also, different methods impose

different constraints on the structure of the transformation. For instance, whether it is parametric or not, if it is a linear transformation or not, etc.

In this chapter, we discuss dimensionality reduction in the context of multivariate time series. We base our investigations on the assumption that time series recorded from several sensors tend to present correlations between them. Consequently, one may search for transformations that shall discard redundancies from the data points and yield more compact representations of the phenomena under study. Related to this assumption is the one of expecting commonalities between multivariate time series that describe the same physical phenomenon, but use different numbers and/or placement of sensors. Such invariance to dimensionality is a reasonable one in practice and implies the existence of a canonical representation of the time series to be determined from the data. We manipulate the time series using the Riemannian geometric (RG) framework described in Chapter 2, that is, for a set of K zero-mean d-dimensional multivariate time series, $\mathcal{X}' = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K\}$, where d is the number of sensors, we have a set of symmetric positive definite (SPD) matrices with dimensions $d \times d$ that describes the statistics of the time series, with

$$\mathcal{X} = \{\boldsymbol{C}_1, \ldots, \boldsymbol{C}_K\} \,, \tag{3.1}$$

where $\boldsymbol{C}_i = \mathbb{E}[\boldsymbol{x}_i(t)\boldsymbol{x}_i(t)^T]$. SPD matrices live in a Riemannian manifold denoted by $\mathcal{P}(d)$ and whose dimensionality is $d \times (d+1)/2$ (that is, the number of degrees of freedom in a $d \times d$ symmetric matrix). Our goal, then, is to determine a transformation $\Phi : \mathcal{P}(d) \to \Omega$, where Ω is some space of dimensionality p with $p < d \times (d+1)/2$.

In what follows, we divide our discussion in two parts. Firstly, we consider linear DR methods, where $\Omega = \mathcal{P}(p)$ and transformation Φ is parametrized by a matrix. We present classical approaches for DR in Euclidean space and show how they may be adapted to take into account the intrinsic geometry of the manifold where the data points are defined. We apply these methods to data from brain computer interface (BCI) recordings and show that one may reduce the dimensionality of the time series without decreasing, in average, the performance of a classifier trained to classify the epochs. In the second part, we consider non-linear DR methods, where the data points defined in $\mathcal{P}(d)$ are mapped into an Euclidean space. We focus our discussion on the *diffusion maps* (DM) method and show how it may be applied to the analysis of datasets containing multivariate time series as samples. As in Chapter 2, we illustrate the methods on data from BCI experiments, but also consider EEG data from sleep recordings and an EEG experiment on the resting-state condition.

3.1.1 Contributions

The content of this chapter is based on (and extends) the works presented in two published papers:

P. L. C. Rodrigues, M. Congedo, and C. Jutten, “Multivariate time-series analysis via manifold learning", 2018 IEEE Statistical Signal Processing Workshop (SSP), Freiburg, Germany, Jun. 2018.

and

P. L. C. Rodrigues, M. Congedo, and C. Jutten, “Dimensionality reduction for BCI classification using Riemannian geometry", BCI 2017 - 7th Graz Brain Computer Interface Conference, Graz, Austria, Sep. 2017.

It is worth mentioning that the scope of our contributions in this chapter is rather limited as compared to the following chapters of this thesis. Nevertheless, DR is a very relevant problem in practice and deserves to be put in perspective. Our contributions have been to apply linear and non-linear dimensionality reduction methods to the study of multivariate time series via their parametrization with SPD matrices. In the case of linear DR, we have applied the method proposed in [Har+17], which is a general method for reducing the dimensionality of SPD matrices, to a context where these matrices describe the statistics of multivariate time series. As for the non-linear DR, we have defined an adequate notion of similarity between multivariate time series based on the geodesic distance between the parameters that describe their statistics. As a result, we can build a kernel matrix for a manifold learning procedure applied to a set of multivariate time series which reflects well their intrinsic geometry. Python code implementing part of the examples presented in the chapter is available at:

`https://github.com/plcrodrigues/PhD-Code`

3.2 Linear methods

In this section, we consider transformations Φ parametrized by a linear transformation $\boldsymbol{W} \in \mathbb{R}^{d \times p}$, where $p < d$. In this context, for a zero-mean multivariate time series $\boldsymbol{x}(t) \in \mathbb{R}^d$ parametrized by a SPD matrix $\boldsymbol{C} \in \mathcal{P}(d)$, the transformation is given by

$$\boldsymbol{C}^{\downarrow} = \Phi_{\boldsymbol{W}}(\boldsymbol{C}) = \boldsymbol{W}^T \boldsymbol{C} \boldsymbol{W} \ , \tag{3.2}$$

where $\boldsymbol{W}^T \boldsymbol{W} = \boldsymbol{I}_p$ and $\boldsymbol{C}^{\downarrow}$ parametrizes the statistics of a zero-mean multivariate time series $\boldsymbol{x}^{\downarrow}(t) \in \mathbb{R}^p$ given by

$$\boldsymbol{x}^{\downarrow}(t) = \boldsymbol{W}^T \boldsymbol{x}(t) \ . \tag{3.3}$$

Matrix $\boldsymbol{W}$ is an orthogonal matrix, so that (3.3) may be interpreted as a projection from $\mathbb{R}^d$ onto $\mathbb{R}^p$.

In what follows, we present two important methods for linear dimensionality reduction in Euclidean space and extend them to multivariate time series. Then, we make the transition to geometry-aware DR methods for SPD matrices and provide details related to their implementation and optimization procedures. We close the section with numerical illustrations on real electroencephalographic (EEG) data from brain-computer interface (BCI) experiments. Our main result is that linear DR can be used to reduce the dimensions of multivariate time series without affecting (in average) the performance of statistical classifiers. This is relevant because it allows reducing the complexity of a set of data points (and, therefore, reduce the computational cost for all succeeding calculations involving them) without significantly losing discriminative power to classify the dataset.

3.2.1 Literature review

The usual setting for linear dimensionality reduction is one where the data points are Euclidean vectors and each of its dimensions represent a different aspect (or feature) of a given measurement. Because linear DR is a rather well known topic in statistics and machine learning, we will assume that the reader is already familiar with the classical setting and present the concepts considering that the data points are d-dimensional zero-mean multivariate time series.

Principal component analysis. Probably the linear DR technique most commonly used in statistical data analysis is the one proposed by Pearson [Pea01] and known as *principal component analysis* (PCA). It is based on the idea of minimizing the sum of squared residual errors between the projected data points and their original counterparts. For a dataset consisting of K multivariate time series, as in $\mathcal{X}' = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K\}$, we have that

$$\boldsymbol{W}_{\text{PCA}} = \underset{\boldsymbol{Q} \in \mathcal{O}^{d \times p}}{\operatorname{argmin}} \sum_{k=1}^{K} \mathbb{E}\left[\| \boldsymbol{x}_k(t) - \boldsymbol{Q}\boldsymbol{Q}^T \boldsymbol{x}_k(t) \|_2^2 \right] , \tag{3.4}$$

where $\mathcal{O}^{d \times p}$ is the set of $d \times p$ orthogonal matrices (also known as the Stiefel manifold). Matrix $\boldsymbol{W}_{\text{PCA}}$ can be determined analitically, as shown in several references in the literature [Has+09]. We have that,

$$\boldsymbol{W}_{\text{PCA}} = \begin{bmatrix} \boldsymbol{w}_1 & \cdots & \boldsymbol{w}_p \end{bmatrix} , \tag{3.5}$$

with $\boldsymbol{w}_k \in \mathbb{R}^d$ obtained from the eigenvector relation,

$$\left(\frac{1}{K} \sum_{k=1}^{K} \boldsymbol{C}_k \right) \boldsymbol{w}_k = \lambda_k \boldsymbol{w}_k , \tag{3.6}$$

where $\lambda_1 \geq \cdots \geq \lambda_p > 0$. The $\{\boldsymbol{w}_k\}_{1\leq k\leq p}$ form an orthonormal set of vectors, and the $\{\boldsymbol{C}_k\}_{1\leq k\leq K}$ are the spatial covariance matrices of the multivariate time series in $\mathcal{X}$. Note that this form of PCA uses the arithmetic mean of a set of SPD matrices, a quantity that is less adapted to the intrinsic geometry of the SPD manifold as compared to the Fréchet mean, as discussed in Chapter 2 and recalled later in the text (also known as the center of mass, or geometric mean, of a set of SPD matrices). Gathering the data points into a matrix $\boldsymbol{X}(t) = [\boldsymbol{x}_1(t), \ldots, \boldsymbol{x}_K(t)] \in \mathbb{R}^{d\times K}$, we may rewrite (3.4) as

$$\begin{aligned}
\boldsymbol{W}_{\text{PCA}} &= \underset{\boldsymbol{Q}\in\mathcal{O}^{d\times p}}{\operatorname{argmin}}\, \mathbb{E}\left[\|\boldsymbol{X}(t) - \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{X}(t)\|_F^2\right] , && (3.7)\\
&= \underset{\boldsymbol{Q}\in\mathcal{O}^{d\times p}}{\operatorname{argmin}}\, \mathbb{E}\left[\operatorname{tr}\Big((\boldsymbol{X}(t) - \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{X}(t))^T(\boldsymbol{X} - \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{X}(t))\Big)\right] , && (3.8)\\
&= \underset{\boldsymbol{Q}\in\mathcal{O}^{d\times p}}{\operatorname{argmin}}\, \mathbb{E}\left[\operatorname{tr}\Big(\boldsymbol{X}(t)^T\boldsymbol{X}(t) - \boldsymbol{X}(t)^T\boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{X}\Big)\right] , && (3.9)\\
&= \underset{\boldsymbol{Q}\in\mathcal{O}^{d\times p}}{\operatorname{argmax}}\, \mathbb{E}\left[\operatorname{tr}\Big(\boldsymbol{X}(t)^T\boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{X}(t)\Big)\right] , && (3.10)\\
&= \underset{\boldsymbol{Q}\in\mathcal{O}^{d\times p}}{\operatorname{argmax}} \sum_{k=1}^{K} \mathbb{E}\left[(\boldsymbol{Q}^T\boldsymbol{x}_k(t))^T(\boldsymbol{Q}^T\boldsymbol{x}_k(t))\right] , && (3.11)\\
&= \underset{\boldsymbol{Q}\in\mathcal{O}^{d\times p}}{\operatorname{argmax}} \sum_{k=1}^{K} \mathbb{E}\left[\|\boldsymbol{Q}^T\boldsymbol{x}_k(t)\|^2\right] , && (3.12)
\end{aligned}$$

and so $\boldsymbol{W}_{\text{PCA}}$ can also be interpreted as the orthogonal matrix that maximizes the total dispersion (or variance) of the set of transformed data points (remember that they have zero mean). In fact, PCA's variance-maximization property is sometimes indicated as the main motivation for defining it, although Pearson's original idea was to minimize the reconstruction error. We will use this interpretation later to define a geometry-aware method for reducing the dimensions of SPD matrices.

Principal component analysis is a standard tool in statistical data analysis and has found applications in several contexts, such as sound compression [CH91], image processing and classification [TP91], and bioinformatics [Rei+08]. Its main force is the fact of having an analytical form and being parametric, meaning that it can be directly extended to new data points. A natural limitation is that it defines a linear transformation on the data points, so it is only optimal when the elements in $\mathcal{X}$ live in some low-dimensional hyperplane; when this is not the case, non-linear methods may yield more adequate transformations (we will discuss them in Section 3.3). Furthermore, PCA proposes the same linear transformation for all data points in $\mathcal{X}$, meaning that it has no flexibility to take into account local information. There have been many proposals for extending PCA to cases where the dataset does not live

on a hyperplane but can be approximated as a 'collage' of several hyperplanes of different dimensionalities; see [WH00] for a survey of such procedures.

Supervised dimensionality reduction. When the data points in $\mathcal{X}$ are accompanied by labels determining to which class they belong to, dimensionality reduction methods may be adapted to take such information into account and yield transformations that enhance the separability of the classes in the projected space. The most popular method for supervised DR is probably Fisher's *linear discriminant analysis* (LDA), which is based on the idea of searching for a projector matrix that maximizes the between-class variability of the projected data points while minimizing their within-class variability. A notable adaptation of such technique to the context of multivariate time series is known as *common spatial patterns* (CSP) [Kol+90] and is a fundamental preprocessing technique in brain-computer interfaces [Lot14]. Recently, [Yge+15] has performed an empirical study comparing the performance of classical CSP to a modified version using geometry-aware manipulations of the data points. Their results demonstrated superior classification performance when the geometry of the SPD manifold was taken into account; the exception was when the SPD matrices were too big and Riemannian geometric methods started to perform poorly due to numerical instabilities.

3.2.2 Geometry-aware methods

Recently, part of the computer vision community has been interested in developing geometry-aware dimensionality reduction techniques for data points defined in the SPD manifold [Har+14; Har+17; Har+18]. In such works, the SPD matrices describe the statistics of the pixels in patches of images [Tuz+06] or patches of videos [Tuz+08], and typically have dimensionalities in the order of 100×100 [Har+18]. An important work in this context is the one described in [Har+18], where the authors propose both an unsupervised and a supervised method for linear dimensionality reduction in the SPD manifold. In what follows, we give a brief description of the unsupervised method considered in [Har+18] and use it in a context where the SPD matrices are actually spatial covariance matrices describing the statistics of multivariate time series. We refer the interested reader also to [Hor+16] and [Dav+17], which have adapted the works in [Har+18] to when the data points describe EEG epochs in brain-computer interfaces.

Ref. [Har+18] proposes a generalization of the principle of variance-maximization from classical PCA to data points defined in the SPD manifold,

$$\mathcal{X} = \{\boldsymbol{C}_1, \ldots, \boldsymbol{C}_K\} \subset \mathcal{P}(d) \,, \tag{3.13}$$

with the projector matrix defined as

$$\boldsymbol{W}_{\text{rie-PCA}} = \underset{\boldsymbol{Q}\in\mathcal{O}^{d\times p}}{\operatorname{argmax}} \sum_{k=1}^{K} \delta_R^2(\boldsymbol{Q}^T\boldsymbol{C}_k\boldsymbol{Q}, \boldsymbol{Q}^T\boldsymbol{M}^{\mathcal{X}}\boldsymbol{Q}) \;, \tag{3.14}$$

where

$$\delta_R^2(\boldsymbol{A}, \boldsymbol{B}) = \left\| \log\left(\boldsymbol{A}^{-1/2}\boldsymbol{B}\boldsymbol{A}^{-1/2}\right)\right\|_F^2 \tag{3.15}$$

is the geodesic distance between two SPD matrices, $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{P}(d)$, and

$$\boldsymbol{M}^{\mathcal{X}} = \underset{\boldsymbol{M}\in\mathcal{P}(d)}{\operatorname{argmax}} \sum_{k=1}^{K} \delta_R^2(\boldsymbol{C}_k, \boldsymbol{M}) \tag{3.16}$$

is the Fréchet mean of the data points in $\mathcal{X}$. In contrast to classical PCA (which we shall call 'euc-PCA' from now on), we are not aware of any analytical solution for determining $\boldsymbol{W}_{\text{rie-PCA}}$ directly from the matrices in $\mathcal{X}$. Consequently, we resort to numerical procedures adapted to optimization problems defined in matrix manifolds. Such methods generalize classical optimization algorithms, such as steepest descent and conjugate gradient, to a setting where the variables are defined in spaces with some special structure, such as orthogonality constraints or low-rankness. We do not intend to give a review of optimization on manifolds and refer the reader to [Abs+09] for a wide presentation of the field. Nevertheless, we present some of the details provided in [Har+18] on how to solve the optimization problem in (3.14) using a conjugate-gradient method on the Stiefel manifold. Firstly, it derives an expression for the Jacobian of the cost function $f_{\text{rie-PCA}}(\cdot)$ in (3.14), with

$$D_{\boldsymbol{Q}} f_{\text{rie-PCA}}(\boldsymbol{Q}) = \sum_{k=1}^{K} D_{\boldsymbol{Q}} f_k(\boldsymbol{Q}) \;, \tag{3.17}$$

where

$$f_k(\boldsymbol{Q}) = \delta_R^2(\boldsymbol{Q}^T\boldsymbol{C}_k\boldsymbol{Q}, \boldsymbol{Q}^T\boldsymbol{M}^{\mathcal{X}}\boldsymbol{Q}) \;, \tag{3.18}$$

and

$$D_{\boldsymbol{Q}} f_k(\boldsymbol{Q}) = 4\Big(\boldsymbol{C}_k\boldsymbol{Q}(\boldsymbol{Q}^T\boldsymbol{C}_k\boldsymbol{Q})^{-1} - \boldsymbol{M}^{\mathcal{X}}\boldsymbol{Q}(\boldsymbol{Q}^T\boldsymbol{M}^{\mathcal{X}}\boldsymbol{Q})^{-1}\Big) \log\left(\boldsymbol{Q}^T\boldsymbol{C}_k\boldsymbol{Q}(\boldsymbol{Q}^T\boldsymbol{M}^{\mathcal{X}}\boldsymbol{Q})\right) . \tag{3.19}$$

Then, it proposes an iterative optimization procedure where, for each iteration, we:

(1) Compute the gradient $\nabla_{\boldsymbol{Q}} f_{\text{rie-PCA}}(\boldsymbol{Q})$ of the objective function $f_{\text{rie-PCA}}(\boldsymbol{Q})$ on the Stiefel manifold at the current solution $\boldsymbol{Q}_n$ using

$$\nabla_{\boldsymbol{Q}_n} f_{\text{rie-PCA}}(\boldsymbol{Q}_n) = D_{\boldsymbol{Q}} f_{\text{rie-PCA}}(\boldsymbol{Q}_n) - \boldsymbol{Q}_n \, D_{\boldsymbol{Q}} f_{\text{rie-PCA}}(\boldsymbol{Q}_n) \, \boldsymbol{Q}_n^T \;. \tag{3.20}$$

(2) Determine a search direction $\boldsymbol{H}_n$ by parallel transporting (see Chapter 2 for a definition) the previous search direction (from iteration $n-1$) and combining it with $\nabla_{\boldsymbol{Q}_n} f_{\text{rie-PCA}}(\boldsymbol{Q}_n)$.

(3) Perform a line search along the geodesic at $\boldsymbol{Q}_n$ in the direction of $\boldsymbol{H}_n$ and determine $\boldsymbol{Q}_{n+1}$.

These steps are repeated until convergence to a local minimum, or until a maximum number of iterations is reached. It is worth mentioning that optimization on manifolds is a rather mature topic in applied mathematics, partly due to the availability of high-quality code implementing the algorithms developed by the research community, such as the Python package `pymanopt` [Tow+16], which we use in all manifold optimization procedures in this thesis.

Note that the cost function in (3.14) does not correspond exactly to the maximization of the dispersion of the projected data points around their geometric mean, since

$$\boldsymbol{Q}^T \boldsymbol{M}^{\mathcal{X}} \boldsymbol{Q} \neq \underset{\boldsymbol{M} \in \mathcal{P}(d)}{\operatorname{argmax}} \sum_{k=1}^{K} \delta_R^2(\boldsymbol{Q}^T \boldsymbol{C}_k \boldsymbol{Q}, \boldsymbol{M}) \; . \tag{3.21}$$

However, if we first re-center the points in $\mathcal{X}$ so that their geometic mean is the identity, as in

$$\boldsymbol{C}_k \to (\boldsymbol{M}^{\mathcal{X}})^{-1/2} \boldsymbol{C}_k (\boldsymbol{M}^{\mathcal{X}})^{-1/2} \; , \tag{3.22}$$

we obtain a new expression

$$\boldsymbol{W}_{\text{rie-PCA}} = \underset{\boldsymbol{Q} \in \mathcal{O}^{d \times p}}{\operatorname{argmax}} \sum_{k=1}^{K} \delta_R^2 \left(\boldsymbol{Q}^T \left((\boldsymbol{M}^{\mathcal{X}})^{-1/2} \boldsymbol{C}_k (\boldsymbol{M}^{\mathcal{X}})^{-1/2} \right) \boldsymbol{Q}, \boldsymbol{I}_d \right) \; , \tag{3.23}$$

for which the cost function does maximize the dispersion of the projected points around their geometric mean. We use this second formulation for all numerical illustrations presented in Section 3.2.3.

3.2.3 Numerical illustrations

We investigate whether the geometry-aware linear DR technique presented above is a good option for when the data points are multivariate time series parametrized via SPD matrices. For this, we use the performance of a statistical classifier as proxy for the adequacy of the DR procedure: if the classification score decreases after the dimensionality reduction step, then it means that discriminatory information was lost and the dimensionality reduction was too severe or not well executed. We consider signals recorded from EEG-based brain computer interfaces (BCI), where the goal is to determine which cognitive task a test subject executed on each experimental trial. We use the Riemannian geometric framework presented in Chapter 2 to parametrize the statistics of the EEG epochs via symmetric positive definite (SPD) matrices.

The BCI data is in the form of d-dimensional multivariate time-series, where each dimension represents an electrode. Each experimental trial i lasts a few seconds and is associated to a matrix $\boldsymbol{X}_i \in \mathbb{R}^{d \times T}$, where T is the number of time samples defining the trial. To every trial we associate a SPD matrix $\boldsymbol{C}_i$ describing its multivariate statistics and a label ℓ_i indicating what was the task performed during the trial. The dataset for each subject is composed of a set of couples $(\boldsymbol{C}_i, \ell_i)$.

Classification pipelines. We classify unlabeled SPD data points via the minimum distance to mean (MDM) algorithm. It determines the geometric mean of the covariance matrices in each class of the *training* set and then assigns to each matrix in the *test* set the class to which the distance to the mean is the smallest [Bar+12]. We compare five different pipelines for classification:

- **MDM**: No dimensionality reduction (DR) and classification using the MDM algorithm.
- **euc-PCA+MDM**: DR using $\boldsymbol{W}_{\text{euc-PCA}}$ to reduce the dimensionality of the time series. Classification using MDM.
- **rie-PCA+MDM**: DR using $\boldsymbol{W}_{\text{rie-PCA}}$ to reduce the dimensionality of the time series. Classification using MDM.
- **SELg+MDM**: DR by choosing the electrodes the closest to the active regions during cognitive tasks related to the BCI paradigm being considered (SELg: *good selection*). Classification using MDM.
- **SELb+MDM**: DR by choosing electrodes placed in regions that do not give much discriminatory information for the BCI paradigm under consideration (SELb: *bad selection*). Classification using MDM.

The performance of each pipeline is assessed via a 10-fold cross-validation procedure and compared by their AUC (area under the receiver operating characteristic curve).

Dataset. We carried out our analysis on the `Physionet` database, a publicly available database with recordings from motor imagery (MI) experiments [Gol+00] on 64 EEG electrodes from 109 subjects. We only used the data from tasks of imagined hands and feet movement, which corresponds to approximately 44 trials per subject (22 for each class). We filtered the EEG signals in the 8-30 Hz band and considered each trial as a segment from 0.5 to 2.5s after each trial onset. We estimated the spatial covariance matrices using Ledoit-Wolf regularization [LW04].

The selection of electrodes for the **SELg+MDM** and **SELb+MDM** pipelines took into account the fact that the motor cortex is the most active region during BCI experiments under the MI paradigm. As such, we have chosen the 12 electrodes the closest to the motor cortex for **SELg+MDM**, which are the electrodes {`F3`, `Fz`, `F4`, `FC1`, `FC2`, `C3`, `Cz`, `C4`, `CP1`, `CP2`, `P3`, `P4`}. They form a symmetric region and cover the

Fig. 3.1: Position of the 12 electrodes selected for the **SELb+MDM** pipeline (in red) and the **SELg+MDM** pipeline (in green).

motor cortex. For the **SELb+MDM** pipeline we have chosen electrodes far from the motor cortex and that do not form a symmetric region: {`FPz`, `FP1`, `AFz`, `AF3`, `AF7`, `F7`, `F5`, `F3`, `F1`, `FT7`, `FC5`, `FC3`}. Figure 3.1 provides a visual depiction of the position of the EEG electrodes selected in each case.

Results and discussion. We have compared the performance of all classification pipelines with a DR step (**euc-PCA+MDM**, **rie-PCA+MDM**, **SELb+MDM** and **SELg+MDM**) to that of the **MDM** pipeline. Figure 3.2 shows the results when comparing pipelines **euc-PCA+MDM** and **rie-PCA+MDM** to **MDM**, for different values of p, the dimension of the reduced SPD matrices. Figure 3.3 shows the comparisons of all pipelines to **MDM** when $p = 12$. In both figures, each dot corresponds to one subject of the dataset and the axis indicate the AUC scores of each pipeline.

Due to the large number of subjects available in the database, we were able to perform statistical tests for assessing whether there was one pipeline that gave better (or worse) results than **MDM** in average. For this, we have estimated a linear regression model with zero intercept for each pair of pipelines in Figure 3.2 and Figure 3.3. If one of the methods is, in average, superior to the other, the regression model should have an angular coefficient ($\hat{m}$) different than one; if the method on the y-axis is superior, $\hat{m}$ is greater than one, if not, it is smaller. If we can not reject the null hypothesis of $\hat{m}$ being equal to one, then the two pipelines are said to be equivalent. We have performed F-tests for each pair of methods and fixed the threshold for the rejection of the null hypothesis to 1%. The results obtained after this analysis are available in Table 3.1.

From these results, we conclude that when reducing 'too much' the dimensionality of the data, e.g. with $p = 6$ or $p = 8$, the performance of the classification pipelines are poorer than using all 64 available electrodes, as revealed by the values of $\hat{m}$ smaller than one. This is mostly because with such reduction we may be discarding

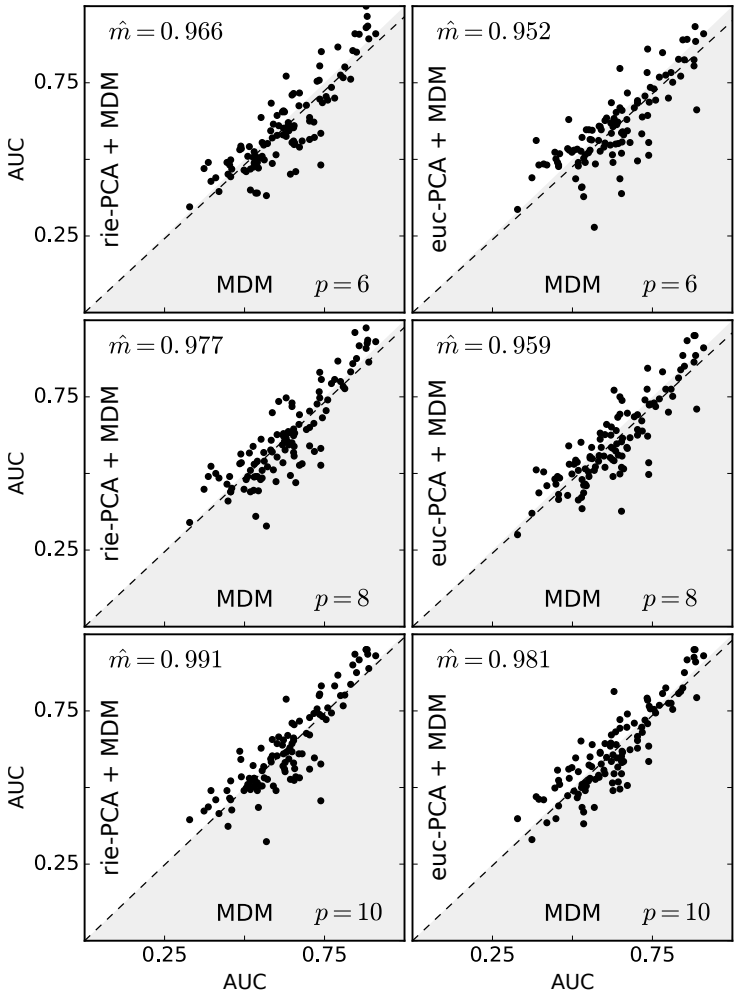

Fig. 3.2: Comparison of the classification scores of pipelines with a DR step versus the **MDM** pipeline. We have considered multiple values for the dimension p of the reduced covariance matrices. Each dot in the graph corresponds to the classification scores of one subject of the database. The parameter $\hat{m}$ is the angular coefficient of a linear regression model applied to the scattered points. It indicates whether a classification pipeline with DR has the same performance as compared to pipeline **MDM** ($\hat{m} = 1$), or if it is superior ($\hat{m} > 1$), or inferior ($\hat{m} < 1$).

Tab. 3.1: Results of the statistical tests for each comparison between classification pipelines. The parameter $\hat{m}$ indicates whether a classification pipeline with DR has the same performance as compared to a pipeline without DR ($\hat{m} = 1$), if it is superior ($\hat{m} > 1$), or inferior ($\hat{m} < 1$). The nulll hypothesis being tested is that of whether $\hat{m}$ is statistically significantly different than one. The threshold for rejecting the null hypothesis was fixed to 1% and the p-values were corrected for the multiple comparisons problem via the Bonferroni method.

	p	$\hat{m}$	rej. H0?
rie-PCA + MDM	6	0.966	OUI
	8	0.977	OUI
	10	0.991	NON
	12	0.997	NON
SELg+MDM	12	1.046	OUI

	p	$\hat{m}$	rej. H0?
euc-PCA + MDM	6	0.952	OUI
	8	0.959	OUI
	10	0.981	OUI
	12	0.99	NON
SELb+MDM	12	0.822	OUI

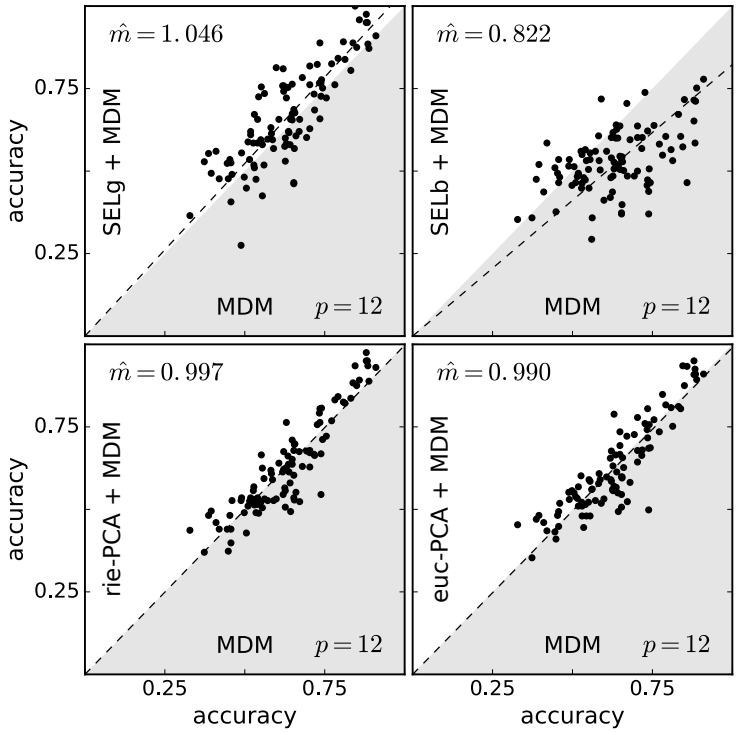

Fig. 3.3: Comparison of the classification scores of pipelines with a DR step versus the **MDM** pipeline. We have fixed $p = 12$ for the dimension of the reduced covariance matrices. Each dot in the graph corresponds to the classification scores of one subject of the database. The parameter $\hat{m}$ is the angular coefficient of a linear regression model applied to the scattered points. It indicates whether a classification pipeline with DR has the same performance as compared to pipeline **MDM** ($\hat{m} = 1$), or if it is superior ($\hat{m} > 1$), or inferior ($\hat{m} < 1$).

too much content from the data, to the point where some important discriminatory information might be lost. For $p = 10$, the **rie-PCA+MDM** pipeline already has equivalent performance to **MDM**, whereas **euc-PCA+MDM** is still inferior. Such result points out to the fact that by using the intrinsic geometry of the data points, one might loose less information when reducing the dimensionality of the SPD data points as compared to the Euclidean approach. This goes in line with the findings in [Hor+16], which demonstrated superior results for geometry-aware unsupervised DR techniques as compared to the their Euclidean counterparts. For $p = 12$, **rie-PCA+MDM** and **euc-PCA+MDM** become equivalent to **MDM**, indicating that there is a threshold value for p from which the DR techniques keep enough information for assuring the same performance as for **MDM**. Note that although such regimes are expected in any set of SPD data points, the exact values of p determining their transitions is most likely dependent on the characteristics of each database.

The results with **SELb+MDM** and **SELg+MDM** indicate what may happen in two opposing situations where the dimensionality of SPD matrices are reduced. In the one hand, **SELb+MDM** selects only electrodes that do not provide much information for the classification of the EEG epochs; consequently, the performance with such pipeline is inferior to **MDM**. On the other hand, **SELg+MDM** attains results that are superior to **MDM**, as indicated by $\hat{m} > 1$ in Table 3.1. This may be explained by the fact that the electrodes chosen in the DR step are only those that provide the maximum of physiological information related to the tasks in the BCI experiment, whereas in **MDM** (where all 64 electrodes are kept) there is room for features that are only linked to noise or are just not discriminative at all. It is worth mentioning that pipelines **rie-PCA+MDM** and **euc-PCA+MDM** provide an automatic procedure for reducing the dimensions of the SPD matrices, whereas **SELg+MDM** relies on a priori information regarding the intricacies of the experiment that generates the signals in the database.

Conclusions. We have demonstrated the possibility of reducing the dimensionality of the covariance matrices that describe the statistics of EEG signals, while maintaining good classification performance (i.e. the same as for when we use the full SPD matrices). Moreover, we have shown that by carefully selecting the electrodes to be kept, the classification performance may be even superior to that of using the whole time series. A natural extension for this work would be to extend the geometry-aware DR methods to take into account the physiological information conveyed by the EEG signals. The goal would be to have an automatic method for reducing the dimensionality that performs as if (or even better) we had selected the electrodes by hand using a priori information.

3.3 Non-linear methods

In this section, we consider non-linear methods for reducing the dimensionality of data points. The mapping that we search is from the original data space (e.g., $\mathcal{P}(d)$ for a set of SPD matrices) to an Euclidean space of reduced-dimensionality. There have been several proposals in the literature for doing non-linear DR and we choose to focus our discussion on one of them, called *diffusion maps* [CL06]. In what follows, we explain the reasons for this choice and present how the method works. We show how to apply diffusion maps to when the data points are multivariate time series and illustrate its use on examples with EEG data.

3.3.1 Literature review

Although linear dimensionality reduction techniques are widely used in machine learning and other related fields, they have two important drawbacks [Laf04]:

(1) They only search for linear transformations on the data points, so they may not be sufficiently rich to capture the geometric structure of datasets that live in a manifold with non-zero curvature (i.e., different than an hyperplane);

(2) They try to find a global transformation that preserves both the local and global geometries of the original dataset, a rather daunting task for a transformation parametrized by a single matrix. Furthermore, preserving large distances between data points is often irrelevant, since there is no interpretable difference between the cases when two high-dimensional points are far from each other or 'very' far from each other (see Chapter 2 of [Laf04] for a thorough discussion on this matter).

Manifold learning. In view of the intrinsic limitations of linear DR techniques, there have been several proposals in the literature for reducing the dimensionality of data points via non-linear transformations, such as: local linear embedding (LLE) [RS00], Laplacian eigenmaps [Bel03], local tangent space alignment (LTSA) [ZZ04], Hessian eigenmaps [DG03], and diffusion maps [CL06](see [Maa+08] for a survey on this topic and a comparison between methods). This collection of non-linear dimensionality reduction techniques are often called *manifold learning* algorithms, because of their shared assumption that, despite its high dimensionality, real-life data often have an underlying structure that can be well described by a low-dimensional manifold [Bel03; LL06]. Note that other approaches for non-linear DR without the low-dimensional manifold assumption exist as well, such as t-SNE [MH08] and UMAP [McI+18] .

Consider we have a dataset $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K\} \subset \Gamma$, where Γ is an abstract space for which we have defined a notion of similarity[1]: for $\boldsymbol{x}, \boldsymbol{y} \in \Gamma$, $w(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}$, with $w(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ for all $\boldsymbol{x}, \boldsymbol{y} \in \Gamma$ and $w(\boldsymbol{x}, \boldsymbol{y}) = w(\boldsymbol{y}, \boldsymbol{x})$. Non-linear DR methods search for a mapping,

$$\begin{array}{rccl} f & : & \mathcal{X} & \to & \mathbb{R}^p \\ & & \boldsymbol{x} & \mapsto & f(\boldsymbol{x}) = \left[\begin{array}{ccc} f_1(\boldsymbol{x}) & \cdots & f_p(\boldsymbol{x}) \end{array} \right]^T , \end{array} \tag{3.24}$$

which preserves the local neighboord information from the data points in $\mathcal{X}$. Note that f is defined from $\mathcal{X}$ to $\mathbb{R}^p$ and not from Γ to $\mathbb{R}^p$. This comes from the fact that most nonlinear DR methods (and, more particularly, those that we consider here) are *transductive* methods [Ben+04], meaning that they only learn an intrinsic mapping for the points in the dataset $\mathcal{X}$ into $\mathbb{R}^p$. Consequently, in principle, one would have to redo the whole manifold learning procedure every time a new data point arrives. However, there are methods based on the Nyström approximation [Fow+04] that explore the regularity of the low-dimensional manifold where the data points live and learn an approximation to f so that new samples can be directly mapped to $\mathbb{R}^p$.

In one way or another, the transformations given by the manifold learning methods cited above are all obtained via the solution of an optimization problem of the form: [CL06]

$$\underset{N(f)=1}{\text{minimize}} \sum_{\boldsymbol{x} \in \mathcal{X}} Q_{\boldsymbol{x}}(f) , \tag{3.25}$$

where N and $\{Q_{\boldsymbol{x}}, \boldsymbol{x} \in \mathcal{X}\}$ are symmetric, positive semi-definite quadratic forms acting on functions defined in Γ. More specifically, $Q_{\boldsymbol{x}}$ measures local variations of f around $\boldsymbol{x}$, whereas N acts as a normalization for f. The idea behind this cost function is that one hopes to obtain an adequate transformation using overlapping local information to capture the global structure of the space where the data points live. The solution of (3.25) can be efficiently obtained via a generalized eigenvector decomposition of the matrix defined by the quadratic form in the cost function. See [Ham+04] for a connection of these methods to the more general problem of kernel PCA [Sch+98].

In what follows, we will describe the diffusion maps (DM) method for nonlinear dimensionality reduction, which is a manifold learning technique based on ideas from diffusion processes in graphs and manifolds. We choose to focus on this method because of the many mathematical results linking it to other areas of pure and applied mathematics (we will mention only a few of them, but the interested reader is referred to [CL06] and [Mém11] for further discussion), as well as the interesting

[1] The specific form of this similarity function is application-driven and should be crafted according to the domain knowledge that we have about the space where the data points are defined

probabilistic interpretation that it gives to the problem of finding a low-dimensional embedding for a set of data points.

Random walk over a graph. The basic setting for the method of diffusion maps consists in seeing the elements of $\mathcal{X} \subset \Gamma$ as nodes of a graph G whose edges are weighted by $w(\boldsymbol{x}, \boldsymbol{y})$. The graph G represents, then, our knowledge of the geometric structure of $\mathcal{X}$. The method of diffusion maps defines a Markov random walk over G and uses the properties of such process to obtain an embedding of the data points into an Euclidean space of lower dimensionality.

To define a Markov random walk on G, we first introduce the degree $d(\boldsymbol{x})$ of a point $\boldsymbol{x} \in \mathcal{X}$ as

$$d(\boldsymbol{x}) = \sum_{z \in \mathcal{X}} w(\boldsymbol{x}, \boldsymbol{z}) \ , \tag{3.26}$$

which describes a local measure of volume of the points around $\boldsymbol{x}$. Then, by defining a matrix $\boldsymbol{P} \in \mathbb{R}^{K \times K}$, whose entries are

$$\boldsymbol{P}_{ij} = \frac{w(\boldsymbol{x}_i, \boldsymbol{x}_j)}{d(\boldsymbol{x}_i)} \ , \tag{3.27}$$

we obtain a description for the transition probabilities of a random walk defined over the graph. To see this, note that, by construction, $\sum_{j=1}^{K} \boldsymbol{P}_{ij} = 1$, so the i-th row of $\boldsymbol{P}$ may be seen as the probability distribution of the transitions of a random walk from node $\boldsymbol{x}_i$ to all other elements in $\mathcal{X}$. In other words, we have, for $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$,

$$\boldsymbol{P}_{ij} = \Pr\{\boldsymbol{x}^{t+1} = \boldsymbol{x}_j | \boldsymbol{x}^t = \boldsymbol{x}_i\} \ , \tag{3.28}$$

where $\boldsymbol{x}^t$ indicates in which point of the graph the random walk is at the time instant t. Note that because it implicitly depends of the similarity function w, quantity $\boldsymbol{P}_{ij}$ reflects the first-order neighborhood structure of the graph G. By taking powers of the matrix $\boldsymbol{P}$, we let the random walk 'run forward in time' and, as a result, we capture the structure of larger neighborhoods in the graph.

We denote by $p(t, \boldsymbol{y}|\boldsymbol{x})$ the probability distribution of a random walk landing at node $\boldsymbol{y}$ at time t, given that it started at node $\boldsymbol{x}$ at time $t = 0$. It is a famous result from Markov theory [Gal13] that if G is a connected graph (meaning that there is at least one path that links every pair of nodes), then

$$\lim_{t \to \infty} \ p(t, \boldsymbol{y}|\boldsymbol{x}) = \phi_1(\boldsymbol{y}) \ , \tag{3.29}$$

where $\phi_1 \in \mathbb{R}^K$ is the left eigenvector of matrix $\boldsymbol{P}$ with eigenvalue $\lambda_1 = 1$ and

$$\phi_1(\boldsymbol{x}_i) = \frac{d(\boldsymbol{x}_i)}{\sum_{\boldsymbol{x}_j \in \mathcal{X}} d(\boldsymbol{x}_j)} \ . \tag{3.30}$$

(Note that $\phi_1(\boldsymbol{x}_i)$ corresponds to the i-th coordinate of vector ϕ_1). This quantity is known as the *stationary distribution* of the random walk over graph G and, as seen from expression (3.30), it is proportional to the degree of each node, serving as a measure of the density of the points in $\mathcal{X}$.

Diffusion distances. Ref. [CL06] defines a metric between points of $\mathcal{X}$ based on the behavior of random walks over the graph G. This metric is called the *diffusion distance* at time $t > 0$, denoted by D_t, and its intuition is that two points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ should be considered close to each other if the corresponding conditional distributions $p(t, \cdot|\boldsymbol{x}_i)$ and $p(t, \cdot|\boldsymbol{x}_j)$ are close in some sense. Mathematically, we have that [CL06]

$$D_t^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|p(t, \cdot|\boldsymbol{x}_i) - p(t, \cdot|\boldsymbol{x}_j)\|^2_{\frac{1}{\phi_1}} = \sum_{\boldsymbol{y}\in\mathcal{X}} \frac{1}{\phi_1(\boldsymbol{y})}\Big(p(t, \boldsymbol{y}|\boldsymbol{x}_i) - p(t, \boldsymbol{y}|\boldsymbol{x}_j)\Big)^2 , \tag{3.31}$$

where the weights $1/\phi_1(\boldsymbol{y})$ penalize more the discrepancies on regions of $\mathcal{X}$ with a lower density of points. This notion of proximity reflects the intrinsic geometry of the dataset $\mathcal{X}$ in terms of the connectivity of its data points in a diffusion process (i.e., the evolution of a random walk). The advantage of this concept over the standard distance between points in the original space (e.g., the geodesic distance between two points on a manifold) is that, while the classical distance between any pair of points is independent of the location of all other points in the dataset, the diffusion distance depends on all possible paths connecting them, including those that pass through other points in the dataset. Consequently, the diffusion distance is more robust to noise and small perturbations on the dataset, since it is an averaged value over all paths connecting two points.

A remarkable result from the spectral theory of Markov processes is that the expression for the diffusion distance (3.31) can be decomposed as a sum of terms involving the eigenvectors and eigenvalues of the probability transition matrix $\boldsymbol{P}$. From [CL06] we have that

$$D_t^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{k=2}^{K} \lambda_k^{2t}\Big(\psi_k(\boldsymbol{x}_i) - \psi_k(\boldsymbol{x}_j)\Big)^2 , \tag{3.32}$$

where, for $k = 1, \ldots, K$, λ_k and ψ_k are the right eigenvalues and eigenvectors of matrix $\boldsymbol{P}$, respectively (it can be verified that $\lambda_1 = 1$ and that ψ_1 is a vector of ones, which is why the sum in (3.32) starts at $k = 2$). The vectors ψ_k are normalized as in

$$\|\psi_k\|^2_{1/\phi_1} = \sum_{\boldsymbol{x}\in\mathcal{X}} \phi_1(\boldsymbol{x})\psi_k^2(\boldsymbol{x}) = 1 \tag{3.33}$$

and we have that $\lambda_1 = 1 \geq |\lambda_2| \geq \cdots \geq |\lambda_K|$.

Embedding data points via diffusion maps. Because of the decaying behavior of the eigenvalues of $\boldsymbol{P}$ (the speed of this decay is related to the structure of the graph G and there are many works on this topic [Chu96]) we may approximate $D_t^2(\boldsymbol{x}_i, \boldsymbol{x}_j)$ using a few terms of the sum (3.32): for a given numerical accuracy δ, and a fixed value of t, there exists a dimensionality $p(\delta, t)$ such that,

$$\left| D_t^2(\boldsymbol{x}_i, \boldsymbol{x}_j) - \sum_{k=2}^{p(\delta,t)} \lambda_k^{2t} \Big(\psi_k(\boldsymbol{x}_i) - \psi_k(\boldsymbol{x}_j) \Big)^2 \right| \leq \delta \ . \tag{3.34}$$

This relation can be used to define a mapping from $\mathcal{X}$ to $\mathbb{R}^p$ defined as

$$\Psi_t : \boldsymbol{x} \mapsto \begin{bmatrix} \lambda_2^t \psi_2(\boldsymbol{x}) \\ \vdots \\ \lambda_p^t \psi_p(\boldsymbol{x}) \end{bmatrix} , \tag{3.35}$$

so that

$$D_t^2(\boldsymbol{x}_i, \boldsymbol{x}_j) \simeq \| \Psi_t(\boldsymbol{x}_i) - \Psi_t(\boldsymbol{x}_j) \|^2 \ . \tag{3.36}$$

The mapping Ψ_t can be interpreted as a parametrization of the dataset $\mathcal{X}$ as a cloud of points in a lower-dimensional Euclidean space $\mathbb{R}^p$. The Euclidean distance between the embedded data points is an approximation of the diffusion distance between the original data points: two points that are close in the embedded space are close in terms of the diffusion distance.

Note that under the optimization framework given by (3.25), the diffusion maps method yields a transformation $f = \Psi_t$ which is the minimizer of

$$\sum_{\boldsymbol{x} \in \mathcal{X}} \left(\sum_{\boldsymbol{y} \in \mathcal{X}} w(\boldsymbol{x}, \boldsymbol{y}) \Big(f(\boldsymbol{x}) - f(\boldsymbol{y}) \Big)^2 \right) , \tag{3.37}$$

with the normalization constraint

$$\sum_{\boldsymbol{x} \in \mathcal{X}} \phi_1(\boldsymbol{x}) f^2(\boldsymbol{x}) = 1 \ . \tag{3.38}$$

Algorithm 1 summarizes the diffusion maps procedure.

Algorithm 1: Diffusion maps

Input: a set of data points $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K\} \subset \Gamma$ and the dimensionality p of the space where we want to embed them.

Output: a set of embedded data points $\Psi_t(\mathcal{X}) = \{\Psi_t(\boldsymbol{x}_1), \ldots, \Psi_t(\boldsymbol{x}_K)\} \subset \mathbb{R}^p$

1 Define a notion of similarity between two points $\boldsymbol{x}_i, \boldsymbol{x}_j \in \Gamma$. This similarity function, $w : \Gamma \times \Gamma \to \mathbb{R}$, is often called a *kernel* and it must satisfies two properties: w is symmetric, $w(\boldsymbol{x}, \boldsymbol{y}) = w(\boldsymbol{y}, \boldsymbol{x})$, and w is positivity preserving, $w(\boldsymbol{x}, \boldsymbol{x}) \geq 0$.

2 Form the matrix $\boldsymbol{K} \in \mathbb{R}^{K \times K}$ for which $\boldsymbol{K}_{ij} = \exp\left(-\frac{d^2(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\varepsilon}\right)$.

3 Set $\boldsymbol{D} = \text{diag}(\boldsymbol{K}\mathbf{1}_K)$, where $\text{diag}(\boldsymbol{v})$ is a diagonal matrix whose values come from vector $\boldsymbol{v}$, and $\mathbf{1}_K$ is a K-dimensional vector filled with ones.

4 Form matrix $\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{K}$ and obtain its set of left eigenvectors, $\{\phi_k\}_{1 \leq k \leq K}$, and right eigenvectors, $\{\psi_k\}_{1 \leq k \leq K}$, as well as their associated eigenvalues, $\{\lambda_k\}_{1 \leq k \leq K}$.

5 Normalize the right eigenvectors so that we have, for $k = 1, 2, \ldots, K$,

$$\sum_{\boldsymbol{x} \in \mathcal{X}} \phi_1(\boldsymbol{x}) \psi_k^2(\boldsymbol{x}) = 1 \, .$$

6 Obtain a mapping for each data point $\boldsymbol{x} \in \mathcal{X}$ given by

$$\Psi_t(\boldsymbol{x}) = \begin{bmatrix} \lambda_2^t \psi_2(\boldsymbol{x}) \\ \vdots \\ \lambda_p^t \psi_p(\boldsymbol{x}) \end{bmatrix} .$$

A common choice for w is the Gaussian kernel, defined as

$$w(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{d^2(\boldsymbol{x}, \boldsymbol{y})}{\varepsilon}\right) , \tag{3.39}$$

where $d(\boldsymbol{x}, \boldsymbol{y})$ is some notion of distance defined in Γ (for Eucliden spaces, this can be simply the Euclidean distance) and the scaling parameter ε sets a notion of 'how large' are the neighborhoods that we consider in Γ. When $\varepsilon \to 0$, the Gaussian kernel behaves as a Dirac distribution on $\mathcal{X}$. There are many empirical ways of deciding a value for ε that seems the most adequate for a given application. The R package `diffusionMap`[2] uses the median value of the distances of each element in $\mathcal{X}$ to its pK nearest neighbor, where p is a percentage usually in the range of 1% to 5%. A more mathematically justified approach was proposed in [Coi+08] and refined in [Ber+15] for when $\Gamma = \mathbb{R}^d$. Note that the matrix $\boldsymbol{K}$ defined in Algorithm 1 has

[2] available at: `https://cran.r-project.org/web/packages/diffusionMap/` (last checked on June 18th, 2019)

the same structure of what is often called a 'kernel matrix' in other machine learning methods from the literature [Has+09]. In this work, however, we do not call $\boldsymbol{K}$ a kernel matrix, because we can not guarantee that it will always be positive definite for any choice of ε. This is a consequence of the results described in [Fer+15], which studies the behavior of kernel matrices defined with different similarity measures. Fortunately, the diffusion maps algorithm does not require $\boldsymbol{K}$ to be positive definite, it only needs to be symmetric and all its values should be positive.

3.3.2 Manifold learning for multivariate time series

When applying diffusion maps to datasets consisting of several multivariate time series, as in $\mathcal{X}' = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K\}$, one needs to define an adequate notion of similarity between the elements of the dataset. Using the Riemannian geometry framework presented in Chapter 2, such similarity can be defined as

$$w(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\delta^2(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\varepsilon}\right) , \tag{3.40}$$

where the choice of δ depends on how the statistics of the time series are parametrized:

- If each $\boldsymbol{x}_i$ is parametrized by its spatial covariance matrix $\boldsymbol{C}_i$, then

$$\delta^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = \delta_R^2(\boldsymbol{C}_i, \boldsymbol{C}_j) = \| \log(\boldsymbol{C}_i^{-1/2} \boldsymbol{C}_j \boldsymbol{C}_i^{-1/2}) \|_F^2 . \tag{3.41}$$

- If each $\boldsymbol{x}_i$ is parametrized by its cross-spectral density matrices $\boldsymbol{S}_i(f)$, with $f \in \mathcal{F}$ (where $\mathcal{F}$ is some set of frequencies of interest), we have

$$\delta^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{f \in \mathcal{F}} \delta_R^2(\boldsymbol{S}_i(f), \boldsymbol{S}_j(f)) . \tag{3.42}$$

Manifold learning methods have been applied to sets of SPD data points [GV08], but its use to study multivariate time series is relatively rare. In fact, works such as [Tal+13] and [Hay+05] have applied diffusion maps to study time series only in the univariate case, defining the similarity function based on the Mahalanobis distance and *dynamic time warping*, respectively. Ref. [GC07] embeds EEG evoked potentials into a lower-dimensional manifold, but with a similarity function based on the Euclidean distance between the epochs. To the best of our knowledge, our work in [Rod+18] is one of the first investigations where the Riemannian geometric framework for multivariate time series is used with diffusion maps and applied to study EEG signals.

In this chapter, we have used the results of the diffusion maps embedding mostly for visualization purposes, but there are many other relevant applications for it on time series analysis. For instance, one could apply DM in an unsupervised setting for clustering epochs related to different experimental conditions (e.g., different sleep

states); this is often called *spectral clustering* [Lux07]. Another use would be in a semi-supervised context, where one has access to the labels of a few epochs in a dataset and then propagates this information to other data points based on their proximity in terms of the diffusion distance. Also, one could use the smoothness of the low-dimensional manifold where the data points live as a regularization term in regression and classification. This is called *Laplacian regularization* in the machine learning literature [Bel03].

3.3.3 Numerical illustrations

In what follows, we apply the diffusion maps algorithm to analyse EEG signals coming from three different experiments. We parametrize the time series via their spatial covariance matrices (for BCI and resting-state datasets) and their cross-spectral density matrices (for the sleep dataset). At each time, we generate scatter plots representing the first two dimensions of the embedded data points (ψ_1 and ψ_2) and use them for performing unsupervised explorations of the datasets.

BCI datasets. Our first example uses the same BCI database considered in Section 3.2.3, where the EEG epochs are recorded on a motor imagery (MI) experimental paradigm with 64 electrodes. Each one of these epochs lasts two seconds (sampling frequency was 160 Hz) and is associated to one of two classes, 'hands' or 'feet'. We start by discussing the qualitative differences between embeddings obtained when the diffusion maps are defined with a kernel function measuring the similarity of two EEG epochs, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, in two cases:

(1) Directly from the Euclidean distance between two T-sample realizations, i.e., $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$, both defined in $\mathbb{R}^{d\times T}$. In this case, in Equation (3.40) we have

$$\delta^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{X}_i - \boldsymbol{X}_j\|_F^2 \,. \tag{3.43}$$

(2) From their spatial covariance matrices, $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$, using the geodesic distance in the SPD manifold described in Equation (3.41).

The scatter plot in Figure 3.4 shows that the embedding of the epochs of each class (represented in different colors) are significantly less separated when we use the similarity function from Eq. (3.43) as compared to when we use the Riemannian distance between SPD matrices. This result is not surprising and is related to the fact that, when we directly compare $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$, we are actually comparing just two realizations of a stochastic process, whereas the distance based on their statistical descriptors compares their actual generating processes.

Our next example illustrates the effects of the linear DR procedures discussed in Section 3.2 on the embedding of the data points with diffusion maps. We consider four different cases:

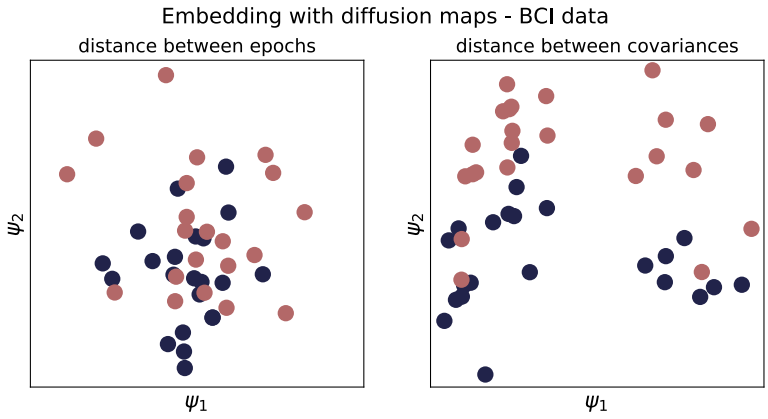

Fig. 3.4: Embedding with diffusion maps of the epochs recorded during the BCI experiment described in the text. Each point corresponds to a trial $\boldsymbol{X}_i \in \mathbb{R}^{64\times 320}$ and the colors indicate to which condition it is associated ('hands' is blue and 'feet' is red). We use two definitions for the similarity function: (left) Euclidean distance between the X_i matrices and (right) Riemannian distance between the spatial covariance matrices of each epoch.

(1) **full**: use the epochs $\boldsymbol{X}_i \in \mathbb{R}^{64\times 320}$ and their covariance matrices $\boldsymbol{C}_i \in \mathcal{P}(64)$

(2) **rie-PCA**: reduce the dimensionality of the epochs using a projector matrix $\boldsymbol{W}^{(\text{uns})}_{\text{rie-PCA}} \in \mathbb{R}^{64\times 12}$ obtained via the optimization procedure discussed in Section 3.2.2. The SPD matrices are defined in $\mathcal{P}(12)$.

(3) **SELg**: select 12 out of 64 electrodes from the epochs $\boldsymbol{X}_i$ which are the most physiologically relevant for the cognitive tasks executed during a motor imagery experiment (see Figure 3.1 for a visual depiction of the placement of the electrodes). The SPD matrices are defined in $\mathcal{P}(12)$.

(4) **SELb**: select 12 out of 64 electrodes from the epochs $\boldsymbol{X}_i$ which we know do not have much relevant information for separating the classes (see Figure 3.1 in page for a visual depiction of the placement of the electrodes). The SPD matrices are defined in $\mathcal{P}(12)$.

The results of the embedding in each one of these cases is portrayed in Figure 3.5. There are some interesting aspects to observe:

- The embeddings in the 'full' and 'rie-PCA' cases are very similar to each other. This goes in line with what was expected from the objective function defining $\boldsymbol{W}^{(\text{uns})}_{\text{rie-PCA}}$.
- The classes of the data points in the 'SELg' embedding are more separated and condensed as compared to the other cases. This can be directly linked to the superior classification performances of the 'SELg' method that we observed in Section 3.2.3.
- The embedding with 'SELb' yields data points whose classes are clearly not well separated. This result is compatible with the poor classification performance observed for the 'SELb' method in Section 3.2.3.

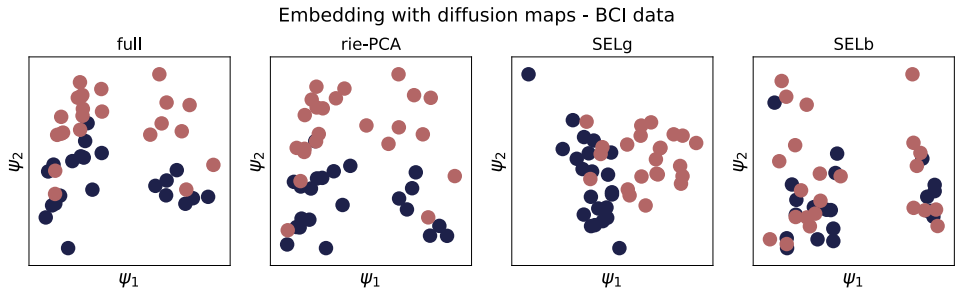

Fig. 3.5: Embedding with diffusion maps of the EEG epochs recorded during the BCI experiment described in the text. Each point corresponds to a trial and the colors indicate to which condition it is associated ('hands' is blue and 'feet' is red). We compare the four pipelines considered in Section 3.2.3. In 'full', the epochs are left untouched, in 'rie-PCA' their dimensionality is reduced by applying a 64×12 projector matrix (see text for details on how this matrix is obtained), in 'SELg' the 12 most physiologically relevant electrodes are selected on each epoch, and in 'SELb' we choose 12 electrodes that do not carry discriminative information regarding the classes of the experiment.

We perform a last analysis with the BCI dataset. Instead of embedding the EEG epochs associated to each class, we use a sliding-window over the continuous recording of the BCI experiment and obtain a sequence of small windowed epochs, each one related to a small interval of time. We estimate the spatial covariance matrices of these small windows and proceed with the diffusion maps method to obtain a low-dimensional embedding. We use a window consisting of 160 points (equivalent to one second) and slide it through approximately 20 seconds of raw EEG signal (the sliding window has 95% overlap). During these 20 seconds, the subject alternates between two states: a 'resting state', during which he has no particular guideline to follow, and an 'active state', during which he is asked to perform an imagined movement task. Figure 3.6 portrays the embedding with diffusion maps of the set of spatial covariance matrices associated to each window of time. Each dot represents the embedding of the SPD matrix that parametrizes the statistics of one small interval of time, and the different colors indicate in which condition the subject was ('active' or 'rest'). We note that the embedded points related to the 'active' state tend to concentrate in the same region of the embedded space, whereas the 'rest' epochs are spread. This result was expected, since the subject's motor imagery signals in the 'active' state tend to have similar statistics, whereas in the 'rest' state the EEG signals are not expected to have any statistical consistency between them.

Sleep recordings. Our second example uses data available at the `Physionet` database [Ter+01; Gol+00] and contains recordings from 9 EEG electrodes on a subject sleeping for approximately 8 hours. The original sampling frequency was 512 Hz but we downsampled it to 128 Hz after band-filtering the EEG signals between 8 Hz and 40 Hz. A specialist in sleep data analysis was responsible for cutting the recordings into several 30-sec clips and then classifying them according

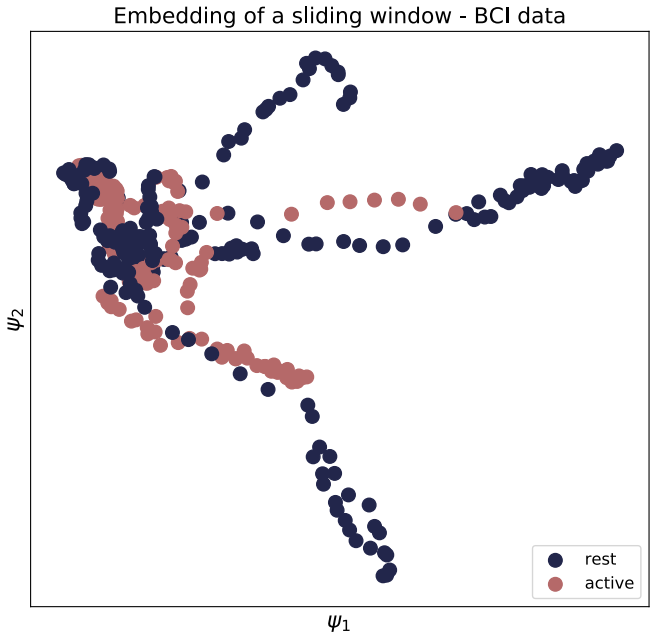

Fig. 3.6: Embedding with diffusion maps of a sliding window (160 points, 95% overlap) applied to the BCI dataset described in the text. Each point corresponds to a window and the color indicates the subject's experimental condition during the time of the window ('active' is red and 'rest' is blue).

to which sleep stage (S1, S2, S3, and REM) they belonged. We ended up with a dataset containing $K = 564$ trials of $T = 3840$ samples each.

The embedding of this dataset with difusion maps is displayed in Figure 3.7. We did not include the S1 sleep stage trials because they are associated to light sleep and their statitical behavior is not well captured by the Riemannian geometric framework that we use. We carried out the diffusion maps procedure using distances (3.41) and (3.42) and observed a better separation of points with the latter. We link this to the traditional way of doing classification of sleep stages, which relies on the spectral content of the trials and indicates that their spectrum is an important feature to classify them. These results are interesting because they show the possibility of extracting information from a dataset containing sleep recordings based only on an adequate notion of similarity between signals.

It could be argued that the quality of the clustering with each type of distance also depends on the number of eigenvectors considered from the diffusion maps (Figure 3.7 shows the results with just the first two). To verify this hypothesis, we assessed the quality of the clustering performed by a Gaussian mixture model (GMM) [Bis07] on the embedded space considering an increasing number of dimensions. Assuming as true labels the classes given by a specialist, we did not observe any improvement on the modified Rand scores [HA85] when considering more eigenvectors for any of the distances. The modified Rand score for a GMM clustering using distance (3.42) on the two first eigenvectors is 0.65 (in a scale from 0 to 1, where 1 is the best). We

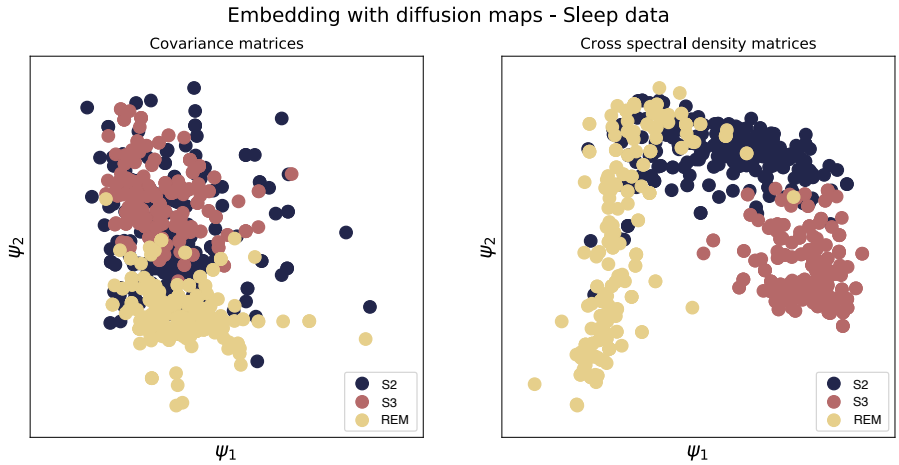

Fig. 3.7: Diffusion maps embedding of the sleep data described in the text. Each point is a 30-sec clip and the colors indicate to which sleep stage it is associated. We used diffusion maps with two types of similarity functions: (left) based on the distance between spatial covariance matrices and (right) based on the distance between cross-spectral density matrices.

consider this a satisfying score because very few assumptions about the dataset were made and the classification was completely unsupervised.

Resting state. Our last example uses the `ALPHA.EEG.2017-GIPSA` database [Cat+18]. It consists of EEG recordings from 16 EEG electrodes of a healthy subject instructed to alternate between keeping his eyes open or closed every eight seconds. The sampling frequency is 128 Hz and the signals were filtered between 8 Hz and 40 Hz. For our analysis, we used a window with $L = 128$ points and slid it through the EEG recordings with 75% overlap. To each window we associate one class label (indicating whether the subject had his eyes open or closed) and we estimate its covariance matrix, which serves as a descriptor of the instantaneous statistical behavior of the time series. Figure 3.8 portrays the embedding via diffusion maps of the set of windows, with each color representing one state. We observe a clear separation between the embedded points from each condition, which can be explained by the rather strong effect that the alpha waves produced in the brain's occipital region have over the EEG recordings when a person has his eyes closed. Figure 3.9 represents the values of the first axis of the diffusion maps embedding (i.e., the values of the first eigenvector in the spectral decomposition described in Section 3.3). We also plot a curve indicating whether the subject had his eyes open or closed and observe that the values of the first eigenvector follow very closely the subject's state. This result shows that we may also use diffusion maps to track the time evolution of the states of a dynamical system. In fact, studying the evolution of the states of a dynamical system with a sliding window is a very common procedure in nonlinear physics and, more particularly, in the study of chaotic systems [BK86; Sau+91]. It is usually called a 'time-lagged embedding' and linear methods such as multidimensional scaling and PCA are traditionally used for the embedding of

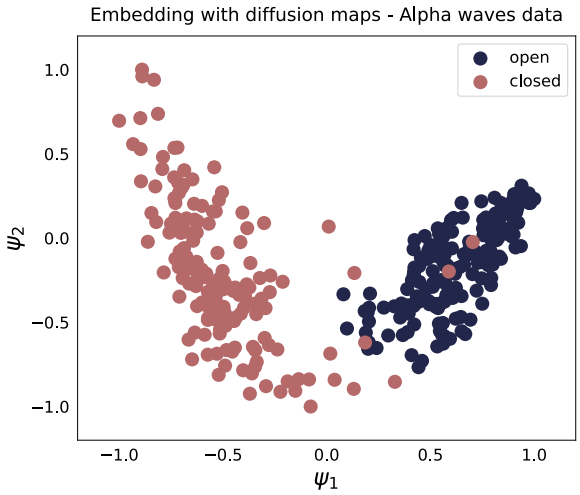

Fig. 3.8: Diffusion maps embedding of the alpha waves experiment described in the text. Each dot corresponds to a window with $L = 128$ samples which slides over the recording with 75% overlap. The similarity function is based on the Riemannian distance between SPD matrices. The color of each dot indicates whether the subject had his eyes open or closed during the time associated to the window.

the samples. Recently, extensions of this approach to non-linear embeddings with diffusion maps have been proposed in [GM12] and [Ber+13], but both are based on Euclidean distances between short windows of univariate recordings. To the best of our knowledge, the embedding displayed in Figure 3.9 (and also Figure 3.6) is the first done in the context of multivariate time series recordings and using a principled method for measuring the similarities between the windows.

3.4 Conclusion

In this chapter, we have considered the problem of dimensionality reduction (DR) applied to multivariate time series. We have applied both linear and non-linear methods to recordings of EEG data from different contexts. A remarkable result was that of being possible to linearly reduce the dimensionality of multivariate time series without decreasing, in average, the performance of classifiers applied to the data points. Such result goes in line with what was expected from our discussion of invariances and redundancies in multivariate time series, where two representations of the same phenomenon, but with different dimensionalities, share commonalities between them that can be explored. Another important result was the extension of the method of diffusion maps to explore datasets consisting of multivariate time series. This allows for different kinds of analysis of time series, such as clustering of epochs related to different sleep states or tracking the evolution of latent parameters that govern the statistics of a time series.

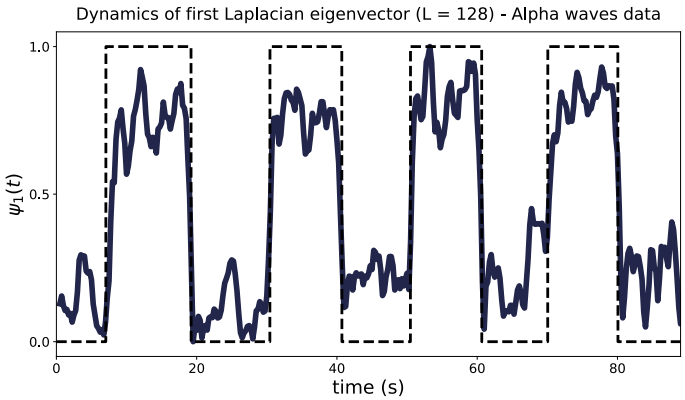

Fig. 3.9: First dimension of the diffusion maps embedding of the samples in a sliding-window ($L = 128$ and 75% overlap) running over the Alpha waves EEG data described in the text. Each point of the blue curve corresponds to a different window of time. The dashed line indicates when the subject had the eyes open (0) and closed (1).

Our results also show that the parametrization of multivariate time series with SPD matrices yields better results in dimensionality reduction. In the linear case, we have shown that it is better to use the geometry-aware method proposed by [Har+17] to reduce the dimensionality of SPD matrices as compared to the classical PCA, which consider the matrices as elements of an Euclidean space. In the non-linear case, we have shown that the similarity matrix based on the statistics of multivariate time series gives much better diffusion maps embedding as compared to similarities based on the Frobenius distance between realizations of the time series.

It is worth mentioning that we could have parametrized the multivariate time series via their cross-spectral density matrices, which are Hermitian positive definite (HPD) matrices. Such description is more complete than just using spatial covariance matrices and has exactly the same geometry as for SPD matrices. However, we have preferred to work with SPD matrices in this chapter for the sake of simplicity of exposition. Note that one could directly extend the results described here to a parametrization via cross-spectral density matrices by simply considering each frequency of the spectrum independently.

Transfer learning

4

Contents

4.1 Introduction **77**
4.1.1 Contributions 78
4.2 Literature review **79**
4.2.1 Learning from different domains 79
4.2.2 Distribution matching 81
4.2.3 Transfer learning for BCI 83
4.3 Riemannian Procrustes analysis **85**
4.3.1 Problem statement 86
4.3.2 An Euclidean motivation 87
4.3.3 Transformations in the SPD manifold 88
4.3.4 Summary of the RPA method 93
4.3.5 A statistical interpretation of the steps in RPA 93
4.3.6 Relation with optimal transport 95
4.3.7 A time series interpretation of the steps in RPA 96
4.3.8 RPA as an optimization problem 97
4.4 Numerical illustrations: simulated data **98**
4.4.1 The dataset 98
4.4.2 Illustration of the steps of RPA 98
4.4.3 Classification accuracy after RPA 99
4.5 Numerical illustrations: real data **101**
4.5.1 The datasets 102
4.5.2 Comparing cross-subject classification accuracies 102
4.5.3 An exploratory analysis of cross-subject classification . . 106
4.5.4 The role of the size of $\mathcal{T}_\ell$ 112
4.5.5 Combining information from multiple subjects 112
4.6 Conclusion **114**

List of notations and acronyms of the chapter

EEG	electroencephalography
BCI	brain-computer interface
HPD	Hermitian positive definite
SPD	symmetric positive definite
RG	Riemannian geometry
AIRM	affine-invariant Riemannian metric
MDM	minimum distance to mean classifier
TL	transfer learning
OT	optimal transport
RPA	Riemannian Procrustes analysis
RCT	re-centering
STR	stretching
ROT	rotation
DM	diffusion maps
MI	motor imagery
ROC	receiver operating characteristic
AUC	area under the ROC curve
$\mathbb{R}^d$	set of d-dimensional real vectors
$\boldsymbol{x}$	multivariate time series
$\boldsymbol{C}$	spatial covariance matrix
δ_E	Frobenius distance between two matrices
δ_R	AIRM-induced distance between two HPD matrices
$\mathcal{P}(d)$	manifold of d-dimensional HPD matrices
$\boldsymbol{M}^{\mathcal{X}}$	geometric mean of the HPD matrices in a set $\mathcal{X}$
K	number of data points
d	dimensionality of data points
$\mathcal{S}$	*source* dataset
$\mathcal{T}$	*target* dataset
$\mathcal{T}_\ell$	labeled partition of $\mathcal{T}$
$\mathcal{T}_u$	unlabeled partition of $\mathcal{T}$

4.1 Introduction

Classical machine learning algorithms usually suppose that the statistical distribution of the data points used to train a classifier is the same as that of the data points to which the classifier is applied. However, in many practical cases, this is not true. For instance, in BCI datasets the statistics of the EEG epochs of a subject may vary between recording sessions and between different subjects. This happens also in computer vision, where the statistics of the data may vary due to changes in lighting conditions and acquisition devices, or in speech processing systems, where the changes in background noise and the differences in speaker genders and voice tonalities may affect the statistics of the signals.

In this chapter, we present and discuss methods for performing statistical analysis on samples from a dataset (the *target* dataset) using information from another dataset (the *source* dataset). A natural advantage of reusing samples from other datasets is that it leads to algorithms which are 'data-efficient' and that can explore all the available information that might be useful for accomplishing a certain task. Furthermore, by reusing information that is already available, such algorithms may be considered as 'ecology-aware', in the sense of avoiding unnecessary energy consumption for obtaining and storing new data samples. Such concerns are very relevant in machine learning and have uses in many contexts with different types of data. The domain of research dealing with this kind of problem is called *transfer learning* and has been covered in several works in the literature (see [PY10] for a survey). The common argument in all such proposals is that the discrepancy between the statistics of the datasets is the main responsible for the poor performance of classifiers directly trained on a *source* dataset and applied to a *target* dataset. The goal in transfer learning methods, then, is to determine a set of transformations over the data points that minimizes the divergence between the statistics of the datasets.

In what follows, we will be particularly interested in the case where the data points of the *source* and *target* datasets are symmetric positive definite (SPD) matrices used to describe the statistics of time series epochs. This is particularly relevant to EEG-based brain computer interfaces (BCI), where the statistics of the data generated by two subjects may be very different. This is reflected in how the SPD matrices used to describe the EEG epochs are distributed in the SPD manifold. To analyse such kind of data points, we use the Riemannian geometric framework for multivariate time series presented in Chapter 2, which was then defined in the HPD manifold and can be naturally particularized to the SPD manifold (we will abuse notation and denote the SPD manifold of $d \times d$ SPD matrices by $\mathcal{P}(d)$). We propose, then, a series of transformations over the two datasets with the goal of making their

distributions become as similar as possible according to some notion of distance between statistical distributions in the SPD manifold.

The basic assumption that we make in this chapter is that, although the time series of two datasets may have different statistical distributions due to a number of factors, if they are related to the same kind of physical phenomenom, then there should be commonnalities that could be explored for a joint analysis. For instance, if two subjects perform the same set of motor imagery tasks, even if the statistics of the EEG epochs in each dataset are very different, the way that the classes distinguish between each other should not be too different from one subject to the other. By exploring this kind of invariance between time series, we propose a simple method for transforming the SPD data points from a *target* dataset and make its statistical distribution more similar to that of a *source* dataset. This procedure is called *Riemannian Procrustes analysis* (RPA) and is the topic of most of the following discussion.

The remainder of the chapter goes as follows: we begin in Section 5.2 with a review of the literature on transfer learning. Section 4.3 presents the method of Riemannian Procrustes analysis. Section 4.4 and Section 4.5 discuss numerical illustrations on simulated and real data, respectively. Section 4.6 concludes the chapter.

4.1.1 Contributions

The content of this chapter is based on (and extends) the works on two published papers :

> P. L. C. Rodrigues, C. Jutten, and M. Congedo, “Riemannian procrustes analysis: transfer learning for brain-computer interfaces", IEEE Transactions on Biomedical Engineering, pp. 1–1, 2018.

and

> P. L. C. Rodrigues, M. Congedo, and C. Jutten, “‘When does it work ?’: an exploratory analysis of transfer learning for BCI", BCI 2019 - 8th Graz Brain Computer Interface Conference, Graz, Austria, Sep. 2019.

Our main contributions have been to propose the RPA method and investigate its performance on a number of practical cases related to EEG-based brain computer interfaces datasets. It is also worth mentioning the fact of having used the `MOABB` framework [JB18] for downloading, processing, and analysing the EEG data, serving as a practical illustration of this powerful benchmarking tool. Python code for the RPA method is available in:

`http://www.github.com/plcrodrigues/PhD-Code/`

4.2 Literature review

In this section, we present an overview of concepts related to transfer learning. We start by commenting on how the discrepancy between the statistics of two datasets affects the generalization error of a classifier trained with data points from one dataset and used to label points from another dataset. Then, we present some of the procedures proposed in the literature for coping with this kind of problem, giving particular attention to a class of methods that do 'distribution matching'. We conclude with a review of the literature on transfer learning applied to brain computer interfaces.

4.2.1 Learning from different domains

In its essence, a machine learning algorithm is the process of extracting knowledge from a dataset, the *training* dataset, and extrapolating this knowledge to another dataset, the *testing* dataset. When both datasets have the same statistical distribution, classifiers such as logistic regression, linear discriminant analysis and support vector machines can be directly used as tools for determining the classes of unlabeled samples in the testing dataset. However, when the statistics on the datasets are different, one has to take into account this discrepancy before using the classifier.

Consider a binary classifier $h : \mathcal{O} \to \{0, 1\}$, where $\mathcal{O}$ is the set of objects to which the classifier assigns labels (e.g. images, time series, text, etc.). The generalization error (or risk) of this classifier in a dataset $\mathcal{X} \subset \mathcal{O}$ whose statistical distribution is $\mathcal{D}_\mathcal{X}$ and labeling function is $f_\mathcal{X} : \mathcal{O} \to \{0, 1\}$ is defined as

$$R_\mathcal{X}(h) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_\mathcal{X}} \left[\mathbf{1}_{\{h(\boldsymbol{x})\neq f_\mathcal{X}(\boldsymbol{x})\}} \right] , \tag{4.1}$$

that is, the probability of h assigning a label to a data point that is different than what $f_\mathcal{X}$ would give. Ref. [BD+09] expresses a relation between the generalization error of a classifier trained with points from a *source* dataset $\mathcal{S}$ and applied to data points in a *target* dataset $\mathcal{T}$. We have that

$$R_\mathcal{T}(h) \leq R_\mathcal{S}(h) + d_1(\mathcal{D}_\mathcal{S}, \mathcal{D}_\mathcal{T}) + \Delta(f_\mathcal{S}, f_\mathcal{T}) , \tag{4.2}$$

where the second term is a L^1 measure of divergence between statistical distributions,

$$d_1(\mathcal{D}_\mathcal{S}, \mathcal{D}_\mathcal{T}) = 2 \sup_{B\in\mathcal{B}} |\Pr_{\mathcal{D}_\mathcal{S}}(B) - \Pr_{\mathcal{D}_\mathcal{T}}(B)| , \tag{4.3}$$

with $\mathcal{B}$ the set of measurable subsets under distributions $\mathcal{D}_\mathcal{T}$ and $\mathcal{D}_\mathcal{S}$. The term $\Delta(f_\mathcal{S}, f_\mathcal{T})$ is related to discrepancies in the labelling functions of the datasets and is expected to be small.

Equation (4.2) shows that the difference in generalization error of a classifier in the *source* and *target* datasets is small when the discrepancy between the statistical distributions in each dataset is small. Thus, algorithms that gather information from datasets with different statistics should, apart from minimizing the generalization error of a classifier in each dataset, also transform the data points on both datasets so that their statistical distributions become similar to one another. It is worth mentioning that $d_1(\mathcal{D}_\mathcal{S}, \mathcal{D}_\mathcal{T})$ is a rather difficult quantity to estimate, because of its combinatorial behavior related to the 'sup' operation [BD+09]. This is why different works in the literature [BD+09; Man+09; Red+17] have proposed other bounds similar to (4.2) but based on divergences between statistical distributions that are easier to estimate.

The branch of Machine Learning that studies the effects of mismatches between statistical distributions and how to cope with them, is called *transfer learning*. It has been of great interest in several domains, such as in computer vision, where the statistics of the data may vary due to changes in lighting conditions and acquisition devices, in speech processing systems, where the changes in background noise and the differences in speaker genders and voice tonalities may affect the statistics of the signals, or in brain computer interfaces, where the statistics of the EEG data from two subjects may be very different. Based on the taxonomy presented in [PY10], transfer learning may be categorized into three large classes of algorithms:

(a) In inductive transfer learning, the tasks in the *source* and *target* datasets are different. For instance, one may want to classify species of dogs in one dataset and species of cats in another dataset. The statistical distributions on each dataset may be the same or not.

(b) In transductive transfer learning, the tasks on both datasets are the same but their statistical distributions are not.

(c) In unsupervised transfer learning, the tasks in each dataset are not the same but are at least related (e.g. classification of dog species from images or from barks). However, one does not have access to the labels from the *source* nor from the *target* datasets.

In this thesis, we are mostly interested in problems related to transductive transfer learning; more specifically, the paradigms of unsupervised and semi-supervised domain adaptation, where one has access to all the labels from the *source* dataset and, in the semi-supervised case, to a few labels from the *target* dataset.

There are many approaches for domain adaptation and they are all based on transforming the samples from the *source* and *target* datasets. One such method is called importance-weighting [Cor+10]. It relies on the idea of giving different weights to each sample in the *target* dataset so that its statistical distribution gets closer to that of the *source* dataset. Another approach is to learn a transformation for

each dataset that maps their data points to a common space where the statistics of the new data points are aligned; this is called subspace alignment in the literature [Fer+13]. Recently, methods based on the theory of optimal transport have found great success in the machine learning community [Cou+17]. They promote geometric transformations of the data points from both datasets in order to match their statistical distributions, in an approach often called 'distribution matching'; we focus on this type of method in the next sub-section.

4.2.2 Distribution matching

We begin by considering a simple case of mistmatch between the statistics of two datasets. Suppose that the *source* dataset ($\mathcal{S}$) has data points which follow an univariate Gaussian distribution with mean $\mu_{\mathcal{S}}$ and variance $\sigma^2_{\mathcal{S}}$ and that the *target* dataset ($\mathcal{T}$) follows a Gaussian distribution with mean $\mu_{\mathcal{T}}$ and variance $\sigma^2_{\mathcal{T}}$. For the data points from $\mathcal{T}$ to be comparable to those from $\mathcal{S}$ (comparable in the sense of being possible to do classification, clustering, or any other kind of statistical analysis with data from both datasets), it is necessary to define a transformation that makes the statistics of the data points the same. We define such transformation as:

$$\begin{aligned} T_{\mathcal{S}\to\mathcal{T}} \;:\; \mathbb{R} &\to \mathbb{R} \\ x &\to \frac{\sigma_{\mathcal{T}}}{\sigma_{\mathcal{S}}}(x-\mu_{\mathcal{S}})+\mu_{\mathcal{T}}\;, \end{aligned} \tag{4.4}$$

which makes any random variable sampled from $\mathcal{S}$ follow the same statistical distribution as if it was sampled from $\mathcal{T}$. We say that $T_{\mathcal{S}\to\mathcal{T}}$ matches the statistical distributions of the two datasets.

Optimal transport. The definition of $T_{\mathcal{S}\to\mathcal{T}}$ in (4.4) relies on the knowledge of the statistical laws that the datasets $\mathcal{S}$ and $\mathcal{T}$ follow precisely. However, in practice one very rarely has access to such information, so a different method for matching the distributions of the two datasets is in order. One possible approach is based on the concept of optimal transport, which is a centuries old discipline in mathematics and has recently gained considerable interest in the field of domain adaptation [Vil09; Cou+17; PC19].

The optimal transport problem (OT) has been defined in various forms in the literature. In this section, we present a version originally formalized by the French mathematician Gaspard Monge in 1781. His motivation at the time was to study the pratical problem of how to optimize the total amount of work of a group of workmen when transforming a terrain with an initial landscape into a terrain with a given target landscape. In mathematical terms, we may write this problem as: consider two multivariate probability densities defined in $\mathbb{R}^d$ and denoted as $\nu_{\mathcal{S}}$ and $\nu_{\mathcal{T}}$ (in fact, we could have chosen any two positive functions having the same

normalization in $\mathbb{R}^d$). We want to determine a mapping $T_{\mathcal{S}\to\mathcal{T}} : \mathbb{R}^d \to \mathbb{R}^d$ that solves the optimization problem

$$\min_{T_{\mathcal{S}\to\mathcal{T}}} \int_{\mathbb{R}^n} c\Big(\boldsymbol{x}, T_{\mathcal{S}\to\mathcal{T}}(\boldsymbol{x})\Big)\nu_{\mathcal{S}}(\boldsymbol{x})\ \mathrm{d}\boldsymbol{x}\ , \tag{4.5}$$

such that for every $B \subset \mathbb{R}^d$ the transformation $T_{\mathcal{S}\to\mathcal{T}}$ preserves the probability measures in both spaces, as in

$$\int_B \nu_{\mathcal{T}}(\boldsymbol{x})\ \mathrm{d}\boldsymbol{x} = \int_{T^{-1}_{\mathcal{S}\to\mathcal{T}}(B)} \nu_{\mathcal{S}}(\boldsymbol{x})\ \mathrm{d}\boldsymbol{x}\ . \tag{4.6}$$

The cost function $c(\boldsymbol{x}, \boldsymbol{y}) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is related to some notion of 'transportation cost' when moving elements from the support of $\nu_{\mathcal{S}}$ to the support of $\nu_{\mathcal{T}}$ (for Monge, this cost was the cost of moving piles of sand from one place to the other). The transformation $T^{\star}_{\mathcal{S}\to\mathcal{T}}$ that minimizes problem (4.5) is called the 'transportation plan' between distributions $\nu_{\mathcal{S}}$ and $\nu_{\mathcal{T}}$.

When $c(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|^p$, with $p \geq 1$, one may define the p-Wasserstein distance between $\nu_{\mathcal{S}}$ and $\nu_{\mathcal{T}}$ as

$$\mathcal{W}_p(\nu_{\mathcal{S}}, \nu_{\mathcal{T}}) = \left(\min_{T_{\mathcal{S}\to\mathcal{T}}} \int_{\mathbb{R}^n} \|\boldsymbol{x} - T_{\mathcal{S}\to\mathcal{T}}(\boldsymbol{x})\|^p\ \nu_{\mathcal{S}}(\boldsymbol{x})\ \mathrm{d}\boldsymbol{x}\right)^{1/p}\ , \tag{4.7}$$

which is an alternative way of comparing statistical distributions instead of the Kullback-Leibler divergence or the maximum-mean discrepancy.

An interesting particular case for the OT problem is when $\nu_{\mathcal{S}}$ and $\nu_{\mathcal{T}}$ are multivariate Gaussian distributions in $\mathbb{R}^d$, denoted as $\mathcal{N}(\boldsymbol{\mu}_{\mathcal{S}}, \boldsymbol{\Sigma}_{\mathcal{S}})$ and $\mathcal{N}(\boldsymbol{\mu}_{\mathcal{T}}, \boldsymbol{\Sigma}_{\mathcal{T}})$. In this case, the solution for (4.5) is

$$\begin{array}{rccl} T_{\mathcal{S}\to\mathcal{T}} & : & \mathbb{R}^d & \to & \mathbb{R}^d \\ & & \boldsymbol{x} & \to & \boldsymbol{A}(\boldsymbol{x} - \boldsymbol{\mu}_{\mathcal{S}}) + \boldsymbol{\mu}_{\mathcal{T}}\ , \end{array} \tag{4.8}$$

where

$$\boldsymbol{A} = \boldsymbol{\Sigma}_{\mathcal{S}}^{-1/2}(\boldsymbol{\Sigma}_{\mathcal{S}}^{1/2}\boldsymbol{\Sigma}_{\mathcal{T}}\boldsymbol{\Sigma}_{\mathcal{S}}^{1/2})^{1/2}\boldsymbol{\Sigma}_{\mathcal{S}}^{-1/2}\ . \tag{4.9}$$

Note that this reduces to transformation (4.4) in the univariate case. The analytical form of the 2-Wasserstein distance between two multivariate Gaussian distributions is

$$\mathcal{W}_2^2(\nu_{\mathcal{S}}, \nu_{\mathcal{T}}) = \|\boldsymbol{\mu}_{\mathcal{S}} - \boldsymbol{\mu}_{\mathcal{T}}\|^2 + \mathrm{Tr}\Big(\boldsymbol{\Sigma}_{\mathcal{S}} + \boldsymbol{\Sigma}_{\mathcal{T}} - 2(\boldsymbol{\Sigma}_{\mathcal{S}}^{1/2}\boldsymbol{\Sigma}_{\mathcal{T}}\boldsymbol{\Sigma}_{\mathcal{S}}^{1/2})^{1/2}\Big)\ . \tag{4.10}$$

We refer the interested reader to [PC19] for a thorough presentation of several other aspects related to optimal transport, such as existence and unicity properties, the Kantorovich relaxation, numerical solutions, etc.

Procrustes analysis. A very appealing aspect of using a transformation between datasets based on optimal transport is that it makes no assumptions regarding the distributions of the *source* and *target* datasets. Furthermore, when discretizing the problem to real data, the solution of (4.5) is obtained via an optimization problem that is well known in the applied mathematics literature [PC19]. However, due to its lack of assumptions and, consequently, rather large class of possible solutions, optimal transport may be sub-optimal in some cases. For instance, when one has access to information regarding the structure of the datasets to be matched, or regarding how the transformation from one dataset to the other is defined, it might be more convenient to use methods that exploit such knowledge. An important example is Procrustes analysis (PA) [GD04], a tool often employed in statistical shape analysis [Ken89] with applications in text analysis [WM08], protein alignment [WM08], and many other fields. PA considers the distributions of data points in each dataset as shapes in a high-dimensional space. Then, it selects a set of pairs of landmark points from the two different shapes and performs geometrical transformations to get these landmarks as close as possible to each other. Because of its linear nature in the Euclidean case, and the fact that the operations are always global (the same rotation/translation/scaling is applied to all points each time), the space of transformations that one can cover using Procrustes analysis does not include all possible transformations between the statistics of datasets. Nevertheless, the results obtained in several applied settings indicate that the set of transformations applied via PA are rich enough to model the difference in statistics between datasets in many situations [GD04; WM08; Rod+18].

4.2.3 Transfer learning for BCI

When considering data from experiments with brain-computer interfaces (BCI), the *source* and *target* datasets may be from the same subject in different recording sessions (cross-session) or from different subjects (cross-subject). These datasets have often different statistical distributions and many works in the BCI literature have investigated ways of characterizing such mismatches. In [Shi00], the phenomenon responsible for the drift in statistical distributions of two datasets was termed *covariate shift* and modelled by assuming that the distributions of the data points can be different for the *source* and *target* datasets, but the conditional distributions of the labels are the same. Ref. [Sug+07] presented examples on BCI experiments and showed that the *covariate shift* describes well the changes in statistics for this kind of application. In other recent papers, such as [Zan+17] and [Rod+18], the differences between the distributions of points from two datasets were portrayed using nonlinear dimensionality reduction techniques such as those presented in Chapter 3. Figure 4.1 uses the diffusion maps algorithm [LL06] to illustrate the drift in statistics between EEG epochs from different recordings sessions of one same subject.

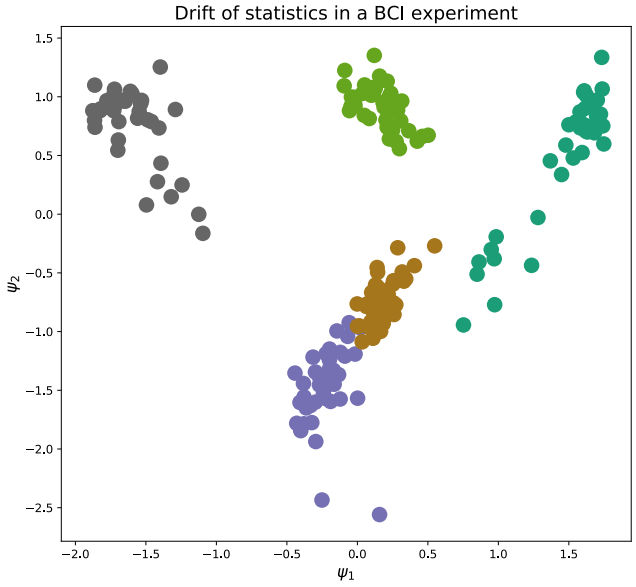

Fig. 4.1: Two-dimensional representation of the spectral embedding obtained via the diffusion maps algorithm applied to the recordings of one subject in the `Cho2017` dataset (see Table 4.1 for a description). Each point corresponds to an EEG epoch and the distances between the data points were calculated via the Riemannian distance between the covariance matrices associated to each epoch. The different colors indicate the experimental sessions related to each epoch.

Traditionally, most methods for transfer learning in BCI are based on two kinds of approach [Lot+18]. One relies on the concept of ensemble classifiers [Bla+08; Faz+09; Con+13; Way+16], where the information from multiple *source* datasets are combined into a “global" classifier, which is then used to label the trials from any other *target* dataset. Another approach uses Bayesian models to describe the variability of the statistics on the *source* datasets and gather information from multiple datasets [Jay+15]. A recent approach that builds upon such Bayesian methods are the works in [Kin+14] and [Hüb+18], which propose a special form of the P300 experimental paradigm to do classification with no calibration.

Recently, some works have used geometrical transformations to match the statistical distributions of two datasets containing EEG recordings. Ref. [Gay+17] applies optimal transport to datasets containing P300 recordings and [Cha+18] uses the same tools on EEG data from sleep recordings. Our work in [Rod+18] is an extension of [Zan+17], which adopts the Riemannian geometric framework for BCI and transforms the data points in the *source* and *target* datasets so that they both have the same geometric mean; [Yai+19] proposes a similar method where the data points are re-centered to the midpoint between the geometric means of the *source* and *target* datasets. Not long after publishing our work on the RPA method in [Rod+18], ref. [Mam+19] proposed an extension for [Yai+19] that introduces a moment alignement step that acts as a rotation on the tangent vectors of the SPD manifold.

It is worth mentioning that distribution matching tries to match as much as possible the information from each pair of *source-target* datasets. Thus, it can be used in addition to an ensembling approach to combine the information from multiple matched-*source* subjects. Also, this kind of approach is paradigm-agnostic and does not rely on any special modification of the experimental setup where the EEG signals are collected (as opposed to [Hüb+18]), a feature that is appealing to a great number of practionners.

4.3 Riemannian Procrustes analysis

In this section, we present a method for matching the statistical distributions of a *source* and a *target* dataset composed of SPD data points. This procedure is a generalization of the classical Procrustes analysis [GD04] to the case when the points to be transformed are defined in the Riemmannian manifold of symmetric positive definite (SPD) matrices, the SPD manifold. Because of its geometric-aware features, the method is called *Riemannian Procrustes analysis* (RPA). To better understand the steps involved in the RPA, the reader is referred to Chapter 2, where a review of properties of the hermitian positive definite (HPD) manifold is presented (and are the same as for the SPD manifold).

RPA can be seen as an evolution of the aforementioned procedures [Zan+17] and [Yai+19], with the re-centering step corresponding to the first of a series of geometrical transformations. Furthermore, [Zan+17] and [Yai+19] are completely unsupervised, since they do not use any information from the labels of the data points, whereas RPA benefits from the labels in the *source* session (which are all known in advance) as well as from (at least part of) the labels that become sequentially available in the *target* session trial after trial.

We begin this section by first formalizing the mathematical context in which the RPA is defined. Then, we introduce the concept of Procrustes analysis on an Euclidean setting and describe how to perform equivalent transformations on the SPD manifold. We justify such operations with the help of a model relating the statistical distributions of the *source* and *target* datasets. This model is the main theoretical contribution of this chapter, since it gives a concrete justification for the geometric operations done in the RPA procedure and allows for a better comprehension of the assumptions that one has to make regarding the statistical distributions of the datasets.

4.3.1 Problem statement

We consider two datasets, the *source* ($\mathcal{S}$) and *target* ($\mathcal{T}$) datasets. They are comprised of couples

$$\begin{aligned} \mathcal{S} &= \left\{ (\boldsymbol{C}_i^{\mathcal{S}},\ \ell_i^{\mathcal{S}}) \text{ for } i = 1, \dots, K_{\mathcal{S}} \right\}, \\ \mathcal{T} &= \left\{ (\boldsymbol{C}_i^{\mathcal{T}},\ \ell_i^{\mathcal{T}}) \text{ for } i = 1, \dots, K_{\mathcal{T}} \right\}, \end{aligned} \tag{4.11}$$

with $\boldsymbol{C}_i^{\mathcal{S}}, \boldsymbol{C}_i^{\mathcal{T}} \in \mathcal{P}(d)$ being data points, and $\ell_i^{\mathcal{S}}, \ell_i^{\mathcal{T}} \in \{1, \dots, L\}$ their corresponding class labels; $K_{\mathcal{S}}$ and $K_{\mathcal{T}}$ are the number of trials in the *source* and *target* sessions respectively. We parametrize the statistical distribution of both datasets as described in Chapter 2, with

$$\Theta_{\mathcal{S}} \sim \left\{ \boldsymbol{M}^{\mathcal{S}}, \boldsymbol{M}_1^{\mathcal{S}}, \dots, \boldsymbol{M}_L^{\mathcal{S}}, \sigma^{\mathcal{S}} \right\} \text{ and } \Theta_{\mathcal{T}} \sim \left\{ \boldsymbol{M}^{\mathcal{T}}, \boldsymbol{M}_1^{\mathcal{T}}, \dots, \boldsymbol{M}_L^{\mathcal{T}}, \sigma^{\mathcal{T}} \right\}, \tag{4.12}$$

where $\boldsymbol{M}^{\mathcal{S}}$ is the geometric mean of all the data points in $\mathcal{S}$, $\boldsymbol{M}_1^{\mathcal{S}}, \dots, \boldsymbol{M}_L^{\mathcal{S}}$ are the geometric means of the points belonging to each class in $\mathcal{S}$, and $\sigma^{\mathcal{S}}$ is the dispersion of the points in $\mathcal{S}$ around $\boldsymbol{M}^{\mathcal{S}}$ (equivalent notation for the parameters in $\Theta_{\mathcal{T}}$). The goal, then, is to define a set of transformations on $\mathcal{S}$ and $\mathcal{T}$ that yields two new datasets, $\mathcal{S}^{(\text{RPA})}$ and $\mathcal{T}^{(\text{RPA})}$, such that distance $\mathcal{W}(\Theta_{\mathcal{S}^{(\text{RPA})}}, \Theta_{\mathcal{T}^{(\text{RPA})}})$ is minimized, where

$$\mathcal{W}^2(\Theta_{\mathcal{X}}, \Theta_{\mathcal{Y}}) = \delta_R^2(\boldsymbol{M}^{\mathcal{X}}, \boldsymbol{M}^{\mathcal{Y}}) + \sum_{\ell=1}^{L} \delta_R^2(\boldsymbol{M}_\ell^{\mathcal{X}}, \boldsymbol{M}_\ell^{\mathcal{Y}}) + \log^2\left(\frac{\sigma^{\mathcal{X}}}{\sigma^{\mathcal{Y}}}\right) \tag{4.13}$$

is a distance between statistical distributions defined in Chapter 2 and δ_R is the natural geodesic distance between two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ in the SPD manifold, given by

$$\delta_R^2(\boldsymbol{A}, \boldsymbol{B}) = \| \log(\boldsymbol{A}^{-1/2} \boldsymbol{B} \boldsymbol{A}^{-1/2}) \|_F^2 . \tag{4.14}$$

We assume the *semi*-supervised transfer learning paradigm [PY10], where one has knowledge of all the labels from the *source* dataset and access to a small subset of labels from the *target* dataset. Put in mathematical terms, we assume knowledge of all the labels from the elements in $\mathcal{S}$ and of a small subset $\mathcal{T}_\ell \subset \mathcal{T}$ with

$$\mathcal{T} = \mathcal{T}_\ell \cup \mathcal{T}_u \quad \text{and} \quad \mathcal{T}_\ell \cap \mathcal{T}_u = \emptyset , \tag{4.15}$$

where ℓ stands for *labeled* and u for *unlabeled*. We further assume that $\mathcal{T}_\ell$ has at least one example from each class. This setup describes well applications where a few labeled calibration points from the *target* dataset can be used to guide the transfer learning procedure. Another relevant case is online algorithms, where labels are available sequentially and augment the $\mathcal{T}_\ell$ dataset after each time step.

4.3.2 An Euclidean motivation

When working with real data, one rarely has access to the actual statistical distributions that generated the data points of the *source* and *target* datasets. Because of this, it is often more interesting to assume that the shapes described by these data points in a high-dimensional space are related to the statistical distributions of each dataset. Then, one can define transformations on the samples so that the shapes of each dataset become as similar as possible. We introduce this approach by first considering the case when the data points are defined in an Euclidean space.

A common method for matching Euclidean geometric shapes is the Procrustes analysis [GD04], which works as follows: suppose we have two sets of landmark points for describing the geometric shapes,

$$\mathcal{S} = \{\boldsymbol{x}_i^{\mathcal{S}} \in \mathbb{R}^n\}_{i=1}^m \text{ and } \mathcal{T} = \{\boldsymbol{x}_i^{\mathcal{T}} \in \mathbb{R}^n\}_{i=1}^m , \tag{4.16}$$

and assume there is a linear relationship relating the m pairs of landmark points as in

$$\boldsymbol{x}_i^{\mathcal{T}} - \boldsymbol{m}^{\mathcal{T}} = s\,\boldsymbol{U}(\boldsymbol{x}_i^{\mathcal{S}} - \boldsymbol{m}^{\mathcal{S}}) , \tag{4.17}$$

where $s \in \mathbb{R}$, $\boldsymbol{m}^{\mathcal{S}}, \boldsymbol{m}^{\mathcal{T}} \in \mathbb{R}^n$, and $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ is an orthogonal matrix. The goal of the PA procedure is to determine the values of $\{s, \boldsymbol{m}^{\mathcal{S}}, \boldsymbol{m}^{\mathcal{T}}, \boldsymbol{U}\}$ so to obtain a new set $\mathcal{X}^{(\text{PA})}$ containing points $\boldsymbol{x}_i^{\mathcal{T}(\text{PA})}$ that matches exactly with $\boldsymbol{x}_i^{\mathcal{S}}$, where

$$\boldsymbol{x}_i^{\mathcal{T}(\text{PA})} - \boldsymbol{m}^{\mathcal{S}} = \frac{1}{s}\boldsymbol{U}^T(\boldsymbol{x}_i^{\mathcal{T}} - \boldsymbol{m}^{\mathcal{T}}) . \tag{4.18}$$

Note that the operations transforming $\boldsymbol{x}_i^{\mathcal{T}}$ can be interpreted as a re-centering to zero (subtracting $\boldsymbol{m}^{\mathcal{T}}$) followed by a stretching or compression (division by s), and a rotation (multiplication by $\boldsymbol{U}^T$); the final re-centering to $\boldsymbol{m}^{\mathcal{S}}$ is optional, since it is often more interesting to re-center the data points in both datasets to the origin and consider only zero-mean shapes.

To apply Procrustes analysis to point clouds of two datasets coming from different statistical distributions, one first has to decide what are the landmarks to consider. A reasonable choice is to assume a linear relationship between the means of all the points from the dataset as well as the means of each class. Figure 4.2 illustrates the results of Procrustes analysis applied to a *source* and a *target* dataset defined in $\mathbb{R}^2$ and whose landmarks are based on the means of the datasets. We observe that a linear classifier trained on the *source* dataset plotted in Figure 4.2A would clearly fail in inferring the labels of the data points in the *target* dataset, as opposed to the matched case in Figure 4.2D.

In the next section, we show how to adapt the traditional Procrustes analysis to the case where the data points are defined in the SPD manifold.

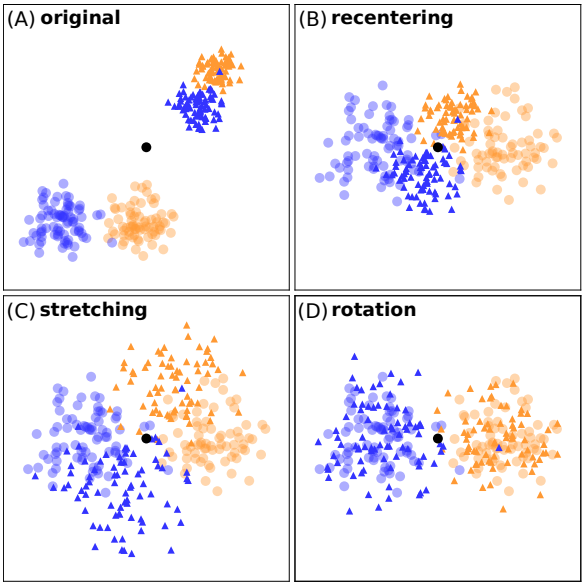

Fig. 4.2: Illustration of the sequence of operations of a Procrustes analysis applied to a dataset consisting of two-dimensional Euclidean vectors (better visualized with colors). The data was simulated with a mixture of two gaussians for each dataset. Each point on the scatter plot represents a data point from the *source* dataset (circles) or the *target* dataset (triangles). The colors blue and yellow indicate the classes of the data points, whereas the black dot is the origin. (A) Distribution of the data points in *source* and *target* datasets as they are originally available and (B) after re-centering their means to zero. In (C) the distribution after the stretching operation and (D) after the rotation.

4.3.3 Transformations in the SPD manifold

In order to apply Procrustes analysis on SPD data points, we have to adapt the steps of re-centering, stretching and rotation according to the intrinsic geometry of $\mathcal{P}(d)$. We call such procedure *Riemannian Procrustes analysis* (RPA) and describe its steps here below.

Re-center to identity. In $\mathcal{P}(d)$, the Identity matrix plays the role of the origin of the space. Therefore, the first step of RPA is to transform the matrices in $\mathcal{S}$ and $\mathcal{T}$ so they are both centered around $\boldsymbol{I}_d$. This amounts to the transformation proposed in [Zan+17] if the covariance matrices used to describe the resting activity of each session were chosen to be the geometric mean of the trials of each dataset.

Due to the affine-invariance of the geodesic distance in the SPD manifold, the geometric mean of a set of re-centered matrices,

$$\boldsymbol{C}_i^{\mathcal{S}(\mathrm{rct})} = \left(\boldsymbol{M}^{\mathcal{S}}\right)^{-1/2} \boldsymbol{C}_i^{\mathcal{S}} \left(\boldsymbol{M}^{\mathcal{S}}\right)^{-1/2}, \tag{4.19}$$

is $\boldsymbol{I}_d$. Moreover, since $\hat{\boldsymbol{M}}^{\mathcal{T}}$ is estimated from a subset of points in $\mathcal{T}_\ell \subset \mathcal{T}$, the geometric mean of the set of matrices

$$\boldsymbol{C}_i^{\mathcal{T}(\text{rct})} = (\hat{\boldsymbol{M}}^{\mathcal{T}})^{-1/2} \boldsymbol{C}_i^{\mathcal{T}} (\hat{\boldsymbol{M}}^{\mathcal{T}})^{-1/2} \tag{4.20}$$

is approximately the identity matrix (it tends to the identity as the number of elements in $\mathcal{T}_\ell$ grows). We have then two new datasets consisting of re-centered (rct) matrices

$$\begin{aligned} \mathcal{S}^{(\text{rct})} &= \left\{ (\boldsymbol{C}_i^{\mathcal{S}(\text{rct})}, \ell_i^{\mathcal{S}}) \text{ for } i = 1, \ldots, K_{\mathcal{S}} \right\}, \\ \mathcal{T}^{(\text{rct})} &= \left\{ (\boldsymbol{C}_i^{\mathcal{T}(\text{rct})}, \ell_i^{\mathcal{T}}) \text{ for } i = 1, \ldots, K_{\mathcal{T}} \right\}, \end{aligned} \tag{4.21}$$

with the indices of the partition $\mathcal{T}^{(\text{rct})} = \mathcal{T}_\ell^{(\text{rct})} \cup \mathcal{T}_u^{(\text{rct})}$ being the same as in (4.11).

Equalize dispersions. The next step of RPA consists in rescaling the distributions on both datasets so that their dispersions around the mean are the same. To do so, we can see from the expression of the AIRM distance that

$$\delta_R^2\Big((\boldsymbol{C}_i^{\mathcal{T}(\text{rct})})^s, \boldsymbol{I}_d \Big) = s^2 \, \delta_R^2 \left(\boldsymbol{C}_i^{\mathcal{T}(\text{rct})}, \boldsymbol{I}_d \right) , \tag{4.22}$$

which implies that one can modulate the dispersion of $\mathcal{T}^{(\text{rct})}$ by simply moving each of its matrices along the geodesic that links it to the identity matrix. Note that the parameter s plays the same role as the scaling factor in (4.17). We match the dispersions from *source* and *target* by defining new stretched (str) data points

$$\boldsymbol{C}_i^{\mathcal{T}(\text{str})} = \left(\boldsymbol{C}_i^{\mathcal{T}(\text{rct})} \right)^s , \tag{4.23}$$

where we require $s \in \mathbb{R}$ to verify

$$s = \sigma^{\mathcal{S}} / \hat{\sigma}^{\mathcal{T}} \tag{4.24}$$

and $\hat{\sigma}^{\mathcal{T}} \simeq \sigma^{\mathcal{T}}$ is estimated from data points in $\mathcal{T}_\ell$. We may then define two new datasets

$$\begin{aligned} \mathcal{S}^{(\text{str})} &= \left\{ (\boldsymbol{C}_i^{\mathcal{S}(\text{rct})}, \ell_i^{\mathcal{S}}) \text{ for } i = 1, \ldots, K_{\mathcal{S}} \right\}, \\ \mathcal{T}^{(\text{str})} &= \left\{ (\boldsymbol{C}_i^{\mathcal{T}(\text{str})}, \ell_i^{\mathcal{T}}) \text{ for } i = 1, \ldots, K_{\mathcal{T}} \right\}, \end{aligned} \tag{4.25}$$

where we note that the SPD matrices for the re-centered *source* dataset do not change after the stretching step.

Note that the re-centering of matrices in Step 1 does not alter the dispersion of the matrices around their geometric mean, which means that the stretching step could have been done before re-centering the matrices in $\mathcal{T}$. However, in this case, the

geodesic move in (4.23) would have to be done with respect to $\boldsymbol{M}^{\mathcal{T}}$, that is, we would have to use a more involved relation

$$\boldsymbol{C}_i^{\mathcal{T}(\text{str})} = \left(\boldsymbol{M}^{\mathcal{T}}\right)^{1/2}\left((\boldsymbol{M}^{\mathcal{T}})^{-1/2}\boldsymbol{C}_i^{\mathcal{T}}(\boldsymbol{M}^{\mathcal{T}})^{-1/2}\right)^{s}\left(\boldsymbol{M}^{\mathcal{T}}\right)^{1/2} . \tag{4.26}$$

Up to this point, no information from the trials' classes has been used. We say then that the *re-centering* and *stretching* operations form the *unsupervised* part of the RPA method.

Rotate. The last step of RPA consists of rotating the matrices from $\mathcal{T}^{(\text{str})}$ around the origin and matching the orientation of its point cloud with that of $\mathcal{S}^{(\text{str})}$ (see Figure 4.3D). To do so, we note that if $\boldsymbol{U}$ is an orthogonal matrix, then

$$\delta_R^2(\boldsymbol{U}^T\boldsymbol{C}_i^{\mathcal{T}(\text{str})}\boldsymbol{U}, \boldsymbol{I}_d) = \delta_R^2(\boldsymbol{U}^T\boldsymbol{C}_i^{\mathcal{T}(\text{str})}\boldsymbol{U}, \boldsymbol{U}^T\boldsymbol{U}) = \delta_R^2(\boldsymbol{C}_i^{\mathcal{T}(\text{str})}, \boldsymbol{I}_d) , \tag{4.27}$$

where the last equality is due to the affine-invariance property of δ_R. This result means that the effect of an orthogonal matrix over a set of matrices centered at the identity is that of a rotation around their mean. We form a new dataset $\mathcal{T}^{(\text{rot})}$ containing rotated (rot) matrices with

$$\boldsymbol{C}_i^{\mathcal{T}(\text{rot})} = \boldsymbol{U}^T\, \boldsymbol{C}_i^{\mathcal{T}(\text{str})}\, \boldsymbol{U} , \tag{4.28}$$

where $\boldsymbol{U}$ is an orthogonal matrix to be determined from the data. By the end of the RPA procedure, we have two transformed versions of $\mathcal{S}$ and $\mathcal{T}$,

$$\begin{aligned} \mathcal{S}^{(\text{RPA})} &= \left\{(\boldsymbol{C}_i^{\mathcal{S}(\text{rct})}, \ell_i^{\mathcal{S}}) \text{ for } i = 1, \ldots, K_{\mathcal{S}}\right\} , \\ \mathcal{T}^{(\text{RPA})} &= \left\{(\boldsymbol{C}_i^{\mathcal{T}(\text{rot})}, \ell_i^{\mathcal{T}}) \text{ for } i = 1, \ldots, K_{\mathcal{T}}\right\} . \end{aligned} \tag{4.29}$$

As we will see next, matrix $\boldsymbol{U}$ is determined using the labels from the trials, so we say it corresponds to the *supervised* part of the RPA.

The orthogonal matrix $\boldsymbol{U}$. The procedure to determine the matrix $\boldsymbol{U}$ comes up naturally once the assumptions of the RPA method are written in mathematical form. For simplicity, we will first assume that $\mathcal{T}_\ell = \mathcal{T}$. We consider the geometric means of the *source* and *target* datasets as landmarks to be matched, so one can write a relation between the full geometric mean of the datasets,

$$\boldsymbol{M}^{\mathcal{T}} = \boldsymbol{A}\, \boldsymbol{M}^{\mathcal{S}}\, \boldsymbol{A}^T , \tag{4.30}$$

and between the class means of the datasets,

$$\boldsymbol{M}_\ell^{\mathcal{T}} = \boldsymbol{A}\, \boldsymbol{M}_\ell^{\mathcal{S}}\, \boldsymbol{A}^T , \ \ell \in \{1, \ldots, L\} , \tag{4.31}$$

where $\boldsymbol{M}^{\mathcal{S}}, \boldsymbol{M}^{\mathcal{T}}, \boldsymbol{M}_{\ell}^{\mathcal{S}}, \boldsymbol{M}_{\ell}^{\mathcal{T}}$ are all defined in Section 4.3.1, and $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is an unknown invertible matrix that models the discrepancies between the statistics of the *source* and *target* datasets. We can rewrite the relation in (4.30) as

$$\left((\boldsymbol{M}^{\mathcal{T}})^{1/2}(\boldsymbol{M}^{\mathcal{T}})^{1/2}\right) = \boldsymbol{A}\left((\boldsymbol{M}^{\mathcal{S}})^{1/2}(\boldsymbol{M}^{\mathcal{S}})^{1/2}\right)\boldsymbol{A}^{T}\,, \tag{4.32}$$

$$\boldsymbol{I}_d = (\boldsymbol{M}^{\mathcal{T}})^{-1/2}\boldsymbol{A}\left((\boldsymbol{M}^{\mathcal{S}})^{1/2}(\boldsymbol{M}^{\mathcal{S}})^{1/2}\right)\boldsymbol{A}^{T}(\boldsymbol{M}^{\mathcal{T}})^{-1/2}\,, \tag{4.33}$$

$$\boldsymbol{I}_d = \left((\boldsymbol{M}^{\mathcal{T}})^{-1/2}\boldsymbol{A}(\boldsymbol{M}^{\mathcal{S}})^{1/2}\right)\left((\boldsymbol{M}^{\mathcal{T}})^{-1/2}\boldsymbol{A}(\boldsymbol{M}^{\mathcal{S}})^{1/2}\right)^{T}\,, \tag{4.34}$$

$$\boldsymbol{U}\boldsymbol{U}^{T} = \left((\boldsymbol{M}^{\mathcal{T}})^{-1/2}\boldsymbol{A}(\boldsymbol{M}^{\mathcal{S}})^{1/2}\right)\left((\boldsymbol{M}^{\mathcal{T}})^{-1/2}\boldsymbol{A}(\boldsymbol{M}^{\mathcal{S}})^{1/2}\right)^{T}\,, \tag{4.35}$$

where $\boldsymbol{U}$ is the $n \times n$ orthogonal matrix that we want to determine. Matrix $\boldsymbol{U}$ can then be simply written as

$$\boldsymbol{U} = (\boldsymbol{M}^{\mathcal{T}})^{-1/2}\,\boldsymbol{A}\,(\boldsymbol{M}^{\mathcal{S}})^{1/2}\,, \tag{4.36}$$

where $\boldsymbol{M}^{\mathcal{S}}$ and $\boldsymbol{M}^{\mathcal{T}}$ are directly estimated from the data points, and $\boldsymbol{A}$ remains unknown. To determine an expression for $\boldsymbol{U}$ only in terms of variables that can be estimated from the data, we use (4.36) in (4.31) to get

$$\boldsymbol{M}_{\ell}^{\mathcal{T}} = \left((\boldsymbol{M}^{\mathcal{T}})^{1/2}\boldsymbol{U}(\boldsymbol{M}^{\mathcal{S}})^{-1/2}\right)\boldsymbol{M}_{\ell}\left((\boldsymbol{M}^{\mathcal{T}})^{1/2}\boldsymbol{U}(\boldsymbol{M}^{\mathcal{S}})^{-1/2}\right)^{T}, \tag{4.37}$$

$$(\boldsymbol{M}^{\mathcal{T}})^{-1/2}\boldsymbol{M}_{\ell}^{\mathcal{T}}(\boldsymbol{M}^{\mathcal{T}})^{-1/2} = \boldsymbol{U}\,(\boldsymbol{M}^{\mathcal{S}})^{-1/2}\boldsymbol{M}_{\ell}^{\mathcal{S}}(\boldsymbol{M}^{\mathcal{S}})^{-1/2}\,\boldsymbol{U}^{T}\,. \tag{4.38}$$

Defining the matrices

$$\boldsymbol{G}_{\ell}^{\mathcal{S}} = (\boldsymbol{M}^{\mathcal{S}})^{-1/2}\boldsymbol{M}_{\ell}^{\mathcal{S}}(\boldsymbol{M}^{\mathcal{S}})^{-1/2} \tag{4.39}$$

and

$$\boldsymbol{G}_{\ell}^{\mathcal{T}} = (\boldsymbol{M}^{\mathcal{T}})^{-1/2}\boldsymbol{M}_{\ell}^{\mathcal{T}}(\boldsymbol{M}^{\mathcal{T}})^{-1/2}\,, \tag{4.40}$$

we have

$$\underbrace{(\boldsymbol{M}^{\mathcal{T}})^{-1/2}\boldsymbol{M}_{\ell}^{\mathcal{T}}(\boldsymbol{M}^{\mathcal{T}})^{-1/2}}_{\boldsymbol{G}_{\ell}^{\mathcal{T}}} = \boldsymbol{U}\,\underbrace{(\boldsymbol{M}^{\mathcal{S}})^{-1/2}\boldsymbol{M}_{\ell}^{\mathcal{S}}(\boldsymbol{M}^{\mathcal{S}})^{-1/2}}_{\boldsymbol{G}_{\ell}^{\mathcal{S}}}\,\boldsymbol{U}^{T}\,, \tag{4.41}$$

$$\boldsymbol{G}_{\ell}^{\mathcal{T}} = \boldsymbol{U}\,\boldsymbol{G}_{\ell}^{\mathcal{S}}\,\boldsymbol{U}^{T}\,. \tag{4.42}$$

It is worth noting that $\boldsymbol{G}_{\ell}^{\mathcal{T}}$ and $\boldsymbol{G}_{\ell}^{\mathcal{S}}$ have the same eigenvalues. To see this, we can use the expression in (4.30) for $\boldsymbol{M}^{\mathcal{T}}$ to rewrite (4.38) as

$$\boldsymbol{G}_{\ell}^{\mathcal{T}} = (\boldsymbol{M}^{\mathcal{T}})^{1/2}\boldsymbol{A}^{-T}(\boldsymbol{M}^{\mathcal{S}})^{-1/2}\,\boldsymbol{G}_{\ell}^{\mathcal{S}}\,(\boldsymbol{M}^{\mathcal{S}})^{1/2}\boldsymbol{A}^{T}(\boldsymbol{M}^{\mathcal{T}})^{-1/2}\,, \tag{4.43}$$

$$\boldsymbol{G}_{\ell}^{\mathcal{T}} = \left((\boldsymbol{M}^{\mathcal{T}})^{1/2}\boldsymbol{A}^{-T}(\boldsymbol{M}^{\mathcal{S}})^{-1/2}\right)\boldsymbol{G}_{\ell}^{\mathcal{S}}\left((\boldsymbol{M}^{\mathcal{T}})^{1/2}\boldsymbol{A}^{-T}(\boldsymbol{M}^{\mathcal{S}})^{-1/2}\right)^{-1}, \tag{4.44}$$

and conclude that $\boldsymbol{G}_\ell^{\mathcal{S}}$ and $\boldsymbol{G}_\ell^{\mathcal{T}}$ are related via a similarity transform and, therefore, have the same set of eigenvalues. We can then write the eigendecompositions

$$\boldsymbol{G}_\ell^{\mathcal{S}} = (\boldsymbol{Q}_\ell^{\mathcal{S}})\boldsymbol{\Lambda}(\boldsymbol{Q}_\ell^{\mathcal{S}})^T \quad \text{and} \quad \boldsymbol{G}_\ell^{\mathcal{T}} = (\boldsymbol{Q}_\ell^{\mathcal{T}})\boldsymbol{\Lambda}(\boldsymbol{Q}_\ell^{\mathcal{T}})^T \,, \tag{4.45}$$

so that (4.38) becomes

$$(\boldsymbol{Q}_\ell^{\mathcal{T}})\boldsymbol{\Lambda}(\boldsymbol{Q}_\ell^{\mathcal{T}})^T = \boldsymbol{U}\ (\boldsymbol{Q}_\ell^{\mathcal{S}})\boldsymbol{\Lambda}(\boldsymbol{Q}_\ell^{\mathcal{S}})^T\ \boldsymbol{U}^T. \tag{4.46}$$

Solving (4.46) for $\boldsymbol{U}$ we obtain, for any $\ell \in \{1, \dots, L\}$ (where L is the number of classes),

$$\boldsymbol{U} = (\boldsymbol{Q}_\ell^{\mathcal{T}})(\boldsymbol{Q}_\ell^{\mathcal{S}})^T \,, \tag{4.47}$$

which is ultimately an expression for the rotation matrix in terms of quantities that can be directly estimated from the dataset.

Note that (4.47) is also the solution to the following optimization problem

$$\underset{\boldsymbol{U}^T\boldsymbol{U}=\boldsymbol{I}_d}{\text{minimize}} \quad \delta_R^2\left(\boldsymbol{G}_\ell^{\mathcal{T}}, \boldsymbol{U}\boldsymbol{G}_\ell^{\mathcal{S}}\boldsymbol{U}^T\right) \,, \tag{4.48}$$

for any $\ell \in \{1, \dots, L\}$, and so it can be interpreted as the orthogonal matrix that acts to minimize the distance between a modified version of the class means of the *source* and *target* datasets. Interestingly, Ref. [BC19] has shown that problem (5.42) has the same solution when considering many other distances between symmetric positive definite matrices, such as the Frobenius distance, the Bures-Wasserstein distance, and the Bhattacharyya divergence.

Determining U from data. Until now, we have assumed that $\mathcal{T}_\ell = \mathcal{T}$. In practice, however, we have $\mathcal{T}_\ell \subset \mathcal{T}$, so the estimation of the class means of the *target* dataset are only approximations of the real class means of the statistical distribution. Because of this, instead of giving preference to a particular noisy estimate of a class mean to determine $\boldsymbol{U}$ via (4.47), we prefer to obtain it as a solution to the following optimization problem on the manifold of orthogonal matrices:

$$\underset{\boldsymbol{U}^T\boldsymbol{U}=\boldsymbol{I}_n}{\text{minimize}} \quad \sum_{\ell=1}^{L} \delta_R^2\left(\boldsymbol{G}_\ell^{\mathcal{T}}, \boldsymbol{U}\boldsymbol{G}_\ell^{\mathcal{S}}\boldsymbol{U}^T\right) \,. \tag{4.49}$$

We solve (4.49) using a special form of the steepest-descent algorithm adapted for optimization procedures on manifolds, as described in [Abs+09]. To do so, we first rewrite each term of the cost function in (4.49) as

$$\mathcal{L}(\boldsymbol{U}) = \sum_{\ell=1}^{L} f_\ell(\boldsymbol{U}) \ \text{ with } \ f_\ell(\boldsymbol{U}) = \delta_R^2(\boldsymbol{G}_\ell^{\mathcal{T}},\ \boldsymbol{U}\boldsymbol{G}_\ell^{\mathcal{S}}\boldsymbol{U}^T) \tag{4.50}$$

and express its Jacobian as

$$D_{\boldsymbol{U}}\mathcal{L}(\boldsymbol{U}) = \sum_{\ell=1}^{L} D_{\boldsymbol{U}} f_\ell(\boldsymbol{U}) \ , \tag{4.51}$$

with

$$D_{\boldsymbol{U}} f_\ell(\boldsymbol{U}) = 4 \log \left(\boldsymbol{G}_\ell^{\mathcal{T}} \boldsymbol{U} \boldsymbol{G}_\ell^{\mathcal{S}} \boldsymbol{U}^T \right) \boldsymbol{U} \ , \tag{4.52}$$

where the derivative of the AIRM distance was obtained from [Moa05]. On each iteration of the gradient descent procedure, the vector $D_{\boldsymbol{U}} f_\ell(\boldsymbol{U})$ is projected onto the tangent space of the manifold of orthogonal matrices (see [Abs+09] for details). We used the `pymanopt` package [Tow+16] for carrying out the optimization procedure.

It is worth noting that in some cases the numerical minimization of problem (4.49) via gradient descent may be computationally costly, specially when the SPD matrices involved in the operations are big. An alternative solution is to rewrite (4.49) using the Frobenius distance instead of δ_R, as in

$$\underset{\boldsymbol{U}^T\boldsymbol{U}=\boldsymbol{I}_n}{\text{minimize}} \quad \sum_{\ell=1}^{L} \| \boldsymbol{G}_\ell^{\mathcal{T}} - \boldsymbol{U} \boldsymbol{G}_\ell^{\mathcal{S}} \boldsymbol{U}^T \|^2 \ . \tag{4.53}$$

In practice, we have observed that the classification performance of pipelines using a rotation matrix estimated via the optimization problem in (4.53) is equivalent to that when we use a rotation matrix from (4.49). Intuitively, we believe that this comes from the fact that, as mentioned before, the solution for (4.49) and (4.53) are the same when $L = 1$.

4.3.4 Summary of the RPA method

Algorithm 2 recapitulates the steps of a classification task using RPA for matching the statistical distributions of the *source* and *target* datasets.

4.3.5 A statistical interpretation of the steps in RPA

We give now an interpretation of the steps of RPA in terms of the statistical distributions of the datasets. Without loss of generality, we will consider that $\sigma^{\mathcal{T}} = \sigma^{\mathcal{S}}$, since the dispersions can always be made equal prior to the transformations. We will also assume that $\mathcal{T}_\ell = \mathcal{T}$ for simplicity of exposition.

The relations in (4.30) and (4.31) define which landmark data points we should match in the RPA procedure, an approach that is justified from the fact that we parametrize the statistics of $\mathcal{S}$ and $\mathcal{T}$ using their geometric means, as described in Section 4.3.1. From this observation, one can also conclude that a simple approach

Algorithm 2: Transfer Learning via RPA

Input: $\mathcal{S}$, $\mathcal{T}_\ell$ and $\mathcal{T}_u$ as defined in (4.11) and (4.15)
Output: accuracy of classification on $\mathcal{T}_u$

1 Estimate $\boldsymbol{M}^{\mathcal{S}}$ and $\boldsymbol{M}^{\mathcal{T}}$ from the data in $\mathcal{S}$ and $\mathcal{T}_\ell$

2 Re-center the matrices in $\mathcal{S}$ and $\mathcal{T}$ using (4.19) and (4.20), and form new datasets

$$\mathcal{S}^{(\text{rct})} \text{ and } \mathcal{T}^{(\text{rct})} = \mathcal{T}_\ell^{(\text{rct})} \cup \mathcal{T}_u^{(\text{rct})}$$

3 Calculate the ratio of dispersions in $\mathcal{S}^{(\text{rct})}$ and $\mathcal{T}_\ell^{(\text{rct})}$ as in (4.24) and use it to form the new datasets

$$\mathcal{S}^{(\text{str})} \text{ and } \mathcal{T}^{(\text{str})} = \mathcal{T}_\ell^{(\text{str})} \cup \mathcal{T}_u^{(\text{str})}$$

with matrices as described in (4.23)

4 Estimate matrices $\boldsymbol{M}_\ell^{\mathcal{S}}$ and $\boldsymbol{M}_\ell^{\mathcal{T}}$ for $\ell \in \{1, \ldots, L\}$ and obtain the orthogonal matrix $\boldsymbol{U}$ as a solution from (4.49)

5 Rotate the matrices from $\mathcal{T}^{(\text{str})}$ as in (4.28) and obtain

$$\mathcal{S}^{(\text{RPA})} \text{ and } \mathcal{T}^{(\text{RPA})} = \mathcal{T}_\ell^{(\text{RPA})} \cup \mathcal{T}_u^{(\text{RPA})}$$

6 Form the training dataset for a classifier with

$$\mathcal{D}_{\text{train}} = \mathcal{S}^{(\text{RPA})} \cup \mathcal{T}_\ell^{(\text{RPA})}$$

and get the accuracy of classification on the data points from the test dataset

$$\mathcal{D}_{\text{test}} = \mathcal{T}_u^{(\text{RPA})}$$

for matching the statistical distributions of $\mathcal{S}$ and $\mathcal{T}$ would be to estimate matrix $\boldsymbol{A}$ from the available data points and apply $\boldsymbol{A}^{-1}$ to all the elements of $\mathcal{T}$, as in

$$\boldsymbol{C}_i^{\mathcal{T}} \mapsto \boldsymbol{A}^{-1}\boldsymbol{C}_i^{\mathcal{S}}\boldsymbol{A}^{-T} \,. \tag{4.54}$$

From (4.36) we can write

$$\boldsymbol{A} = (\boldsymbol{M}^{\mathcal{T}})^{1/2}\boldsymbol{U}(\boldsymbol{M}^{\mathcal{S}})^{-1/2} \,, \tag{4.55}$$

where $\boldsymbol{M}^{\mathcal{S}}$ and $\boldsymbol{M}^{\mathcal{T}}$ are estimated directly from the dataset, and the orthogonal matrix $\boldsymbol{U}$ is determined as discussed in Section 4.3.3. Applying $\boldsymbol{A}^{-1}$ to the matrices in $\mathcal{T}$, we get

$$\boldsymbol{A}^{-1}\boldsymbol{C}_i^{\mathcal{T}}\boldsymbol{A}^{-T} = \left(\boldsymbol{M}^{\mathcal{S}}\right)^{1/2}\left[\boldsymbol{U}^T\left(\left(\boldsymbol{M}^{\mathcal{T}}\right)^{-1/2}\boldsymbol{C}_i^{\mathcal{T}}\left(\boldsymbol{M}^{\mathcal{T}}\right)^{-1/2}\right)\boldsymbol{U}\right]\left(\boldsymbol{M}^{\mathcal{T}}\right)^{1/2} \,, \tag{4.56}$$

which describes the same steps of RPA: re-center to identity, stretch with $s = 1$ (since dispersions are the same) and rotate, followed by a translation of the mean back to $\boldsymbol{M}^{\mathcal{S}}$. From the expressions above, we see that the sequence of operations in RPA are nicely justified by the assumptions of our statistical model for the data points.

4.3.6 Relation with optimal transport

The transformations defined by RPA are the same for all data points (re-center, stretch, and rotation) and they minimize the distance $\mathcal{W}$ between the statistical distributions of the *source* and *target* datasets. As mentioned before, these operations are justified by the way that we parametrize the statistics of the two datasets, that is, as mixtures of Riemannian Gaussians in the SPD manifold.

Remember that, in the case of Euclidean data points, the transport plan that minimizes the Monge problem for two multivariate Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}_{\mathcal{S}}, \boldsymbol{\Sigma}_{\mathcal{S}})$ and $\mathcal{N}(\boldsymbol{\mu}_{\mathcal{T}}, \boldsymbol{\Sigma}_{\mathcal{T}})$ is given by

$$\begin{aligned} T_{\mathcal{S}\to\mathcal{T}} \;:\; \mathbb{R}^d &\to \mathbb{R}^d \\ \boldsymbol{x} &\to \boldsymbol{A}(\boldsymbol{x} - \boldsymbol{\mu}_{\mathcal{S}}) + \boldsymbol{\mu}_{\mathcal{T}} \,, \end{aligned} \tag{4.57}$$

where

$$\boldsymbol{A} = \boldsymbol{\Sigma}_{\mathcal{S}}^{-1/2}(\boldsymbol{\Sigma}_{\mathcal{S}}^{1/2}\boldsymbol{\Sigma}_{\mathcal{T}}\boldsymbol{\Sigma}_{\mathcal{S}}^{1/2})^{1/2}\boldsymbol{\Sigma}_{\mathcal{S}}^{-1/2} \,. \tag{4.58}$$

This can be interpreted as a re-centering of points in $\mathcal{S}$ to the origin (subtraction of $\boldsymbol{\mu}_{\mathcal{S}}$), a rotation and stretching (action of the matrix $\boldsymbol{A}$) and a re-centering to the new mean (adding $\boldsymbol{\mu}_{\mathcal{T}}$). We see, then, that the three rigid transformations used in Procrustes analysis in the Euclidean space are in fact the solution to an optimal transport problem between two Gaussian distributions. An extension for mixtures of Gaussian distributions is a non-trivial problem and has been studied recently in

the literature [Che+19], but no explicit solution for the transport plan between two such distributions has been determined. We conjecture that the three geometric operations done in RPA may be seen as analogous to those that solve the optimal transport problem for Gaussian distributions in the Euclidean space but for statistical distributions defined in the SPD manifold.

Unfortunately, optimal transport between statistical distributions defined in the SPD manifold has not gained much interest in the literature yet, so confirming (or disproving) our claim remains an open question. We refer the interested reader to [McC01], where it was shown that a diffeomorphism (i.e., a bijective map) connecting two probability distributions, $\nu_{\mathcal{S}}$ and $\nu_{\mathcal{T}}$, defined in a general Riemannian manifold $\mathcal{M}$ (e.g., the SPD manifold) may be factored as the composition of a volume-preserving map (e.g., an unitary matrix in the SPD manifold) and a map which is the transport plan solving the optimal transport (OT) problem relating $\nu_{\mathcal{S}}$ and $\nu_{\mathcal{T}}$; the cost function of this OT problem is the square of the geodesic distance in $\mathcal{M}$. Recently, [Yai+19] has used this result to justify their extension of the domain adaptation via optimal transport proposed in [Cou+17] to the case when the data points are defined in a SPD manifold.

4.3.7 A time series interpretation of the steps in RPA

A relevant application of the Riemannian geometric framework of SPD matrices is when the data points parametrize the statistics of multivariate time series, as described in Chapter 2. In this case, each SPD matrix $\boldsymbol{C} \in \mathcal{P}(d)$ represents the spatial covariance matrix of a time series $\boldsymbol{x}(n) \in \mathbb{R}^d$ and the operations on the SPD manifold can be interpreted as transformations over the dimensions of the time series. We will now examine what are the interpretations of the steps in RPA under this point of view. As in Section 4.3.5, we assume that $\sigma^{\mathcal{S}} = \sigma^{\mathcal{T}}$.

We consider two zero-mean multivariate time series, $\boldsymbol{x}^{\mathcal{S}}(t)$ and $\boldsymbol{x}^{\mathcal{T}}(t)$, which we say are 'representative' examples of datasets $\mathcal{S}$ and $\mathcal{T}$ in the sense that

$$\boldsymbol{M}^{\mathcal{S}} = \mathbb{E}\left[\boldsymbol{x}^{\mathcal{S}}(t)\boldsymbol{x}^{\mathcal{S}}(t)^T\right] \quad \text{and} \quad \boldsymbol{M}^{\mathcal{T}} = \mathbb{E}\left[\boldsymbol{x}^{\mathcal{T}}(t)\boldsymbol{x}^{\mathcal{T}}(t)^T\right] \ . \tag{4.59}$$

Rewriting relation (4.30) in terms of these time series we have that

$$\mathbb{E}\left[\boldsymbol{x}^{\mathcal{T}}(t)\boldsymbol{x}^{\mathcal{T}}(t)^T\right] = \boldsymbol{A}\,\mathbb{E}\left[\boldsymbol{x}^{\mathcal{S}}(t)\boldsymbol{x}^{\mathcal{S}}(t)^T\right]\,\boldsymbol{A}^T \ , \tag{4.60}$$

$$\mathbb{E}\left[\boldsymbol{x}^{\mathcal{T}}(t)\boldsymbol{x}^{\mathcal{T}}(t)^T\right] = \mathbb{E}\left[\boldsymbol{A}\boldsymbol{x}^{\mathcal{S}}(t)\left(\boldsymbol{A}\boldsymbol{x}^{\mathcal{S}}(t)\right)^T\right] \ , \tag{4.61}$$

and, by inspection,

$$\boldsymbol{x}^{\mathcal{T}}(t) = \boldsymbol{A}\boldsymbol{x}^{\mathcal{S}}(t) \ . \tag{4.62}$$

This means that RPA assumes $\boldsymbol{x}^{\mathcal{T}}(t)$ and $\boldsymbol{x}^{\mathcal{S}}(t)$ may be linearly related via a mixing matrix $\boldsymbol{A}$. Expanding the expression for $\boldsymbol{A}$ with (4.55), we also observe that

$$\boldsymbol{x}^{\mathcal{T}}(t) = (\boldsymbol{M}^{\mathcal{T}})^{1/2}\boldsymbol{U}(\boldsymbol{M}^{\mathcal{S}})^{-1/2}\boldsymbol{x}^{\mathcal{S}}(t)\,, \tag{4.63}$$

which can be interpreted as a whitening step applied to $\boldsymbol{x}^{\mathcal{S}}(t)$, followed by the action of an orthogonal matrix (which could, for instance, be a permutation matrix that reorders the dimensions of the time series on one dataset so that they are aligned with the dimensions of another dataset) and, then, a de-whitening step to make the new time series have covariance matrix $\boldsymbol{M}^{\mathcal{T}}$.

4.3.8 RPA as an optimization problem

As mentioned before (see Section 4.3.1), the goal of RPA is to minimize the $\mathcal{W}$ distance between the statistical distributions of the *source* and *target* datasets. Supposing that the two datasets have the same dispersion ($\sigma^{\mathcal{S}} = \sigma^{\mathcal{T}}$), one can rewrite this objective function as an optimization problem

$$\min_{\boldsymbol{B}\in GL_d(\mathbb{R})} \left(\delta_R^2(\boldsymbol{M}^{\mathcal{S}}, \boldsymbol{B}\boldsymbol{M}^{\mathcal{T}}\boldsymbol{B}^T) + \sum_{\ell=1}^{L}\delta_R^2(\boldsymbol{M}_\ell^{\mathcal{S}}, \boldsymbol{B}\boldsymbol{M}_\ell^{\mathcal{T}}\boldsymbol{B}^T)\right)\,, \tag{4.64}$$

where $GL_d(\mathbb{R})$ is the set of invertible $d \times d$ real matrices. The solution $\boldsymbol{B}^\star$ of (4.64) is then used to transform the data points in the *target* dataset, $\mathcal{T}$, and make their statistical distribution similar to that of the *source* dataset, $\mathcal{S}$.

The usual approach for solving (4.3.5) would be to use some gradient descent technique to minimize its cost function. However, by assuming that there exists some matrix $\boldsymbol{A} \in GL_d(\mathbb{R})$ such that

$$\begin{aligned} \boldsymbol{M}^{\mathcal{T}} &= \boldsymbol{A}\,\boldsymbol{M}^{\mathcal{S}}\,\boldsymbol{A}^T\,, \\ \boldsymbol{M}_\ell^{\mathcal{T}} &= \boldsymbol{A}\,\boldsymbol{M}_\ell^{\mathcal{S}}\,\boldsymbol{A}^T\,, \end{aligned} \tag{4.65}$$

for $\ell \in \{1, \ldots, L\}$, the operations carried out by RPA end up directly determining the matrix $\boldsymbol{B}^\star = \boldsymbol{A}^{-1}$ that makes the cost function in (4.64) equal to zero.

Note, however, that if the assumptions in (4.65) are not valid, the set of transformations in RPA might be sub-optimal and one could prefer to solve (4.64) via a gradient descent procedure. Furthermore, if one wishes to impose some structure on $\boldsymbol{B}$, adding constraints to (4.64) would be the way to do so.

4.4 Numerical illustrations: simulated data

In this section, we compare the performance of several pipelines for doing classification using data from a *source* and a *target* dataset consisting of SPD matrices. We investigate whether this kind of classification can be improved when using the RPA method for matching the statistics of the datasets. Each pipeline trains a classifier on a *training* dataset composed of different kinds of transformed data points (coming from the *source* dataset and, in a lesser extent, from the *target* dataset). Then, the trained classifier is used to classify unlabeled data points from the *target* dataset. The performance of a pipeline is defined as being the area under the ROC curve (AUC score) for the classification task on the unlabeled data points from the *target* dataset.

4.4.1 The dataset

We simulated data for a *source* and a *target* dataset containing 2×2 SPD matrices belonging to two classes. Data points from the *source* dataset were generated as follows:

(1) Generate a random SPD matrix $\boldsymbol{M}_1^{\mathcal{S}} \in \mathcal{P}(2)$ and define it to be the geometric mean of class 1 in dataset $\mathcal{S}$;

(2) Generate $N_t = 100$ random SPD matrices around $\boldsymbol{M}_1^{\mathcal{S}}$ by mapping small random tangent vectors (norm fixed to $\varepsilon = 10$) from $T_{\boldsymbol{M}_1^{\mathcal{S}}}\mathcal{P}(2)$ back to the SPD manifold. We associate to each of these matrices the label $\ell_i^{\mathcal{S}} = 1$;

(3) Generate a random SPD matrix $\boldsymbol{M}_2^{\mathcal{S}}$ whose distance to $\boldsymbol{M}_1^{\mathcal{S}}$ is $\eta = 5$. For this, we generate a random SPD matrix and move it over a geodesic path starting at $\boldsymbol{M}_1^{\mathcal{S}}$, until the distance between the matrices is the one we desire. This is the geometric mean for class two in $\mathcal{S}$;

(4) Generate $N_t = 100$ random SPD matrices around $\boldsymbol{M}_2^{\mathcal{S}}$ by mapping small random tangent vectors (norm fixed to $\varepsilon = 10$) from $T_{\boldsymbol{M}_2^{\mathcal{S}}}\mathcal{P}(2)$ back to the SPD manifold. These matrices have label $\ell_i^{\mathcal{S}} = 2$ associated to them.

We generated the data points for the *target* dataset ($\mathcal{T}$) exactly as for the *source* dataset, but added an extra translation step that ensured that the geometric mean $\boldsymbol{M}^{\mathcal{S}}$ of all the matrices from $\mathcal{S}$ are at a distance $\zeta = 8$ from the geometric mean $\boldsymbol{M}^{\mathcal{T}}$ of $\mathcal{T}$.

4.4.2 Illustration of the steps of RPA

We first used the algorithm of diffusion maps [CL06] explained in Chapter 3 to obtain new representations of our data points using only two axis. Figure 4.3 illustrates the distribution of data points after each step of RPA applied to the *source* and

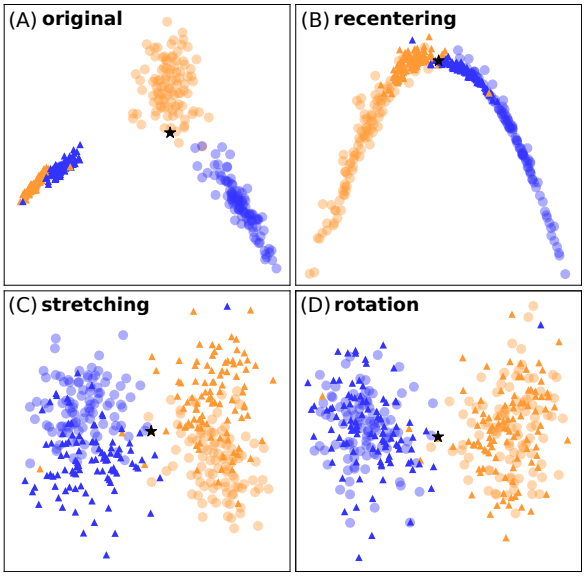

Fig. 4.3: Representation of the sequence of operations of RPA applied to a dataset simulated as described in Section 4.4.1 (better visualized with colors). Each point on the scatter plot represents a SPD matrix and the axes for the figures were obtained using Diffusion Maps [LL06]. The triangles represent the *target* dataset whereas the circles are the *source* dataset. Each color represents a class and the black star is the Identity matrix. (A) Distribution of the SPD matrices in the *source* and *target* datasets as they are originally available and (B) after re-centering their geometric means to the Identity. In (C) the distribution after the stretching operation and (D) after the rotation.

target datasets. In this example, we consider that we know the labels of all matrices from the *target* dataset, i.e., $\mathcal{T} = \mathcal{T}_\ell$. Figure 4.3A shows the point clouds of each dataset, which are clearly unmatched. After re-centering (Figure 4.3B), stretching (Figure 4.3C) and rotating (Figure 4.3D), the statistical distributions get matched and the same classifier can be used on both datasets.

4.4.3 Classification accuracy after RPA

We compared the classification accuracy on the simulated dataset for six different pipelines. In each of them, the training ($\mathcal{D}_{\text{train}}$) and testing ($\mathcal{D}_{\text{test}}$) datasets were different but we always used the minimum distance to mean (MDM) classifier (see Chapter 2 for details on this classifier):

- **direct (DCT)**: directly use the points from the *source* dataset to do classification on the unlabeled points from the *target* dataset (i.e., no transformation whatsoever),

$$\mathcal{D}^{\text{DCT}}_{\text{train}} = \mathcal{S} \cup \mathcal{T}_\ell \quad \text{and} \quad \mathcal{D}^{\text{DCT}}_{\text{test}} = \mathcal{T}_u \,. \tag{4.66}$$

- **re-centering (RCT)**: transfer learning considering only the data points of each dataset re-centered to $\boldsymbol{I}_d$. This corresponds to step (1) in the RPA procedure and is similar to what has been done in [Zan+17], with

$$\mathcal{D}_{\text{train}}^{\text{RCT}} = \mathcal{S}^{(\text{rct})} \cup \mathcal{T}_{\ell}^{(\text{rct})} \quad \text{and} \quad \mathcal{D}_{\text{test}}^{\text{RCT}} = \mathcal{T}_{u}^{(\text{rct})} \,. \tag{4.67}$$

- **parallel transport (PRL)**: transfer learning using the method proposed in [Yai+19]. The procedure is analogous to **RCT**, but with the SPD matrices being re-centered to the halfway point along the geodesic path linking the geometric means of each dataset instead of the Identity matrix.

- **optimal transport (OPT)**: transfer learning using the optimal transport approach proposed in [Cou+17] and adapted to take into account the fact that we have data points defined in the SPD manifold instead of Euclidean vectors. See [Yai+19] for details.

- **RPA**: transform matrices using RPA as described in Section 4.3.3,

$$\mathcal{D}_{\text{train}}^{\text{RPA}} = \mathcal{S}^{(\text{RPA})} \cup \mathcal{T}_{\ell}^{(\text{RPA})} \quad \text{and} \quad \mathcal{D}_{\text{test}}^{\text{RPA}} = \mathcal{T}_{\ell}^{(\text{RPA})} \,. \tag{4.68}$$

- **calibration (CLB)**: classification using only the labeled trials available in the *target* dataset, with no help from the data in the *source* dataset,

$$\mathcal{D}_{\text{train}}^{\text{CLB}} = \mathcal{T}_{\ell} \quad \text{and} \quad \mathcal{D}_{\text{test}}^{\text{CLB}} = \mathcal{T}_{u} \,. \tag{4.69}$$

We assessed the performance of each method via a randomized cross-validation procedure consisting of:

(1) Select $2n$ random elements from $\mathcal{T}$ (n from each class). These data points define $\mathcal{T}_{\ell}$.

(2) Define $\mathcal{T}_{u}$ containing the other $200 - 2n$ elements of $\mathcal{T}$.

(3) Obtain the classification score of MDM for this particular partition of $\mathcal{T}$.

(4) Repeat the above steps ten times and get the mean score for each method.

The results in Figure 4.4 show that the **DCT** pipeline gives classification results at chance level (0.5) independently of the number of matrices available in $\mathcal{T}_{\ell}$. We also observe that simply using **RCT** already greatly improves classification accuracy, as reported in [Zan+17]. Our **RPA** method further improves the results. We also observe that **PRL** has virtually the same performance as **RCT**, which is not surprising, since they are both unsupervised methods based on the idea of re-centering the datasets to a common point in the SPD manifold. The results with **OPT** are equivalent to **RCT** and **PRL** as well. The accuracy with **CLB** improves when the number of

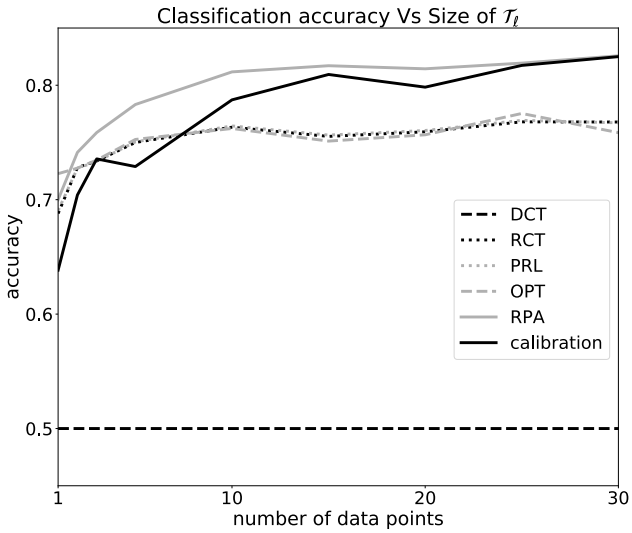

Fig. 4.4: Accuracy of the classification of unlabeled data points from the *target* dataset for different methods of transfer learning. The curve shows how the accuracy for each method evolves when the number of data points in $\mathcal{T}_\ell$ increases. The generation of the data points is explained in Section 4.4.1.

available labels in the *target* dataset increases, eventually converging to the same performance as **RPA**. This result is not surprising, because with a sufficient amount of data in $\mathcal{T}_\ell$ it is already possible to train a good classifier without the need of doing transfer learning.

Our observations in this session are in accordance with the theoretical results of [BD+09], which says that "if there is enough target data, then no source data are needed (...). This is because the possible reduction in error due to additional source data is always less than the increase in error caused by the source data being too far from the target data". Such a result points to the existence of a certain saturation effect in the quality of transfer learning when too many trials are available in the *target* session, a behavior that could be exploited to decide when to stop transferring information from previous experimental sessions.

4.5 Numerical illustrations: real data

In this section, we consider the problem of cross-subject classification with data from BCI experiments. We use the Riemannian geometric framework presented in Chapter 2 to parametrize the EEG epochs via symmetric positive definite matrices. Each pair of *source-target* datasets comes from a different pair of subjects and the goal is to assess whether the transformations with RPA can improve this kind of classification. As usual, the BCI data is in the form of d-dimensional multivariate time-series, where each dimension represents an electrode. Each experimental trial i lasts a few seconds and is associated to a matrix $\boldsymbol{X}_i \in \mathbb{R}^{d \times T}$, where T is the

Tab. 4.1: Main features describing each dataset used in this work.

dataset	paradigm	subjects	classes	trials per class	reference
PhysionetMI	MI	109	2	22	[Sch+04]
Cho2017	MI	50	2	100	[Cho+17]
SSVEP	SSVEP	12	3	8	[Kal+16]
P300	P300	24	2	72 and 360	[Con+11]
BNCI2014001	MI	9	4	72	[Tan+12]
BNCI2014002	MI	15	2	80	[Ste+16]
BNCI2015001	MI	13	2	100	[Fal+12]
MunichMI	MI	11	2	150	[GW+09]

number of time samples defining the trial. To every trial we associate a SPD matrix $\boldsymbol{C}_i$ describing its multivariate statistics and a label ℓ_i indicating what was the task performed during the trial. The dataset for each subject is composed of a set of couples $(\boldsymbol{C}_i, \ell_i)$. Our investigation focus on the classification accuracy of a MDM classifier that is trained with the data from a *source* subject plus a few labeled points from a *target* subject and is used to classify the unlabeled signals from the *target* subject. We compare the performance of such classifier using the different transfer learning strategies described in Section 4.4.3.

4.5.1 The datasets

Our investigations were carried out on eight datasets covering three different BCI paradigms. All motor imagery (MI) and P300 datasets are publicly available and were downloaded and pre-processed using the `MOABB` framework [JB18]. The `SSVEP` dataset was the same as the one presented in [Kal+16]. See Table 4.1 for a brief overview of each dataset's features. We estimated the SPD matrices parametrizing the EEG epochs of each BCI paradigm differently, as discussed in Chapter 2: for MI datasets, the SPD matrices were the spatial covariance matrices of the multivariate EEG recordings. The signals of each trial in the SSVEP paradigm were filtered using bandpass filters around certain frequencies of interest and its SPD matrices were diagonal blocks concatenating the spatial covariance matrices of the filtered signals [Con13]. For the P300, the SPD matrix of each trial was obtained using the approach from [BC14], where one estimates a special form of covariance matrix that captures the influence of event-related potentials in each trial.

4.5.2 Comparing cross-subject classification accuracies

We first compared the performance of a MDM classifier considering all pairwise combinations of *source* and *target* subjects. The classification scores were assessed using the same cross-validation scheme explained in Section 4.4.3.

Seriation. We begin with a qualitative analysis of the cross-subject classification scores using a tool from combinatorial data analysis called *seriation* [Lii10]. This procedure sorts the lines and columns of a data matrix in order to make relevant patterns appear. In our case, the matrix $\boldsymbol{S}$ to be re-ordered contains at its (i, j) coordinate the accuracy of the classification using subject i as *target* and subject j as *source*. Suppose that $\boldsymbol{S} \in \mathbb{R}^{N_{\mathcal{T}} \times N_{\mathcal{S}}}$, where $N_{\mathcal{T}}$ is the number of *target* subjects and $N_{\mathcal{S}}$ is the number of *source* subjets. We proceed as follows:

(1) For each row i, obtain the sum along the columns of $\boldsymbol{S}$, denoted by

$$\boldsymbol{S}_{i,:} = \sum_{j=1}^{N_{\mathcal{S}}} \boldsymbol{S}_{i,j} \ . \tag{4.70}$$

(2) Sort the rows of $\boldsymbol{S}$ in decreasing order according to $\boldsymbol{S}_{i,:}$. We obtain a new row-sorted matrix $\boldsymbol{S}^{r\downarrow}$ where

$$\boldsymbol{S}^{r\downarrow}_{1,:} \geq \boldsymbol{S}^{r\downarrow}_{2,:} > \cdots \geq \boldsymbol{S}^{r\downarrow}_{K_{\mathcal{T}},:} \ . \tag{4.71}$$

(3) Obtain the sum along the rows for each column j of $\boldsymbol{S}^{r\downarrow}$, denoted by

$$\boldsymbol{S}^{r\downarrow}_{:,j} = \sum_{i=1}^{N_{\mathcal{T}}} \boldsymbol{S}_{i,j} \ . \tag{4.72}$$

(4) Sort the columns of $\boldsymbol{S}^{r\downarrow}$ in decreasing order according to $\boldsymbol{S}^{r\downarrow}_{:,j}$. We obtain a new matrix $\boldsymbol{S}^{\downarrow}$.

The output of this procedure is a new representation where the pairs of *source-target* subjects with the best accuracy are located at the top-left region of the matrix, while the worst pairs are at the bottom-right region. Figure 4.5 shows the results of this seriation procedure on the `PhysionetMI` dataset for two sizes of $\mathcal{T}_\ell$ (the number of labeled *target* trials) and three different pipelines: **DCT**, **RCT**, and **RPA**. We observe that with **RCT** and **RPA** there are more pairs of subjects with high values of cross-subject classification than with **DCT**. In particular, for **RCT** and **RPA** we note that there are a few *target* subjects that have very good accuracy on classification for almost all possible *source* subjects, a feature that is possibly related to the performance of each *target* subject to classify its own trials (intra-subject accuracy). This can been interpreted as: subjects that are “good" for classifying their own data should be “good" for receiving information from other *source* subjects. We also observe a clear improvement in the average value of the cross-subject classification accuracies when more points are available in $\mathcal{T}_\ell$.

Average performance. Our next analysis consists in calculating the mean over all cross-subject AUC’s (Area Under the ROC Curve) for each transfer learning pipeline on each dataset. We used these values as quantitative measures for assessing whether

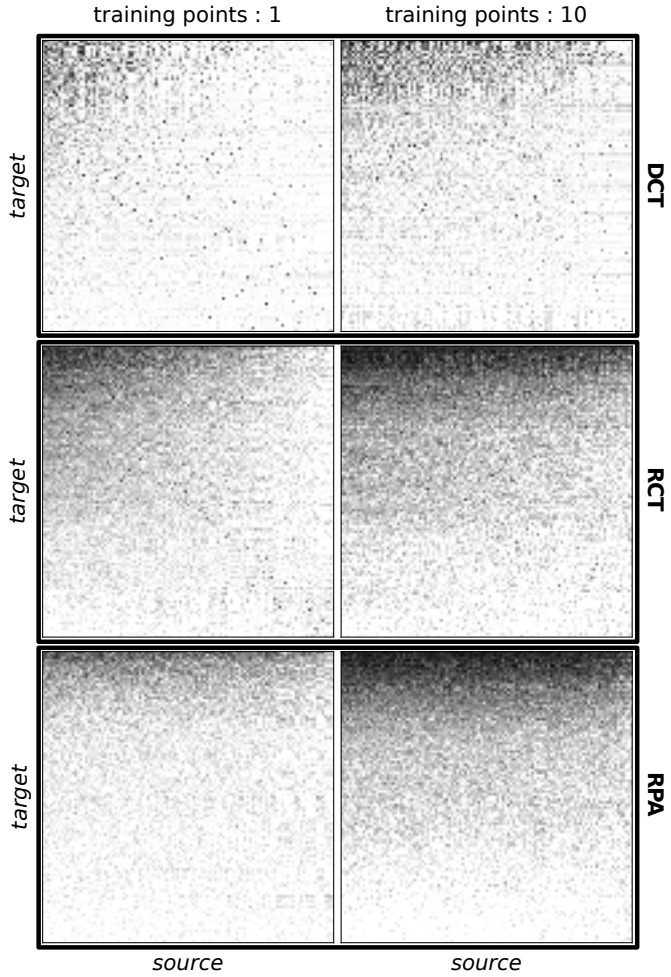

Fig. 4.5: Accuracies of the cross-subject classification for three different transfer learning procedures on the `PhysionetMI` database. The rows and columns of each subplot were reordered using the seriation procedure explained in the text. The colormap shows white for accuracies of 0.5 or less and black when it is 1.0. The compared methods are described in Section 4.4.3 and we consider the cases when there are one and ten labeled matrices in $\mathcal{T}_\ell$.

Tab. 4.2: Mean values of the cross-subject AUC (Area Under the ROC Curve) for five pipelines (all described in Section 4.4.3) on eight different datasets. Parameter N is the number of training points available on the *target* dataset on each situation. The best method in each instance is written in **bold**. The pipelines under comparison were presented in Section 4.4.3.

PhysionetMI

MEAN AUC						
N	DCT	RCT	PRL	OPT	RPA
1	0.54	**0.61**	**0.61**	0.59	0.56
5	0.55	**0.65**	**0.65**	0.60	0.63
10	0.56	**0.67**	**0.67**	0.60	0.66
15	0.57	**0.68**	**0.68**	0.60	0.67

Cho2017

MEAN AUC						
N	DCT	RCT	PRL	OPT	RPA
1	0.54	**0.59**	0.58	0.57	0.54
5	0.55	**0.61**	**0.61**	0.57	0.59
10	0.55	**0.62**	**0.62**	0.57	**0.62**
25	0.57	0.64	0.64	0.58	**0.66**

BNCI2014001

MEAN AUC						
N	DCT	RCT	PRL	OPT	RPA
1	0.58	**0.69**	**0.69**	0.65	0.62
6	0.59	**0.73**	**0.73**	0.65	0.71
18	0.61	**0.76**	**0.76**	0.65	**0.76**
36	0.64	0.78	0.77	0.66	**0.79**

BNCI2014002

MEAN AUC						
N	DCT	RCT	PRL	OPT	RPA
1	0.55	**0.68**	0.67	0.64	0.63
5	0.56	**0.71**	0.70	0.64	0.70
10	0.57	**0.72**	0.71	0.65	**0.72**
20	0.58	**0.73**	0.72	0.65	**0.73**

SSVEP

MEAN AUC						
N	DCT	RCT	PRL	OPT	RPA
1	0.64	0.67	0.66	0.59	**0.70**
2	0.67	0.71	0.71	0.59	**0.75**
4	0.72	0.76	0.76	0.59	**0.80**
6	0.74	0.78	0.78	0.57	**0.82**

P300

MEAN AUC						
N	DCT	RCT	PRL	OPT	RPA
6	0.57	0.56	0.56	**0.58**	0.55
12	0.62	**0.64**	**0.64**	0.61	0.63
32	0.71	**0.74**	**0.74**	0.67	0.73
48	0.74	**0.76**	**0.76**	0.69	0.75

BNCI2015001

MEAN AUC						
N	DCT	RCT	PRL	OPT	RPA
1	0.51	0.56	0.55	0.54	**0.57**
5	0.51	0.57	0.57	0.55	**0.60**
10	0.52	0.59	0.58	0.56	**0.63**
25	0.54	0.62	0.62	0.56	**0.65**

MunichMI

MEAN AUC						
N	DCT	RCT	PRL	OPT	RPA
1	0.55	**0.63**	**0.63**	0.61	0.55
25	0.58	**0.69**	**0.69**	0.62	0.68
50	0.60	0.71	0.71	0.62	**0.72**
75	0.62	0.72	0.72	0.62	**0.73**

one pipeline is better than the other on cross-subject classification. The scores are shown in Table 4.2. We should mention that only the subjects whose intra-AUC (i.e., classification of its own data) was above chance level were used in these calculations. Figure 4.6 shows the results of statistical tests performed on each pair of methods, allowing for a more substantiated assessment of the performance of the methods. The statistical tests comparing method **A** versus method **B** were carried out in the following way: (1) For each *target* subject i, we perform a signed paired t-test comparing the scores of method **A** to method **B** along all *source* subjects. Each of these tests yields a statistic T_i and a p-value p_i is obtained via permutations tests [EO07]. (2) We combine the p-values of all the *target* subjects using Stouffer's Z-score method [Zay11]. This yields a single p-value for the comparison between methods as well as the direction to which the null hypothesis has been rejected (i.e., whether method **A** is better than **B** or vice-versa). (3) We adjust the p-values of each pairwise comparison using Holm's step-down procedure [Hol79] to account for the multiple comparison problem.

The results in Figure 4.6 indicate that when there are enough points in $\mathcal{T}_\ell$ ("enough" depending on each dataset), transforming the data points with **RCT**, **PRL** or **RPA** is always better than not doing any distribution matching (**DCT**). We also observe that most of the time there is no statistical significance between the results with **PRL** and **RCT**, as expected, since they both amount to re-centering the datasets to a new point in the SPD manifold. For increasing values of N (the number of labeled trials in the *target* dataset), **RPA** gets better in comparison to almost all other methods, as expected and observed in Figure 4.4 for simulated data. Interestingly, **OPT** has very poor results in comparison to all other methods, probably because it does not use any prior hypothesis on the statistical distributions of the datasets and has to solve a difficult optimization problem to determine its transportation plan. Lastly, during our statistical analysis of the results, we have observed that better results on transfer learning via RPA are often associated to good intra-subject accuracy, since in this case the estimation of the class means is more stable and thus the rotation matrix $\boldsymbol{U}$ is better estimated. This explains why for some databases (e.g. `PhysionetMI`) the

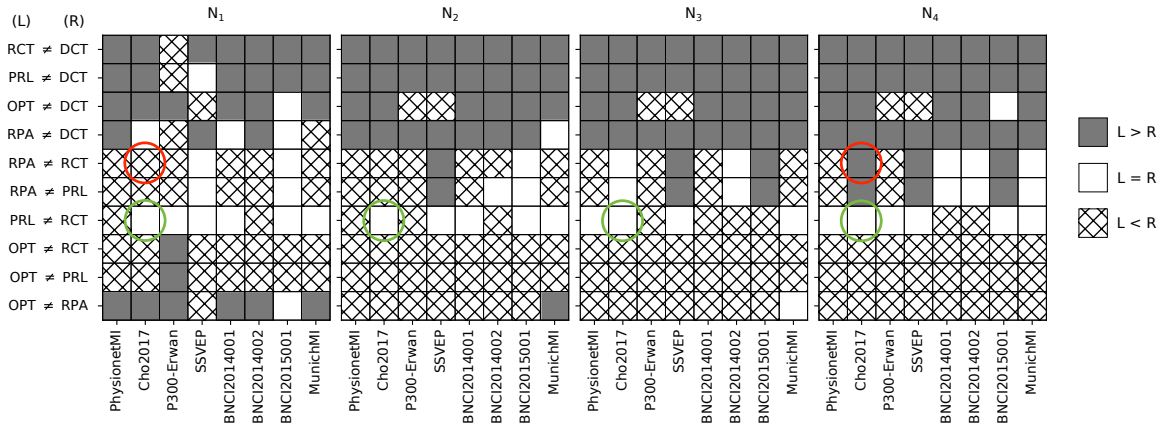

Fig. 4.6: Results of the statistical tests on each pair of pipelines for all possible values of N on each dataset as indicated in Table 4.2 (for instance, on `Cho2017` we have $N_1 = 1, N_2 = 5, N_3 = 10$, and $N_4 = 25$). The color/pattern of the squares indicate whether there's no statistical difference between two methods (white squares), if the Left method is superior to the Right one (L and R in the legend) (dark gray squares) or the contrary (squares with crossed patterns). All conclusions are with $p < 0.05$ corrected via Holm's adjustement [Hol79]. For instance, we see that for dataset `Cho2017`, the method **RPA** is inferior to **RCT** when $N = 1$, but **RPA** becomes superior when $N = 25$ (red circles). Furthermore, for this same dataset, there's no statistical difference in the comparison between **PRL** and **RCT** for any N (green circles).

RPA is not necessarily the best method for transfer learning and an unsupervised approach like **RCT** has better results.

Conclusions. We compared the performance of the RPA procedure for transfer learning to that of other distribution-matching methods proposed in [Zan+17; Yai+19] and [Cou+17]. The results demonstrate that the RPA yields a superior classification accuracy in both simulated and real datasets. We also observe that, in general, RPA needs a very small amount of labeled trials from the *target* dataset to work well.

4.5.3 An exploratory analysis of cross-subject classification

We have observed in the previous section that although any pair of *source–target* subjects can go through a transfer learning procedure, some pairs of subjects yield better results in classification than others. We investigate now some of the factors that might explain this variability and how one might try to predict beforehand (i.e., before doing any matching of the datasets or classifying their data points) the "compatibility" between the datasets. We consider three classification pipelines: {**DCT**, **RCT**, **RPA**} (see Section 4.4.3 for a description).

Our exploratory analysis relies on the estimation of linear models and the study of the statistical significance of the coefficients estimated for those models. We use as explanatory factors the intra-scores for the *source* and *target* subjects (which is

the cross-validated classification score using the subject's dataset as training and testing dataset), and distance $\mathcal{W}$ between the statistical distribution of the two datasets as defined in (4.13) in Section 4.3.1. We observe that the intra-score for the *target* subject plays an important role in determining how well the transfer learning will work, as opposed to the intra-scores of the *source* subjects, which plays no statistically significant role in most cases. We also observe that before doing any transformation on the data points of the *source* and *target* datasets, distance $\mathcal{W}$ between their statistical distributions plays a statistically significant role over the performance of the transfer learning. However, once the RPA is applied, the distance $\mathcal{W}$ between datasets becomes very small and no longer carries statistical information to describe the variability of the cross-subject scores. This confirms the relevance of the RPA method.

Linear regression models. Given a dataset, all cross-subject transfer learning performance is summarized in a matrix $\boldsymbol{S}^{(m)}$, where the $\boldsymbol{S}_{ij}^{(m)}$ element contains the accuracy of the classification with method $m \in \{\textbf{DCT}, \textbf{RCT}, \textbf{RPA}\}$ using subject i as *target* and subject j as *source*. We use linear regression models to describe the variability on the values of $\boldsymbol{S}_{ij}^{(m)}$ and estimate a different linear model $\mathcal{L}_i^{(m)}$ for each *target* subject i and method m. We do this because the cross-subject scores for two different *target* subjects and the same *source* subject are statistically dependent, which would undermine the estimation of a full linear model mixing all scores. Moreover, the results after the **RPA** method are related to those for the **RCT** one, since the latter includes the former as a processing step.

We define the linear model $\mathcal{L}_i^{(m)}$ as:

$$\boldsymbol{S}_{ij}^{(m)} = \beta_{1,i}^{(m)} S_i + \beta_{2,i}^{(m)} S_j + \beta_{3,i}^{(m)} \eta_{ij}^{(m)} + \epsilon_i^{(m)} \,, \tag{4.73}$$

where

- S_i (S_j) is the intra classification score of *target* (*source*) subject i (j), obtained via cross-validation with training and testing datasets coming from the same subject. Note that since each model $\mathcal{L}_i^{(m)}$ is estimated for one fixed *target* subject i, S_i is a constant in (4.73) and acts as a scaling for the intercept; thus, it is not considered as an independent variable in the statistical analysis.

- Factor $\eta_{ij}^{(m)}$ is the distance $\mathcal{W}$ between the statistical distributions of datasets $\mathcal{S}$ and $\mathcal{T}$ after the operations of method m.

- Variable $\epsilon_i^{(m)}$ stands for all residual factors that are not explained by the linear regression model.

Once the linear models are all estimated, we perform a set of hypothesis tests for each *target* subject i. The goal is to assess the statistical significance of the coefficients of each model. The first kind of test is a F-test for the omnibus null hypothesis:

$$\begin{aligned} \mathcal{H}_0 &: \quad \beta_{2,i}^{(m)} = \beta_{3,i}^{(m)} = 0 \ , \\ \mathcal{H}_1 &: \quad \beta_{k,i}^{(m)} \neq 0 \text{ for at least one } k \text{ in } \{2,3\} \ . \end{aligned} \tag{4.74}$$

This is a standard test used for inspecting whether the set of independent variables of a linear regression model, S_j and η_{ij} in (4.73), is statistically significant for explaining at least part of the variability of the dependent variable, $\boldsymbol{S}_{ij}^{(m)}$ in (4.73). When the null hypothesis is rejected, we say that there is enough statistical evidence for considering that the slope of at least one of the independent variables is different than zero. In this case, we perform t-tests for checking which explanatory variable in $\mathcal{L}_i^{(m)}$ is statistically significant. We put:

$$\begin{aligned} \mathcal{H}_0 &: \quad \beta_{\ell,i}^{(m)} = 0 \ , \\ \mathcal{H}_1 &: \quad \beta_{\ell,i}^{(m)} \neq 0 \ , \end{aligned} \tag{4.75}$$

for $\ell \in \{2,3\}$. When the null hypothesis of (4.75) is rejected for $\beta_{\ell,i}^{(m)}$, we say that there is statistical evidence for considering it different than zero and so it contributes for explaining the variability of the dependent variable $\boldsymbol{S}_{ij}^{(m)}$.

The statistical procedure explained above yields two sets of p-values for each method $m \in \{\textbf{DCT}, \textbf{RCT}, \textbf{RPA}\}$. The first set contains the p-values for each F-test on each *target* subject i, whereas the second set gathers the p-values of the t-tests. The results presented next are based on the analysis of these sets of p-values and how they are distributed along different *source* subjects for each method.

Dataset. We carried out our analysis on dataset `Cho2017` presented in Table 4.1. The dataset contains recordings of subjects performing BCI trials following a Motor Imagery (MI) paradigm with 64 EEG electrodes (sampling frequency 512 Hz) from 52 subjects, each one performing 200 trials (100 of each class). We filtered the EEG signals in the 8-30 Hz band and each trial was considered as a segment from 0.5 to 2.5 seconds after the trial onset. We used the approach described in Chapter 2 for parametrizing BCI recordings under the MI paradigm: each epoch is associated to its spatial covariance matrix. Not all subjects in `Cho2017` have data which can be well discriminated, so we kept only those whose the intra-score in terms of AUC (Area Under the ROC-curve) is above chance level; this keeps 40 subjects out of the 52 in total.

Cross-subject classification accuracy. Figure 4.7 shows the output of the seriation procedure (described in Section 4.5.2) on the cross-subject transfer learning scores for the `Cho2017` dataset on three classification methods: **DCT**, **RCT**, and **RPA**. We

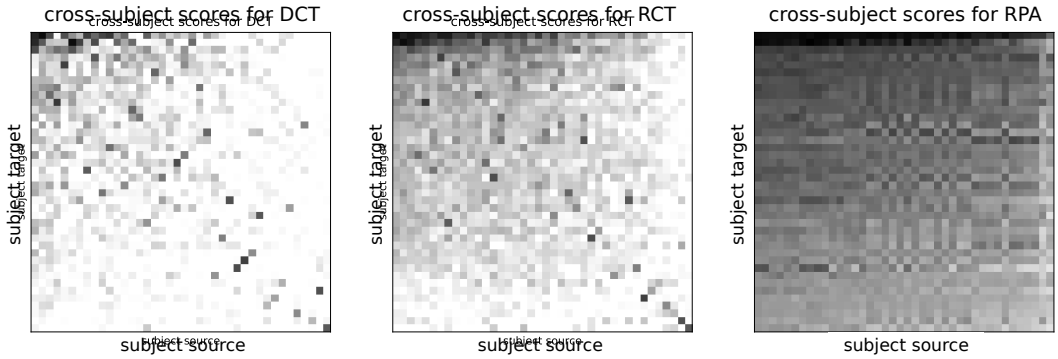

Fig. 4.7: Accuracies of the cross-subject classification for three different transfer learning procedures on the `Cho2017` database. The rows and columns of each subplot were reordered using the *seriation* procedure explained in Section 4.5.2. The colormap varies from white (accuracy 0.5) to black (accuracy 1.0).

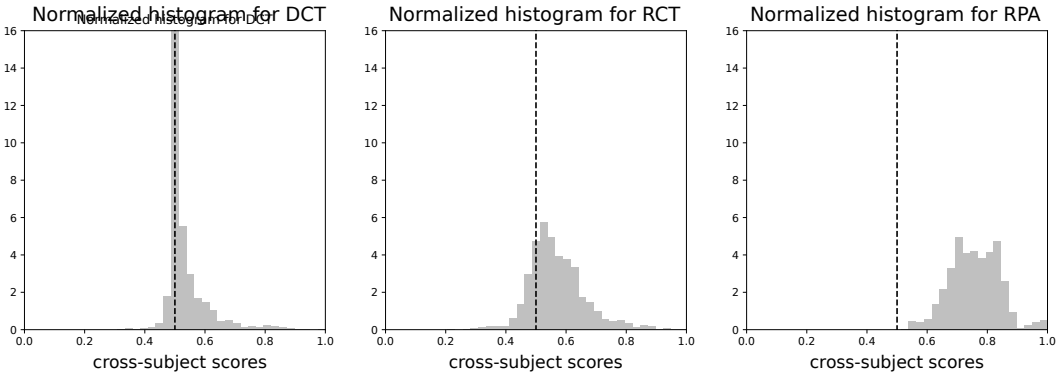

Fig. 4.8: Normalized histograms of the cross-subject transfer learning scores for the three methods described in the text. The vertical dashed line indicates chance level.

observe that, with **RCT** and **RPA**, there are more pairs of subjects with high values of cross-subject classification than with **DCT**. In particular, we note that for **RCT** and **RPA** there are many *target* subjects for which the classification accuracy is high for almost all possible *source* subjects. To investigate the possible explanations for this behavior, we perform a *Spearman* correlation test between the average cross-subject score for each target (given by the average value along the rows of matrix S) and the intra-subject accuracy of the corresponding *target* subject. For the **RPA** method, we obtain a correlation of 0.58 ($p < 10^{-3}$), whereas for **RCT** it is 0.44 ($p < 10^{-2}$) and **DCT** is 0.45 ($p < 10^{-2}$). Similarly to what was obtained in Section 4.5.2 for the `PhysionetMI` dataset, we interpret these results as: subjects that are "good" for classifying their own data can better receive information from other *source* subjects. We also provide a quantitative analysis of the results. Figure 4.8 portrays the histograms of all cross-subject transfer learning scores $S_{ij}^{(m)}$ (rows and columns confounded) for each method and their means are displayed in Table 4.3. These results show that the transformations over the *source* and *target* datasets do improve the cross-subject classification scores on the average.

Tab. 4.3: Average values of the cross-subject transfer learning scores and the $\mathcal{W}$ distance between *source* and *target* datasets for each method.

Method	$(\eta_{ij})_{\text{avg}}$	$(S_{ij})_{\text{avg}}$
DCT	0.63	0.53
RCT	0.01	0.58
RPA	0.01	0.76

Changes in $\mathcal{W}$ after each RPA step. We evaluate how the distance $\mathcal{W}$ between each pair of *source–target* subjects changes after the re-centering step and the full RPA procedure. Table 4.3 gives the average values of $\mathcal{W}$ for each method and shows that there is a clear decrease after each transformation. This result is not surprising, since each step of the RPA procedure was conceived exactly to make the distributions of $\mathcal{S}$ and $\mathcal{T}$ closer in some sense and the $\mathcal{W}$ allows for a quantitative assessement of it.

Study of the linear models $\mathcal{L}_i$. After exploring the grand averages of the cross-subject transfer learning scores and how they relate to a few explanatory factors, we analysed the linear models $\mathcal{L}_i$ defined in (4.73) and estimated on each *target* subject i for the three methods of interest: **DCT**, **RCT**, and **RPA**.

We first plotted the p-values of the F-test for each model sorted in ascending order. Under the omnibus null hypothesis for *target* subjects (that is, when the coefficients of the linear model are all zero), the p-values follow an uniform distribution and, thus, when sorted they will lie on a straight line [CB01]. By analysing the size of the p-values (i.e., inspecting whether it is close to zero and, therefore, the null hypothesis should be rejected) on the leftmost plot in Figure 4.9, we see that for almost all subjects the variability of the cross-subject performance is well explained by the linear model estimated for the **DCT** method and, in a lesser extent, for the **RCT** method. It is worth remembering that the statistical significance of the coefficient for the intercept, and, therefore, the influence of the intra-score S_i on describing the values of S_{ij}, is not assessed via the F-test. This is why we have calculated the *Spearman* correlation between the row-averaged S_{ij} and the S_i above in the text.

The distribution of the p-values in the center plot of Figure 4.9 shows that $\beta_{2,i}$ (the coefficient associated to the *source* scores) has no statistical significance in the linear model $\mathcal{L}_i$ for any of the *target* subjects in the **DCT** and **RPA** methods. However, for **RCT** it does seem to play a role for some *target* subjects. What we can conclude from these observations is that **RPA** is able to make the cross-subject transfer learning score independent of the choice of *source* subject (at least in terms of its intra score). As a consequence, it makes it easier to find "good" source subjects for each *target* subject, as it was already observed during our qualitative analysis of Figure 4.7.

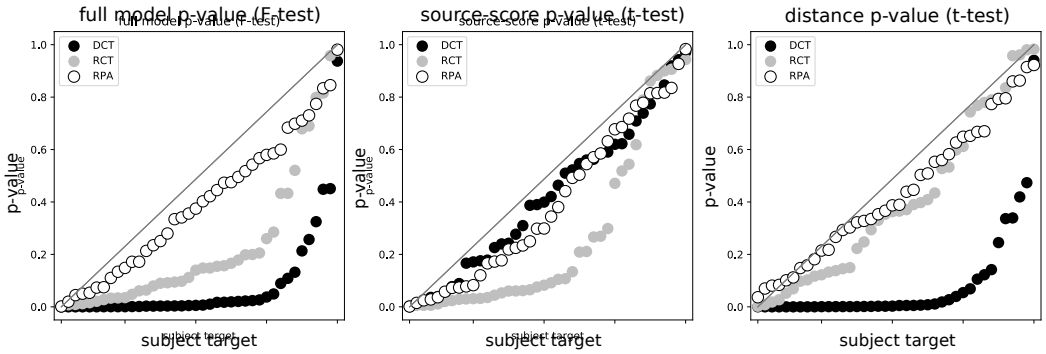

Fig. 4.9: p-values of different statistical tests over the linear models $\mathcal{L}_i$ (each one associated to a *target* subject i). Each circle represents the p-value of a given test on a given *target* subject and the x-axis has been rearranged so that all the p-values are in increasing order. The leftmost plot represents the results of the F-test of the full linear model $\mathcal{L}_i$, whereas the center plot illustrates the p-values for the t-test of the coefficient $\beta_{2,i}$ in $\mathcal{L}_i$ (related to the intra-score of the *source* subject), and the rightmost plot displays the p-values for the t-test on the coefficient $\beta_{3,i}$ in $\mathcal{L}_i$ (related to distance $\mathcal{W}$ between the statistical distributions of the *source* and *target* datasets).

Finally, the distribution of the p-values of $\beta_{3,i}$ on the rightmost plot in Figure 4.9 shows that the distance $\mathcal{W}$ between *source* and *target* datasets plays a role in describing the cross-subject transfer learning scores only for the **DCT** method. This result is comforting, since it brings evidence to the fact that the operations in the RPA procedure are capable of factoring out most of the differences between the statistical distributions of $\mathcal{S}$ and $\mathcal{T}$. As a consequence, we may say that any further improvement that one might want to do on the transfer learning procedure should take into account other aspects of the mismatch between datasets besides the distance $\mathcal{W}$ between them.

Conclusions. In this section, we have investigated the influence of different factors on the variability of cross-subject transfer learning scores. Our goal has been to assess whether some basic explanatory variables, such as the intra-score of the *source* and *target* subjects, play any role for determining the scores obtained in the cross-subject classification. A simple, and yet important, application of this study is being able to predict beforehand (i.e., before doing all transformations and then classifying the trials) which *source* subject would be the most appropriate for doing classification on a given *target* subject. It is our opinion that investigating the factors determining the sucess of transfer learning is instrumental for devising new and more powerful strategies for doing it. The present study is a little step in this direction. Future works shall include the search for richer models for describing the cross-subject transfer learning scores. Some approaches would be to consider non-linear relations between the explanatory variables as well as adding new factors related to other features of the *source* and *target* subjects.

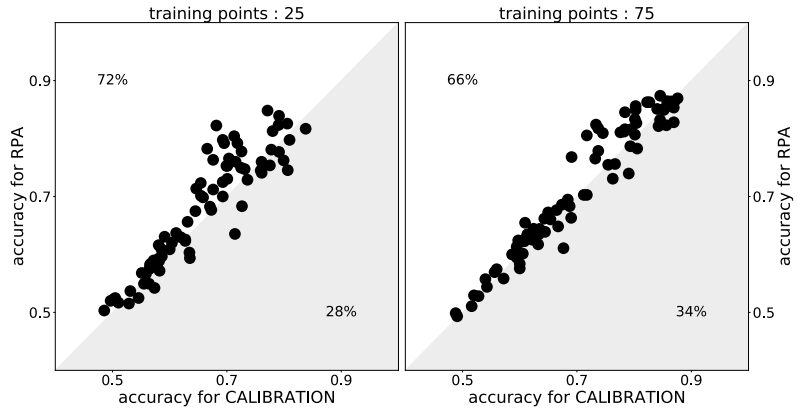

Fig. 4.10: Scatter plots comparing the accuracies of the cross-subject classification on the `MunichMI` dataset for the **RPA** and **CLB** pipelines. We consider two sizes for $\mathcal{T}_\ell$. The percentage numbers indicate the proportion of dots above or below the diagonal line.

4.5.4 The role of the size of $\mathcal{T}_\ell$

As pointed out in the simulation results from Section 4.4.3, when the size of $\mathcal{T}_\ell$ increases, using transfer learning is no longer relevant, since one may already have enough data to build a good classifier for the *target* subject. To investigate this behavior on our real datasets, we compared the cross-subject classification accuracy of **RPA** to that of **CLB** (calibration, which is when one uses only the labeled points in $\mathcal{T}_\ell$ for training a classifier).

Figure 4.10 shows a scatter plot comparing the classification accuracies on the `MunichMI` dataset. We see that as $\mathcal{T}_\ell$ grows, there are more pairs of subjects for which using the **CLB** pipeline on the *target* subject is better than doing transfer learning via **RPA** (28% to 34% of all the pairs of subjects). However, the location of the cloud of points in the figure indicates that the transfer learning with **RPA** is still superior to the **CLB** method for most pairs of subjects. We used a one-sided paired *t*-test with random data permutation [EO07] to compare the accuracies of **RPA** and **CLB** on each dataset for different sizes of the $\mathcal{T}_\ell$. The null hypothesis of equivalency between the two methods was rejected ($p < 0.01$) on almost all tests, the only exception being for those on the `BNCI2014001` dataset; for the tests where $\mathcal{H}_0$ was rejected, we observed a superiority of **RPA** in comparison to **CLB**.

4.5.5 Combining information from multiple subjects

In this sub-section, we investigate how the matching of statistical distributions via RPA affects the performance of two baseline methods for gathering information from the data of multiple subjects: *pooling* and *ensembling*. The MDM classification for each *target* subject is done using information coming from all other *source* subjects available in the database. Following the same approach as in previous sections, we

only considered *source* subjects featuring an intra-subject accuracy above chance level, i.e., subjects in which it is meaningful to use transfer learning. The experiments were done on `PhysionetMI` with $|\mathcal{T}_\ell| = 15$ labeled trials available for each *target* subject and `Cho2017` with $|\mathcal{T}_\ell| = 25$.

Pooling. The pooling strategy consists of gathering for each *target* subject the data from all other *source* subjects into one big dataset. Then, a classifier is trained on the pooled dataset and used to infer the trials from the *target* subject. We compared the performance of a MDM classifier when the *source* subjects were pooled with no transformation (**DCT**) to when the statistical distributions of each *source* subject were matched to that from the *target* subject using **RCT** or **RPA** (**PRL** is not fit for *pooling*, since the matrices are not all re-centered to the same place in the SPD manifold). The boxplots in Figure 4.11 show the distributions of the classification scores of each of the *target* subjects. Using pairwise one-sided paired t-test with random permutations, the null hypothesis of equivalency between the scores of **DCT**, **RCT**, and **RPA** were all rejected with $p < 10^{-6}$ (adjusted for multiple comparisons). The results show a clear improvement in the average score for the *pooling* strategy when using a method for matching the statistics of the *source* and *target* datasets, with differences of at least 15% between **RPA** and **DCT** for both datasets.

Ensembling. Our second analysis considered an ensembling strategy, where the trials of each *target* subject were classified using a majority voting scheme. These votes came from MDM classifiers trained on all other *source* subjects and were weighted equally. The results in Figure 4.11 show the scores with each method (including the **PRL** approach this time). To compare the scores of each method, we used pairwise one-sided paired t-tests with random permutations (corrected for multiple comparisons). The results of the statistical tests indicate that the ensembling strategy with **RPA** is superior as compared to **DCT** in the PhysionetMI dataset ($p < 0.05$) but they are equivalent for the Cho2017 dataset ($p = 0.23$). The **RCT** method is superior to **DCT** for both datasets ($p < 0.01$) whereas the scores with **PRL** are equivalent to **DCT** for both datasets. We see then that the ensembling strategy can also be improved when adding an extra step for matching the statistics of the datasets of each pair of *source-target* subjects.

Conclusions. Our tests have shown that one can achieve significant improvement in classification when matching the statistics of the datasets for every pair of subjects. The results with the pooling strategy show a significant improvement in the average performance for all subjects, whereas with ensembling the improvement is smaller but still present. Note that we did not make any selection or weighting on the contribution of each *source* subject for the classification on the *target* subject. However, works like [Way+16] and [Jay+15] have demonstrated clear improvements in cross-subject classification when this is done. We believe that, after the results observed in this sub-section, further improvements to the referred methods could

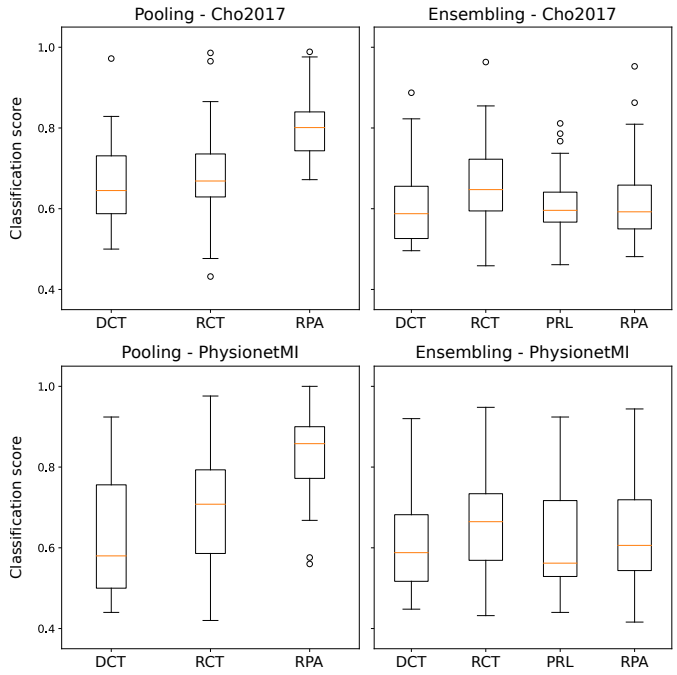

Fig. 4.11: Box plots with the distribution of the classification scores for the ensembling and pooling strategies for different methods of statistical matching between datasets. For the `Cho2017` dataset we had 25 labeled trials in the *target* dataset and for the `PhysionetMI` there were 15 labeled trials in it.

be attained if an extra step using RPA would be used for matching the statistical distributions of each pair of *source-target* subjects.

4.6 Conclusion

In this chapter, we have presented a new method for overcoming the negative effects of statistical distribution mismatch between datasets consisting of SPD matrices. Our proposal consists of a sequence of geometrical transformations on the elements of the datasets with the intention of making the shapes of the point clouds that they describe in a high-dimensional space as similar as possible. The inspiration for this method comes from Procrustes analysis, however, here the method has been adapted to the case where the elements of the datasets live in a Riemannian manifold. A relevant theoretical contribution is the mathematical framework proposed in Section 4.3.3, which includes the methods in [Zan+17] and [Yai+19] and extends them, leading to our RPA method. Such formalism allows for a better understanding of the intrinsic assumptions regarding the statistics of the data points during the distribution matching procedure.

When considering the SPD matrices as statistical descriptors of multivariate time series, the RPA procedure may be interpreted as a method that learns a linear transformation that mixes the dimensions of the time series in a *target* dataset so that

their statistics are aligned with those from a *source* dataset. The basic assumption behind this approach is that although the time series in two datasets may have different statistical distributions, if they are associated to the same phenomenon (e.g. the same cognitive tasks in a BCI experiment), then there should exist commonalities that could be exploited for relating the two datasets.

On a more practical note, an important aspect of RPA is that it exploits the availability of supervised information in the *source* session as well as the sequential nature of the trials in the *target* session. It should be noted, however, that when no labels are available for the *target* session, a re-centering of data points based solely on the geometric means of each dataset (which does not rely on any supervised information) already greatly improves the cross-session and cross-subject classification, as first noted in [Zan+17] and observed in the results of Section 4.4 and Section 4.5 (the **RCT** pipeline). This would be the case, for instance, in BCI applications for people with extreme motor disability, where the labeling of classes is very challenging. In this kind of situation, one may still perform the re-centering and stretching steps of the RPA method for matching the statistical distributions, turning the transfer learning procedure into an unsupervised one. Another relevant practical aspect to mention is that, when applied to brain-computer interfaces, RPA provides a way of thriving from information available in previous recording sessions and, therefore, reducing energy consumption and calibration time spent by a subject in a new session; this has a direct positive impact over the cost and feasibility of new BCI systems.

We have assessed the superiority of the RPA method on several publicly available BCI datasets and have used a heterogeneous panel of statistical tools to analyze the results. Also, we have included in our study other recent contributions from the literature, leading to a comprehensive comparison of the performance of state-of-the-art methods. We hope that the breath of the analysis performed here will be useful as a reference for future works related to transfer learning on the SPD manifold. In order to foster reproducible research, complete Python code for the results in this chapter is available at `https://github.com/plcrodrigues/RPA`.

Future perspectives shall include an online implementation of the RPA method, where usual drifts in statistics from data points on the same recording session would be corrected via distribution matching. An important challenge for such procedure would be to detect when changes in the statistics occur, as well as when the number of new trials is already large enough so that no information from data points drawn from previous statistical distributions are needed. Another interesting line of work would be to go further in the analysis of Section 4.5.5 by extending the methods proposed in [Faz+09], [Jay+15], and [Way+16] with a statistical matching step based on the RPA. Finally, another interesting topic to investigate would be to include hyper parameters in some of the cost functions of the RPA procedure. For instance,

the terms in the sum in (4.49) could be balanced by coefficients related to the 'quality' of the estimation of the geometric mean of each class on the *target* dataset. Another possibility would be to have a coefficient for weighthing the contribution of the data points from a *source* dataset as compared to the few labeled points from the *target* dataset. These parameters would provide more flexibility to the RPA and possibly yield better results.

Transcending dimensions

5

Contents

5.1 Introduction **119**
5.1.1 Contributions 120
5.2 Literature review **121**
5.3 Dimensionality transcending **124**
5.3.1 Problem statement 124
5.3.2 Expanding the dimensions of a SPD matrix 125
5.3.3 Matching the statistics of two datasets 129
5.3.4 Summary of the method 130
5.4 Application to BCI datasets **131**
5.4.1 Data imputation 131
5.4.2 Matching datasets 133
5.5 Numerical illustrations **134**
5.5.1 Data imputation 134
5.5.2 Matching datasets 137
5.6 Conclusion **144**

List of notations and acronyms of the chapter

EEG	electroencephalography
BCI	brain-computer interface
HPD	Hermitian positive definite
SPD	symmetric positive definite
RG	Riemannian geometry
AIRM	affine-invariant Riemannian metric
MDM	minimum distance to mean classifier
DT	dimensionality transcending
TL	transfer learning
RPA	Riemannian Procrustes analysis
RCT	re-centering
STR	stretching
ROT	rotation
DM	diffusion maps
MI	motor imagery
ROC	receiver operating characteristic
AUC	area under the ROC curve
$\mathbb{R}^d$	set of d-dimensional real vectors
$\boldsymbol{x}$	multivariate time series
$\boldsymbol{C}$	spatial covariance matrix
δ_E	Frobenius distance between two matrices
δ_R	AIRM-induced distance between two HPD matrices
$\mathcal{P}(d)$	manifold of d-dimensional HPD matrices
$\boldsymbol{M}^{\mathcal{X}}$	geometric mean of the HPD matrices in a set $\mathcal{X}$
$K_{\mathcal{A}}$	number of data points in set $\mathcal{A}$
$d_{\mathcal{A}}$	dimensionality of data points in set $\mathcal{A}$

5.1 Introduction

When setting up an experiment for measuring some physical phenomenon, an experimenter is faced with several practical choices, such as the kind and number of sensors to adopt, where to place them, which sampling frequency to use, etc. However, in general, it is reasonable to expect that the physical phenomenon under study is invariant to such choices and that small changes in the experiment's setup should not have drastic consequences in its ability to describe how the system evolves. For example, having 16 or 17 electrodes placed in similar positions on an electroencephalography (EEG) experiment does not have much impact regarding what activity one can observe from a subject's brain. Similarly, if an electrode presents a problem during the recording of an EEG epoch, it should be possible to still use the information from the other sensors without having to discard the whole epoch.

In traditional multivariate statistical analysis, the dimensionality of the data is always considered as being the same for all samples. However, in some practical cases, it might be useful to consider data samples that do not have the same dimensionality but describe the same phenomenon. For instance, a dataset $\mathcal{A}$ may be composed of three-dimensional samples describing the age, height and weight of the members of a certain population and another dataset $\mathcal{B}$ may have two-dimensional samples describing the age and the body mass index of people from the same population. Although the samples of each dataset do not have the same dimensionality (nor the same features), it is clear that they share some commonalities that could be jointly exploited to study the statistics of the population that $\mathcal{A}$ and $\mathcal{B}$ describe.

In this chapter, we propose a method that transforms multivariate time series recorded with different numbers of electrodes so that they become compatible in terms of dimensionality and statistical distributions. By the end of the procedure, the transformed datasets can be jointly used for performing different statistical tasks. For example, with our method, two datasets containing signals from subjects executing the same set of cognitive tasks for a brain computer interface (BCI), but recorded with different electrode configurations, may be used together for improving the classification performance of EEG epochs on the data from both experiments. Another application is when one (or some) electrode from the EEG recording presents a problem and the signal it registers must be rejected. The simplest approach would be to discard the whole epoch, but our method fills the missing values from the problematic channel in a way that the epoch can still be considered in the analysis.

Our proposal relies on the Riemannian geometric framework presented in Chapter 2, in which the statistics of the multivariate time series are parametrized via symmetric

positive definite (SPD) matrices[1]. The procedure consists of two steps : firstly, we transform the dimensionality of the SPD matrices so that they all become data points defined in the same space with a common dimensionality. Then, we apply geometric transformations to the data points of these dimension-transformed datasets so that their statistical distributions become as close as possible according to a distance that we will define later in the text. In the end, we have datasets that are defined in the same mathematical space and have compatible statistical distributions; this allows us to perform different statistical tasks on the SPD data points, such as clustering, classification, etc. Because it expands the dimensionality of the data points and surpasses the intrinsic limitations related to dimensionality mismatch, we call our method *dimensionality transcending* (DT).

The rest of the chapter goes as follows : Section 5.2 presents a literature review on methods and ideas related to the problem of dimensionality mismatch in statistical data analysis. Then, in Section 5.3, we present the dimensionality transcending method by first formalizing it mathematically and demonstrating some important properties. In Section 5.4, we describe how to apply DT to two practical problem involving EEG multivariate recordings and in Section 5.5 we use publicly-available datasets to illustrate these applications.

5.1.1 Contributions

The content of this chapter is based on (and extends) the works on two papers :

> P. L. C. Rodrigues, C. Jutten, and M. Congedo, “Dimensionality transcending: a method for working with datasets defined in different SPD manifolds", under preparation

and

> P. L. C. Rodrigues, M. Congedo, and C. Jutten, “A data imputation method for matrices in the symmetric positive definite manifold", XXVIIème colloque GRETSI, Lille, France, Aug. 2019.

Our main contributions in these works have been to propose the dimensionality transcending method and investigate its performance on a number of practical cases related to EEG-based brain computer interfaces datasets. It is also worth mentioning the fact of having used the `MOABB` framework [JB18] for downloading, processing, and analysing the EEG data, serving as a practical illustration of this powerful

[1] Chapter 2 also presents the parametrization of time series via their cross-spectral density matrices, which are Hermitian positive definite (HPD) matrices. Such description is more complete than just using the covariance matrices, but we will prefer the exposition with SPD matrices for the sake of simplicity for the exposition. Note that one could extend the results to a parametrization via HPD matrices by simply considering each frequency f of the cross-spectral density matrices independently.

benchmarking tool. Python code for the dimensionality transcending method is available in:

`http://www.github.com/plcrodrigues/PhD-Code/`

5.2 Literature review

In spite of its practical relevance, there have not been many works in the literature concerning the problem of using datasets with different dimensionalities for performing joint statistical analysis. In fact, the most common approach in such cases is to simply discard the dimensions of the data samples until they all share the same features and are defined in the same space. A clear downside of such approach is that it discards information that may be useful for extracting knowledge from the datasets. An alternative approach is to expand the data samples until they all have the same dimensionality and fill the new dimensions of the expanded data points with values that are adapted to the statistics of the datasets. A similar, but more general, method is to define transformations that map the data points with different dimensionalities into a common space, where all transformed samples have the same dimensionality and can be naturally compared. The method that we propose in this chapter follows this approach.

Heterogeneous domain adaptation. The branch of machine learning concerned with problems of statistical and dimensionality mismatch is called *heterogeneous domain adaptation*. Most methods proposed in the literature are based on procedures that learn the best projection of the datasets into a common latent space where their dimensionalities are the same and their differences in statistical distribution are minimized. An example is transfer component analysis [Pan+11], a method that learns a projection of the datasets into a reproducible kernel hilbert space and then searches for a transformation that matches the projected data points by minimizing their maximum mean discrepancy. Although this method works rather well in practice, it is not crafted for taking into account the intrinsic geometry of the manifold where the data points might be defined. Furthermore, it relies on an optimization procedure that solves a semi-definite programming problem, which can be computationally costly in some cases.

Data imputation. A particular practical case in which the data points have different dimensions is when a sample (or several of them) presents a problem in one (or more) of its features; for instance, when building the dataset $\mathcal{A}$ as defined in Section 5.1, one of the subjects in the population might have not been willing to inform his (or her) weight, or maybe the person responsible for gathering the data forgot to write down the age of one of the subjects, etc. When this happens, one may typically fill the missing value using the average of the problematic feature along the rest

of the population, in a procedure that is called *mean imputation* [Ber+18]. When the dataset is composed of different clusters and the average of a feature does not mean anything in particular, other imputation strategies may be in order. The field in statistics concerned with handling problems related to missing values is called 'statistical analysis with missing data' and the methods associated to it are usually named *missing data imputation*, or simply data imputation [LR02].

Comparing point clouds. A more abstract way of handling two datasets, $\mathcal{A}$ and $\mathcal{B}$, containing data points with different dimensionalities is to think of them as point clouds defined in high-dimensional spaces of dimensions $d_{\mathcal{A}}$ and $d_{\mathcal{B}}$, respectively. One may then use concepts from computational geometry [MS04] to study the geometrical properties of the datasets and investigate commonalities between them. For instance, the Gromov-Haussdorf distance may be used to compare the geometry of the set $\mathcal{A}$ to that of the set $\mathcal{B}$. This distance first requires defining two isometric transformations (i.e. transformations that preserve pairwise distances),

$$\begin{aligned} T_{\mathcal{A}} &: \mathbb{R}^{d_{\mathcal{A}}} \to \mathbb{R}^{d} , \\ T_{\mathcal{B}} &: \mathbb{R}^{d_{\mathcal{B}}} \to \mathbb{R}^{d} , \end{aligned} \tag{5.1}$$

where d is the smallest dimensionality of a space that ensures the existence of isometric transformations for the points in $\mathcal{A}$ and $\mathcal{B}$ into $\mathbb{R}^d$ (note that $d \geq \max(d_{\mathcal{A}}, d_{\mathcal{B}})$ always satisfies this condition). Then, after applying $T_{\mathcal{A}}$ to the elements of $\mathcal{A}$ and $T_{\mathcal{B}}$ to those from $\mathcal{B}$, we obtain two new sets of points, $T_{\mathcal{A}}(\mathcal{A})$ and $T_{\mathcal{B}}(\mathcal{B})$, defined in the same space. These new sets have the same geometrical structure of sets $\mathcal{A}$ and $\mathcal{B}$, respectively, since the transformations defined in (5.1) are isometric transformations. Therefore, measuring the distance between $T_{\mathcal{A}}(\mathcal{A})$ and $T_{\mathcal{B}}(\mathcal{B})$ in $\mathbb{R}^d$ is an adequate proxy for determining how 'close' $\mathcal{A}$ and $\mathcal{B}$ are. The distance between $T_{\mathcal{A}}(\mathcal{A})$ and $T_{\mathcal{B}}(\mathcal{B})$ is determined via the Haussdorf distance, which is a measure of similarity between sets of points commonly used in topology and defined as [MS04],

$$\delta_{\mathcal{H}}(T_{\mathcal{A}}(\mathcal{A}), T_{\mathcal{B}}(\mathcal{B})) = \max_{\boldsymbol{a}\in\mathcal{A}} \left(\min_{\boldsymbol{b}\in\mathcal{B}} \delta(\boldsymbol{a}, \boldsymbol{b}) \right) , \tag{5.2}$$

where δ is some distance in $\mathbb{R}^d$.

Dynamical systems. When the data points are multivariate time series describing the evolution of some physical system, the fact of handling data with different dimensionalities may be interpreted as having recordings of a dynamical system with different sets of observables. More precisely, consider two multivariate time series, $\boldsymbol{x}_{\mathcal{A}}(t) \in \mathbb{R}^{d_{\mathcal{A}}}$ and $\boldsymbol{x}_{\mathcal{B}}(t) \in \mathbb{R}^{d_{\mathcal{B}}}$, monitoring the same physical phenomenon.

We may model them as transformations of a vector of state variables $\boldsymbol{x}(t) \in \mathbb{R}^d$ that evolves according to some physical law, as in

$$\begin{aligned} \boldsymbol{x}_{\mathcal{A}}(t) &= f_{\mathcal{A}}(\boldsymbol{x}(t)) \,, \\ \boldsymbol{x}_{\mathcal{B}}(t) &= f_{\mathcal{B}}(\boldsymbol{x}(t)) \,, \end{aligned} \tag{5.3}$$

where $f_{\mathcal{A}} : \mathbb{R}^d \to \mathbb{R}^{d_{\mathcal{A}}}$ and $f_{\mathcal{B}} : \mathbb{R}^d \to \mathbb{R}^{d_{\mathcal{B}}}$ are functions that may be modelled according to the physical laws that drive the system under study. It is clear, then, that there are commonalities to be explored between the two time series. A practical example is when the multivariate time series are EEG recordings using different numbers of electrodes and/or placed in different locations over a subject's scalp. If the subject always performs the same task, it is reasonable to expect that the data recorded on different experimental setups share common information useful for describing the phenomena under study. Based on these assumptions, a recent work [Eng+18] has investigated how the performance of a statistical classifier based on random forests changes when using EEG data from different experimental setups. The results of the study indicate that it is indeed possible to gather information from different EEG databases with different electrode configurations and obtain 'reasonably good' classification scores.

The method that we present in this chapter considers the case when the data points are multivariate time series coming from recordings with different numbers and/or placement of sensors. Using the Riemannian geometric framework defined in Chapter 2, the statistics of these time series are parametrized via symmetric positive definite (SPD) matrices and our goal is to leverage from the information available on the datasets even if they are defined on spaces of different dimensionalities. For this, we proceed similarly to what was done for the Gromov-Hausdorff distance and define isometric transformations defined over the data points of each dataset. The new transformed samples live in the same space and preserve the intrinsic geometry of the initial datasets. Then, we use a domain adaptation technique crafted for SPD data points [Rod+18] to make the statistical distributions of the two transformed datasets compatible. From that moment forward, the samples of the two datasets live in the same space and have similar statistical distributions, so one can perform statistical tasks using the data from both datasets.

Before continuing, we should mention another work from the literature (appeared during the process of writing this thesis) that proposes a geometric distance between SPD matrices of different dimensionalities [Lim+19]. The proposal is based on the interpretation that SPD matrices may be associated to ellipsoids defined in high-dimensional spaces and that, when they have different dimensionalities, they may be compared with the help of embedding transformations. Note, however, that the work in [Lim+19] only proposes a notion of distance between points and does not

illustrate its use on practical problems, whereas we present a whole framework for working with datasets defined in different SPD spaces.

5.3 Dimensionality transcending

In this section, we present our method for matching the dimensionalities and statistics of two datasets consisting of matrices defined in symmetric positive definite (SPD) manifolds of different dimensions. We first formulate the problem mathematically and define a notation for it. Then, we discuss how to determine a transformation between SPD manifolds so that (1) the expanded matrices are also SPD and (2) the geometrical characteristics of a dataset containing expanded data points (e.g. its center of mass and dispersion) are easily determined from the characteristics of the original dataset. Finally, we recall the Riemannian Procrustes analysis (RPA) method presented in Chapter 4 and summarize the whole procedure. Because it expands the dimensionality of the data points and bypass the intrinsic limitations related to dimensionality mismatch, we call our method *dimensionality transcending* (DT).

5.3.1 Problem statement

Consider two datasets,

$$\mathcal{A} = \left\{ (\boldsymbol{C}_i^{\mathcal{A}},\ \ell_i^{\mathcal{A}}) \text{ for } i = 1, \ldots, K_{\mathcal{A}} \right\} \text{ and } \mathcal{B} = \left\{ (\boldsymbol{C}_i^{\mathcal{B}},\ \ell_i^{\mathcal{B}}) \text{ for } i = 1, \ldots, K_{\mathcal{B}} \right\}, \tag{5.4}$$

with data points $\boldsymbol{C}_i^{\mathcal{A}} \in \mathcal{P}(d_{\mathcal{A}})$ and $\boldsymbol{C}_i^{\mathcal{B}} \in \mathcal{P}(d_{\mathcal{B}})$, and class labels $\ell_i^{\mathcal{A}}, \ell_i^{\mathcal{B}} \in \{1, \ldots, L\}$, where L is the number of classes. We denote $\boldsymbol{M}^{\mathcal{A}}$ and $\boldsymbol{M}^{\mathcal{B}}$ the geometric mean of the matrices of each dataset, and $\sigma^{\mathcal{A}}$ and $\sigma^{\mathcal{B}}$ the dispersions around the geometric mean (see Chapter 2 for a definition of these quantities). The class means for each dataset are denoted $\boldsymbol{M}_\ell^{\mathcal{A}}$ and $\boldsymbol{M}_\ell^{\mathcal{B}}$ with $\ell \in \{1, \ldots, L\}$. Following the formalism for statistical analysis in the SPD manifold presented in Chapter 2, we parametrize the statistical distributions of the data points in $\mathcal{A}$ and $\mathcal{B}$ as

$$\Theta_{\mathcal{A}} \sim \left\{ \boldsymbol{M}^{\mathcal{A}}, \boldsymbol{M}_1^{\mathcal{A}}, \ldots, \boldsymbol{M}_L^{\mathcal{A}}, \sigma^{\mathcal{A}} \right\} \text{ and } \Theta_{\mathcal{B}} \sim \left\{ \boldsymbol{M}^{\mathcal{B}}, \boldsymbol{M}_1^{\mathcal{B}}, \ldots, \boldsymbol{M}_L^{\mathcal{B}}, \sigma^{\mathcal{B}} \right\} \tag{5.5}$$

and our goal is to define a procedure for transforming the elements of both datasets so to make the statistical distributions of the transformed data points as close as possible according to some notion of distance to be defined. Note that if $d_{\mathcal{A}} = d_{\mathcal{B}}$, the problem reduces to that of transfer learning in the SPD manifold, which was considered in Chapter 4. However, when the datasets are defined in spaces of different dimensionality, one can not use directly the RPA to match their statistics.

Our proposal consists of two parts : first, we transform the data points in $\mathcal{A}$ and $\mathcal{B}$ so that they all become d-dimensional SPD matrices, where $d \geq \max(d_{\mathcal{A}}, d_{\mathcal{B}})$[2]; this expansion is an isometric transformation and preserves the geometry of the datasets. We denote the new datasets $\mathcal{A}^{\uparrow}$ and $\mathcal{B}^{\uparrow}$. Then, we apply the RPA on the dimension-matched matrices so to make their statistical distributions as close as possible according to a distance precised later in the text. By the end of the procedure, we have two new datasets defined on the same manifold and for which the distance between the statistical distributions has been minimized.

5.3.2 Expanding the dimensions of a SPD matrix

In what follows, we present the general problem of transforming a d'-dimensional SPD matrix into a d-dimensional SPD matrix ($d > d'$). We show how such transformation has to be defined in order to guarantee the positive definiteness of the d-dimensional matrices and how certain geometric constraints can be imposed.

Choosing how to expand. Without loss of generality, we will first assume that $d = d' + 1$, so that expanding a matrix $\boldsymbol{C} \in \mathcal{P}(d')$ amounts to defining two parameters $\boldsymbol{v} \in \mathbb{R}^{d'}$ and $\alpha \in \mathbb{R}$ in

$$\boldsymbol{C}^{\uparrow} = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{v} \\ \boldsymbol{v}^T & \alpha \end{bmatrix} \in \mathbb{R}^{(d'+1)\times(d'+1)} . \tag{5.6}$$

To guarantee that $\boldsymbol{C}^{\uparrow}$ is an element of $\mathcal{P}(d'+1)$, one can use the fact that a matrix is SPD if, and only if, all of its principal minors have positive determinants. Since $\boldsymbol{C}$ is SPD, the determinant of all of its principal minors are positive, so we can conclude that $\boldsymbol{C}^{\uparrow}$ will be SPD if, and only if, its determinant is positive. From basic matrix analysis, we have that

$$\det\left(\begin{bmatrix} \boldsymbol{C} & \boldsymbol{v} \\ \boldsymbol{v}^T & \alpha \end{bmatrix}\right) = \det(\boldsymbol{C}) \det\left(\alpha - \boldsymbol{v}^T \boldsymbol{C}^{-1} \boldsymbol{v}\right) , \tag{5.7}$$

thus a necessary and sufficient condition for $\boldsymbol{C}^{\uparrow}$ being SPD is

$$\boldsymbol{v}^T \boldsymbol{C}^{-1} \boldsymbol{v} < \alpha . \tag{5.8}$$

Geometry of expanded points. Once we know the conditions for α and $\boldsymbol{v}$, the next natural question is regarding how the geometry of a set of data points $\mathcal{A} \subset \mathcal{P}(d')$ changes when its elements are expanded via (5.7) and forms a new set $\mathcal{A}^{\uparrow} \subset \mathcal{P}(d)$.

[2]Note that it may happen that the dimensions of the datasets do not describe the same features (for instance, the EEG electrodes are placed in different positions in each dataset). In this case, even if we have $d_{\mathcal{A}} = d_{\mathcal{B}}$, it is necessary to choose a d greater than both $d_{\mathcal{A}}$ and $d_{\mathcal{B}}$.

For this, we need to understand how the distance between two expanded data points in $\mathcal{P}(d')$ relates to their distance in $\mathcal{P}(d)$.

Consider we expand two SPD matrices $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$ using (5.7). We will assume that $\boldsymbol{v}$ respects condition (5.8) for both $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$, and, without loss of generality, that $\alpha = 1$. The Riemannian geodesic distance between the expanded matrices is given by

$$\delta_R^2(\boldsymbol{C}_i^{\uparrow}, \boldsymbol{C}_j^{\uparrow}) = \sum_{k=1}^{d} \log^2(\lambda_k^{\uparrow}) \;, \tag{5.9}$$

where $\text{sp}((\boldsymbol{C}_i^{\uparrow})^{-1}\boldsymbol{C}_j^{\uparrow}) = \{\lambda_1^{\uparrow}, \ldots, \lambda_d^{\uparrow}\}$ is the set of eigenvalues of $(\boldsymbol{C}_i^{\uparrow})^{-1}\boldsymbol{C}_j^{\uparrow}$. Similarly, the distance between $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$ is given by

$$\delta_R^2(\boldsymbol{C}_i, \boldsymbol{C}_j) = \sum_{k=1}^{d} \log^2(\lambda_k) \;, \tag{5.10}$$

where $\text{sp}(\boldsymbol{C}_i^{-1}\boldsymbol{C}_j) = \{\lambda_1, \ldots, \lambda_d\}$. Our goal is to be able to write $\delta_R^2(\boldsymbol{C}_i^{\uparrow}, \boldsymbol{C}_j^{\uparrow})$ in terms of $\delta_R^2(\boldsymbol{C}_i, \boldsymbol{C}_j)$. For this, we write explicitly the expression for the expanded matrix

$$(\boldsymbol{C}_i^{\uparrow})^{-1}\boldsymbol{C}_j^{\uparrow} = \begin{bmatrix} \boldsymbol{C}_i^{-1}\boldsymbol{C}_j + \boldsymbol{C}_i^{-1}\boldsymbol{v}\boldsymbol{v}^T & \dfrac{\boldsymbol{C}_i^{-1}\boldsymbol{C}_j - \boldsymbol{I}_{d'}}{1 - \boldsymbol{v}^T\boldsymbol{C}_i^{-1}\boldsymbol{v}} & \boldsymbol{0}_{d'\times 1} \\ \boldsymbol{v}^T(\boldsymbol{I}_{d'} - \boldsymbol{C}_i^{-1}\boldsymbol{C}_j) & & 1 \end{bmatrix} \;, \tag{5.11}$$

where $\boldsymbol{0}_{r\times s}$ is a $r \times s$ dimensional matrix filled with zeros and $\boldsymbol{I}_{d'}$ is a d'-dimensional Identity matrix. Because of the block structure of $(\boldsymbol{C}_i^{\uparrow})^{-1}\boldsymbol{C}_j^{\uparrow}$, it is easy to see that

$$\text{sp}\Big((\boldsymbol{C}_i^{\uparrow})^{-1}\boldsymbol{C}_j^{\uparrow}\Big) = \{1\} \;\cup\; \text{sp}\Big(((\boldsymbol{C}_i^{\uparrow})^{-1}\boldsymbol{C}_j^{\uparrow})_{\text{UL}}\Big) \;, \tag{5.12}$$

where $((\boldsymbol{C}_i^{\uparrow})^{-1}\boldsymbol{C}_j^{\uparrow})_{\text{UL}}$ is the upper-left block of $(\boldsymbol{C}_i^{\uparrow})^{-1}\boldsymbol{C}_j^{\uparrow}$.

Different choices of $\boldsymbol{v}$ lead to different $\text{sp}((\boldsymbol{C}_i^{\uparrow})^{-1}\boldsymbol{C}_j^{\uparrow})$ and, consequently, different relations between $\delta_R^2(\boldsymbol{C}_i, \boldsymbol{C}_j)$ and $\delta_R^2(\boldsymbol{C}_i^{\uparrow}, \boldsymbol{C}_j^{\uparrow})$. A particularly interesting case is when $\boldsymbol{v} = \boldsymbol{0}_{d'\times 1}$, so that

$$(\boldsymbol{C}_i^{\uparrow})^{-1}\boldsymbol{C}_j^{\uparrow} = \begin{bmatrix} \boldsymbol{C}_i^{-1}\boldsymbol{C}_j & \boldsymbol{0}_{d'\times 1} \\ \boldsymbol{0}_{1\times d'} & 1 \end{bmatrix} \;, \tag{5.13}$$

and, consequently,

$$\text{sp}\Big((\boldsymbol{C}_i^{\uparrow})^{-1}\boldsymbol{C}_j^{\uparrow}\Big) = \{1\} \;\cup\; \text{sp}(\boldsymbol{C}_i^{-1}\boldsymbol{C}_j) \;. \tag{5.14}$$

We have then,

$$\delta_R^2(\boldsymbol{C}_i^{\uparrow}, \boldsymbol{C}_j^{\uparrow}) = \sum_{k=1}^{d} \log^2(\lambda_k^{\uparrow}) = \sum_{k=1}^{d'} \log^2(\lambda_k) + \log^2(1) = \delta_R^2\left(\boldsymbol{C}_i, \boldsymbol{C}_j\right) \ , \tag{5.15}$$

which means that the expansion preserves the pairwise distances from the datasets in $\mathcal{P}(d')$ in the new space $\mathcal{P}(d)$. Furthermore, this choice of $\boldsymbol{v}$ ensures that (5.8) is verified for any positive α and any pair of matrices $\boldsymbol{C}_i, \boldsymbol{C}_j \in \mathcal{P}(d')$.

By induction, one can easily show that the same results hold for any $d' > d$ and an expansion given by

$$\boldsymbol{C}^{\uparrow} = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{0}_{d' \times p} \\ \boldsymbol{0}_{p \times d'} & \boldsymbol{I}_p \end{bmatrix} \ , \tag{5.16}$$

where $p = d' - d$.

An isometric transformation. From the results above, we see that transformation

$$\begin{aligned} E_{d' \to d} \ : \ \mathcal{P}(d') \ &\to \quad \mathcal{P}(d) \\ \boldsymbol{C} \quad &\mapsto \begin{bmatrix} \boldsymbol{C} & \boldsymbol{0}_{d' \times p} \\ \boldsymbol{0}_{p \times d'} & \boldsymbol{I}_p \end{bmatrix} , \end{aligned} \tag{5.17}$$

with $p = d - d'$, is an isometric transformation between manifolds $\mathcal{P}(d')$ and $\mathcal{P}(d)$ in terms of the AIRM distance between SPD matrices, that is,

$$\delta_R^2\Big(E_{d' \to d}(\boldsymbol{C}_i), E_{d' \to d}(\boldsymbol{C}_j)\Big) = \delta_R^2\left(\boldsymbol{C}_i, \boldsymbol{C}_j\right) \ . \tag{5.18}$$

An interesting consequence is that classification algorithms that use distances between data points as features (e.g., the MDM classifier) have exactly the same performance when applied to the data points in $\mathcal{P}(d')$ or to their transformed version in $\mathcal{P}(d)$. Therefore, we are ensured that the dimensionality augmentation does not affect (negatively nor positively) the discriminatory power of classifiers over the transformed datasets.

Statistics of the expanded data points. Consider a set of SPD data points

$$\mathcal{A} = \left\{\boldsymbol{C}_1, \ldots, \boldsymbol{C}_{K_{\mathcal{A}}}\right\} \subset \mathcal{P}(d') \ , \tag{5.19}$$

with geometric mean $\boldsymbol{M}^{\mathcal{A}}$ and dispersion $\sigma^{\mathcal{A}}$. Expanding each element of $\mathcal{A}$, we obtain a new set of SPD matrices

$$\mathcal{A}^{\uparrow} = \left\{\boldsymbol{C}_1^{\uparrow}, \ldots, \boldsymbol{C}_{K_{\mathcal{A}}}^{\uparrow}\right\} \subset \mathcal{P}(d) \ , \tag{5.20}$$

where $\boldsymbol{C}_k^{\uparrow} = E_{d' \to d}(\boldsymbol{C}_k)$. The definition of the geometric mean of $\mathcal{A}^{\uparrow}$ is

$$\boldsymbol{M}^{\mathcal{A}^{\uparrow}} = \underset{\boldsymbol{M}^{\uparrow} \in \mathcal{P}(d)}{\operatorname{argmin}} \sum_{k=1}^{K_{\mathcal{S}}} \delta_R^2(\boldsymbol{M}^{\uparrow}, \boldsymbol{C}_k^{\uparrow}) \,, \tag{5.21}$$

$$= \underset{\begin{bmatrix} \boldsymbol{M} & \boldsymbol{0}_{d' \times p} \\ \boldsymbol{0}_{p \times d'} & \boldsymbol{I}_p \end{bmatrix} \in \mathcal{P}(d)}{\operatorname{argmin}} \sum_{k=1}^{K_{\mathcal{A}}} \delta_R^2(\boldsymbol{M}, \boldsymbol{C}_k) \,, \tag{5.22}$$

$$= \begin{bmatrix} \boldsymbol{M}^{\mathcal{A}} & \boldsymbol{0}_{d' \times p} \\ \boldsymbol{0}_{p \times d'} & \boldsymbol{I}_p \end{bmatrix} \,, \tag{5.23}$$

where $p = d - d'$. This is to show that the geometric mean of the expanded matrices can be written directly in terms of the geometric mean of the original matrices, such as

$$\boldsymbol{M}^{\mathcal{A}^{\uparrow}} = E_{d' \to d}(\boldsymbol{M}^{\mathcal{A}}) \,. \tag{5.24}$$

The dispersion around $\boldsymbol{M}^{\mathcal{A}^{\uparrow}}$ is

$$\left(\sigma^{\mathcal{A}^{\uparrow}}\right)^2 = \frac{1}{K_{\mathcal{A}}} \sum_{k=1}^{K_{\mathcal{A}}} \delta_R^2(\boldsymbol{M}^{\mathcal{A}^{\uparrow}}, \boldsymbol{C}_k^{\uparrow}) \,, \tag{5.25}$$

$$= \frac{1}{K_{\mathcal{A}}} \sum_{k=1}^{K_{\mathcal{A}}} \delta_R^2\Big(E_{d' \to d}(\boldsymbol{M}^{\mathcal{A}}), E_{d' \to d}(\boldsymbol{C}_k)\Big) \,, \tag{5.26}$$

$$= \frac{1}{K_{\mathcal{A}}} \sum_{k=1}^{K_{\mathcal{A}}} \delta_R^2\left(\boldsymbol{M}^{\mathcal{A}}, \boldsymbol{C}_k\right) \,, \tag{5.27}$$

$$= \left(\sigma^{\mathcal{A}}\right)^2 \,, \tag{5.28}$$

which is a direct consequence of the isometric property of transformation (5.17). Note that if each element of $\mathcal{A}$ had a class label associated to it, the class means of their expanded counterparts would be determined as in (5.24). We conclude that if $\mathcal{A}$ is parametrized as

$$\Theta_{\mathcal{A}} \sim \left\{\boldsymbol{M}^{\mathcal{A}}, \boldsymbol{M}_1^{\mathcal{A}}, \ldots, \boldsymbol{M}_L^{\mathcal{A}}, \sigma^{\mathcal{A}}\right\} \,, \tag{5.29}$$

then

$$\Theta_{\mathcal{A}^{\uparrow}} \sim \left\{E_{d' \to d}(\boldsymbol{M}^{\mathcal{A}}), E_{d' \to d}(\boldsymbol{M}_1^{\mathcal{A}}), \ldots, E_{d' \to d}(\boldsymbol{M}_L^{\mathcal{A}}), \sigma^{\mathcal{A}}\right\} \,. \tag{5.30}$$

5.3.3 Matching the statistics of two datasets

Expanding the $d_{\mathcal{A}}$-dimensional data points from $\mathcal{A}$ and the $d_{\mathcal{B}}$-dimensional data points from $\mathcal{B}$ yields two new datasets,

$$\mathcal{A}^{\uparrow} = \left\{ (\boldsymbol{C}_i^{\mathcal{A}^{\uparrow}},\ \ell_i^{\mathcal{A}}) \text{ for } i = 1, \ldots, K_{\mathcal{A}} \right\} \text{ and } \mathcal{B}^{\uparrow} = \left\{ (\boldsymbol{C}_i^{\mathcal{B}^{\uparrow}},\ \ell_i^{\mathcal{B}}) \text{ for } i = 1, \ldots, K_{\mathcal{B}} \right\}, \tag{5.31}$$

where the $\boldsymbol{C}_i^{\mathcal{A}^{\uparrow}}$ and $\boldsymbol{C}_i^{\mathcal{B}^{\uparrow}}$ are all d-dimensional SPD matrices. The next step is to transform the elements of each dataset so that their statistical distributions, $\Theta_{\mathcal{A}^{\uparrow}}$ and $\Theta_{\mathcal{B}^{\uparrow}}$, get as close as possible. To do so, we use the Riemannian Procrustes analysis RPA, which was thoroughly discussed in Chapter 4 and we recapitulate now using a notation that is more adapted for this chapter:

(1) **Re-center** the data points in $\mathcal{A}^{\uparrow}$ and $\mathcal{B}^{\uparrow}$ such as

$$\boldsymbol{C}_i^{\mathcal{A}^{\uparrow}(\text{rct})} = \left(\boldsymbol{M}^{\mathcal{A}^{\uparrow}}\right)^{-1/2} \boldsymbol{C}_i^{\mathcal{A}^{\uparrow}} \left(\boldsymbol{M}^{\mathcal{A}^{\uparrow}}\right)^{-1/2}, \tag{5.32}$$

$$\boldsymbol{C}_i^{\mathcal{B}^{\uparrow}(\text{rct})} = \left(\boldsymbol{M}^{\mathcal{B}^{\uparrow}}\right)^{-1/2} \boldsymbol{C}_i^{\mathcal{B}^{\uparrow}} \left(\boldsymbol{M}^{\mathcal{B}^{\uparrow}}\right)^{-1/2}. \tag{5.33}$$

This forms two new datasets, $\mathcal{A}^{\uparrow(\text{rct})}$ and $\mathcal{B}^{\uparrow(\text{rct})}$, whose statistical distributions are parametrized by

$$\Theta_{\mathcal{A}^{\uparrow(\text{rct})}} \sim \left\{ \boldsymbol{I}_d, \boldsymbol{M}_1^{\mathcal{A}^{\uparrow(\text{rct})}}, \ldots, \boldsymbol{M}_L^{\mathcal{A}^{\uparrow(\text{rct})}}, \sigma^{\mathcal{A}} \right\}, \tag{5.34}$$

$$\Theta_{\mathcal{B}^{\uparrow(\text{rct})}} \sim \left\{ \boldsymbol{I}_d, \boldsymbol{M}_1^{\mathcal{B}^{\uparrow(\text{rct})}}, \ldots, \boldsymbol{M}_L^{\mathcal{B}^{\uparrow(\text{rct})}}, \sigma^{\mathcal{B}} \right\}. \tag{5.35}$$

(2) **Stretch** the dispersion around the mean for the points in $\mathcal{A}^{\uparrow(\text{rct})}$ and $\mathcal{B}^{\uparrow(\text{rct})}$ so that they are equal to one, as

$$\boldsymbol{C}_i^{\mathcal{A}^{\uparrow(\text{rct+str})}} = \left(\boldsymbol{C}_i^{\mathcal{A}^{\uparrow(\text{rct})}}\right)^{1/\sigma^2_{\mathcal{A}}}, \tag{5.36}$$

$$\boldsymbol{C}_i^{\mathcal{B}^{\uparrow(\text{rct+str})}} = \left(\boldsymbol{C}_i^{\mathcal{B}^{\uparrow(\text{rct})}}\right)^{1/\sigma^2_{\mathcal{B}}}. \tag{5.37}$$

This yields two new datasets $\mathcal{A}^{\uparrow(\text{rct+str})}$ and $\mathcal{B}^{\uparrow(\text{rct+str})}$ with equal dispersions and distributions parametrized as

$$\Theta_{\mathcal{A}^{\uparrow(\text{rct+str})}} \sim \left\{ \boldsymbol{I}_d, \boldsymbol{M}_1^{\mathcal{A}^{\uparrow(\text{rct+str})}}, \ldots, \boldsymbol{M}_L^{\mathcal{A}^{\uparrow(\text{rct+str})}}, 1 \right\}, \tag{5.38}$$

$$\Theta_{\mathcal{B}^{\uparrow(\text{rct+str})}} \sim \left\{ \boldsymbol{I}_d, \boldsymbol{M}_1^{\mathcal{B}^{\uparrow(\text{rct+str})}}, \ldots, \boldsymbol{M}_L^{\mathcal{B}^{\uparrow(\text{rct+str})}}, 1 \right\}. \tag{5.39}$$

(3) **Rotate** the data points from $\mathcal{B}^{\uparrow(\text{rct+str})}$ to make its class means as close as possible to the class means of $\mathcal{A}^{\uparrow(\text{rct+str})}$. We have then

$$\boldsymbol{C}_i^{\mathcal{A}^{\uparrow(\text{rct+str+rot})}} = \boldsymbol{C}_i^{\mathcal{A}^{\uparrow(\text{rct+str})}} , \tag{5.40}$$

$$\boldsymbol{C}_i^{\mathcal{B}^{\uparrow(\text{rct+str+rot})}} = \boldsymbol{U}^T \boldsymbol{C}_i^{\mathcal{B}^{\uparrow(\text{rct+str})}} \boldsymbol{U} , \tag{5.41}$$

with $\boldsymbol{U}$ obtained from the optimization problem

$$\underset{\boldsymbol{U}^T\boldsymbol{U}=\boldsymbol{I}_d}{\text{minimize}} \sum_{c=1}^{L} \delta_R^2 \left(\boldsymbol{U}^T \boldsymbol{M}_c^{\mathcal{B}^{\uparrow(\text{rct+str})}} \boldsymbol{U},\ \boldsymbol{M}_c^{\mathcal{A}^{\uparrow(\text{rct+str})}} \right) . \tag{5.42}$$

(4) Form two new datasets

$$\mathcal{A}^{\uparrow(\text{RPA})} = \left\{ (\boldsymbol{C}_i^{\mathcal{A}^{\uparrow(\text{rct+str+rot})}},\ \ell_i^{\mathcal{A}}) \text{ for } i = 1, \ldots, K_{\mathcal{A}} \right\} , \tag{5.43}$$

and

$$\mathcal{B}^{\uparrow(\text{RPA})} = \left\{ (\boldsymbol{C}_i^{\mathcal{B}^{\uparrow(\text{rct+str+rot})}},\ \ell_i^{\mathcal{B}}) \text{ for } i = 1, \ldots, K_{\mathcal{B}} \right\} . \tag{5.44}$$

By the end of the RPA procedure, we have two transformed datasets whose statistical distributions are closer as compared to their original versions. This implies that the distance between the two statistical distributions decreases at each step, as per

$$\mathcal{W}^2(\mu_{\mathcal{A}^{\uparrow(\text{RPA})}}, \mu_{\mathcal{B}^{\uparrow(\text{RPA})}}) \leq \mathcal{W}^2(\mu_{\mathcal{A}^{\uparrow(\text{rct+str})}}, \mu_{\mathcal{B}^{\uparrow(\text{rct+str})}}) \leq \mathcal{W}^2(\mu_{\mathcal{A}^{\uparrow}}, \mu_{\mathcal{B}^{\uparrow}}) , \tag{5.45}$$

where

$$\mathcal{W}^2(\Theta_{\mathcal{A}}, \Theta_{\mathcal{B}}) = \delta_R^2(\boldsymbol{M}^{\mathcal{A}}, \boldsymbol{M}^{\mathcal{B}}) + \sum_{\ell=1}^{L} \delta_R^2(\boldsymbol{M}_\ell^{\mathcal{A}}, \boldsymbol{M}_\ell^{\mathcal{B}}) + \log^2 \left(\frac{\sigma^{\mathcal{A}}}{\sigma^{\mathcal{B}}} \right) . \tag{5.46}$$

5.3.4 Summary of the method

Dimensionality transcending may be summed up as the application of transformations,

$$T_{\mathcal{A}} : \mathcal{P}(d_{\mathcal{A}}) \to \mathcal{P}(d) \ \text{ and } \ T_{\mathcal{B}} : \mathcal{P}(d_{\mathcal{B}}) \to \mathcal{P}(d) , \tag{5.47}$$

to the data points of $\mathcal{A}$ and $\mathcal{B}$, forming

$$\widetilde{\mathcal{A}} = \left\{ (\boldsymbol{C}_i^{\widetilde{\mathcal{A}}},\ \ell_i^{\mathcal{A}}) \text{ for } i = 1, \ldots, K_{\mathcal{A}} \right\} \ \text{ and } \ \widetilde{\mathcal{B}} = \left\{ (\boldsymbol{C}_i^{\widetilde{\mathcal{B}}},\ \ell_i^{\mathcal{B}}) \text{ for } i = 1, \ldots, K_{\mathcal{B}} \right\} , \tag{5.48}$$

with

$$\boldsymbol{C}_i^{\widetilde{\mathcal{A}}} = T_{\mathcal{A}} \left(\boldsymbol{C}_i^{\mathcal{A}} \right) \in \mathcal{P}(d) \ \text{ and } \ \boldsymbol{C}_i^{\widetilde{\mathcal{B}}} = T_{\mathcal{B}} \left(\boldsymbol{C}_i^{\mathcal{B}} \right) \in \mathcal{P}(d) , \tag{5.49}$$

where $d \geq \max\{d_{\mathcal{A}}, d_{\mathcal{B}}\}$. Transformations $T_{\mathcal{A}}$ and $T_{\mathcal{B}}$ are formed by the composition of two operations, a dimensionality augmentation step (described in Section 5.3.2) followed by a distribution matching step (described in Section 5.3.3).

5.4 Application to BCI datasets

This section describes two pratical problems where dimensionality transcending proves useful. Both examples are related to classification tasks with EEG signals from brain computer interfaces, but they can be expanded to other types of multivariate time series as well. We use the Riemannian geometric framework detailed in Chapter 2 to parametrize the statistics of the EEG epochs via symmetric positive definite matrices. The first example concerns the case when one (or several) electrode presents a problem during an EEG recording and the signal it records has to be rejected due to, for instance, high amplitudes, low signal-to-noise ratio, etc. In this situation, one could either simply discard the problematic trial or try to fill the missing data with statistically relevant information. Dimensionality transcending is an approach for the latter option, which is often called *data imputation* in the literature. The second example considers the case when we have datasets from BCI recordings containing different numbers and/or positions of electrodes, yet, we would like to use information from one dataset to improve the classification performance of trials on the other dataset (transfer learning). Dimensionality transcending provides a way to do this.

5.4.1 Data imputation

In EEG experiments, it is not uncommon that the recording at some electrodes present problems. When this happens, the simplest thing to do is to reject the problematic trial. However, in BCI experiments, each data point is the recording of the EEG activity of a subject that may last a few seconds. In this case, discarding a trial because of only one or a few malfunctioning electrodes is not desirable.

The solution we work out here is to replace the corrupted measurements of problematic trials with information that is statistically justified. Such approach is commonly known as *missing-data imputation* [LR02] and is based on the idea of filling the missing values of data points in a way that preserves the statistics of the full dataset.

Consider we have two datasets, $\mathcal{A}$ and $\mathcal{B}$, as defined in (5.52). Each data point $\boldsymbol{C}_i^{\mathcal{A}}$ is a spatial covariance matrix estimated from a time series recorded over $d_{\mathcal{A}}$-electrodes, which we denote $\boldsymbol{X}_i^{\mathcal{A}}$. The data points from $\mathcal{B}$ are $d_{\mathcal{B}}$-dimensional SPD matrices estimated from trials where p electrodes presented problems ($d_{\mathcal{B}} = d_{\mathcal{A}} - p$) and are denoted $\boldsymbol{C}_i^{\mathcal{B}}$. We argue that dimensionality transcending can be used to transform the data points in $\mathcal{B}$ so that they become matrices defined in a SPD manifold of the

same dimension as $\mathcal{A}$ and whose statistical distribution is close to $\Theta_{\mathcal{A}}$ according to distance (5.46).

Parameter estimation. Dimensionality transcending relies on the estimation of parameters for describing the statistics of datasets $\mathcal{A}$ and $\mathcal{B}$. However, in general one has access to just a few data points in $\mathcal{B}$, since the problematic trials are assumed to be not too numerous. Consequently, one can expect rather poor estimates of the statistical parameters that describe $\Theta_{\mathcal{B}}$. To cope with this limitation, we discard the same p problematic electrodes that define the elements in $\mathcal{B}$ from all the $d_{\mathcal{A}}$-dimensional data points in $\mathcal{A}$ and estimate the spatial covariance matrices of these reduced time series to form a new dataset $\mathcal{A}^{(-p)} \subset \mathcal{P}(d_{\mathcal{B}})$. Then, we estimate the parameters that describe the statistics of $\mathcal{A}^{(-p)}$ and use them as descriptors for $\Theta_{\mathcal{B}}$. This procedure relies on the assumption that the statistics for datasets $\mathcal{A}^{(-p)}$ and $\mathcal{B}$ are similar to each other, which is justified by the fact that $\mathcal{B}$ was obtained during the same experiment that generated the data points from $\mathcal{A}$.

Time series interpretation. It is interesting to note that the imputation method can be interpreted as filling p dimensions of a problematic multivariate time series $\boldsymbol{X} \in \mathbb{R}^{(d_{\mathcal{A}}-p)\times T}$ in a way that the second-order statistics of its expanded counterpart, $\boldsymbol{X}^{\uparrow} \in \mathbb{R}^{d_{\mathcal{A}}\times T}$, has some particular structure. Defining

$$\boldsymbol{X}^{\uparrow} = \begin{bmatrix} \boldsymbol{X} \\ \mathbf{x}_p \end{bmatrix} \in \mathbb{R}^{d_{\mathcal{A}}\times T} , \tag{5.50}$$

where $\mathbf{x}_p \in \mathbb{R}^{p\times T}$ is a T-sample realization of a p-dimensional time series with zero mean and spatial covariance $\boldsymbol{I}_p$, we have that

$$\boldsymbol{C}^{\uparrow} = E_{(d_{\mathcal{A}}-p)\to d_{\mathcal{A}}}(\boldsymbol{C}) , \tag{5.51}$$

where $\boldsymbol{C}$ and $\boldsymbol{C}^{\uparrow}$ are covariance matrices estimated from $\boldsymbol{X}$ and $\boldsymbol{X}^{\uparrow}$, respectively, and $E_{d\to d'}$ is an isometric transformation from $\mathcal{P}(d)$ to $\mathcal{P}(d')$ defined in (5.17).

An example. Suppose we have a dataset consisting of 4-dimensional EEG epochs such that the dimensions of the multivariate time series correspond to electrodes $\{\texttt{Fz}, \texttt{C3}, \texttt{C4}, \texttt{Pz}\}$, in this exact order (see Figure 5.1 for a visual depiction of the position of these electrodes). We denote the set of time series epochs by $\mathcal{A}'$ and estimate the spatial covariance matrices that parametrize their statistics, forming dataset $\mathcal{A} \subset \mathcal{P}(4)$. Suppose, now, that we have an EEG epoch presenting problems in the dimension corresponding to the signal recorded at electrode `C3`. The SPD parametrization of this epoch defines a set $\mathcal{B} \subset \mathcal{P}(d_{\mathcal{A}} - p)$, with $d_{\mathcal{A}} - p = 3$. We create a new dataset, $\mathcal{A}^{(-1)} \subset \mathcal{P}(3)$, consisting of SPD data points which parametrize the statistics of the time series from $\mathcal{A}'$ that had their second dimension discarded (dimension corresponding to electrode `C3`). We estimate the parameters describing

the statistics of $\mathcal{A}^{(-1)}$ (that is, its full geometric mean, the class means, and the dispersion) and use them as descriptors for the statistical distribution of the set $\mathcal{B}$. Finally, we apply the dimensionality transcending procedure to $\mathcal{A}$ and $\mathcal{B}$ to match their dimensionalities and statistical distributions. At the end of the procedure, we have a transformed version of the SPD matrix that describes the statistics of the problematic EEG epoch; it lives in $\mathcal{P}(4)$ even though the time series is only 3-dimensional.

5.4.2 Matching datasets

Another use for dimensionality transcending is when gathering information from experiments registered under the same BCI paradigm but with different electrode configurations (e.g. different number of electrodes, different electrode positions). This is interesting in practice because it allows working with datasets recorded with different experimental setups in an unified way, sharing the information from one dataset to classify the trials from another dataset. Note, however, that the dimensionality transcending method does not add any new information : if the electrodes originally chosen for a certain dataset do not have any discriminatory power for a given BCI task, expanding the dimensions of the data points will not improve the performance of classifiers trained on them.

In order to keep the exposition simple, we will consider the case with just two datasets

$$\mathcal{A} = \left\{ (\boldsymbol{C}_i^{\mathcal{A}},\ \ell_i^{\mathcal{A}}) \text{ for } i = 1, \dots, K_{\mathcal{A}} \right\} \text{ and } \mathcal{B} = \left\{ (\boldsymbol{C}_i^{\mathcal{B}},\ \ell_i^{\mathcal{B}}) \text{ for } i = 1, \dots, K_{\mathcal{B}} \right\}. \tag{5.52}$$

with data points $\boldsymbol{C}_i^{\mathcal{A}} \in \mathcal{P}(d_{\mathcal{A}})$ and $\boldsymbol{C}_i^{\mathcal{B}} \in \mathcal{P}(d_{\mathcal{B}})$, and labels $\ell_i^{\mathcal{A}}, \ell_i^{\mathcal{B}} \in \{1, \dots, L\}$. Sets $\mathcal{E}_{\mathcal{A}}$ and $\mathcal{E}_{\mathcal{B}}$ contain the names and positions of the electrodes used for the recordings in each dataset.

Suppose, at first, that $d_{\mathcal{A}} \leq d_{\mathcal{B}}$ and $\mathcal{E}_{\mathcal{A}} \subseteq \mathcal{E}_{\mathcal{B}}$. To match the datasets, we first use permutation matrices to make sure that the order of the electrodes in the $d_{\mathcal{A}}$ dimensions of each trial in $\mathcal{A}$ is the same as for the first $d_{\mathcal{A}}$ dimensions in $\mathcal{B}$. Then, we apply the dimensionality augmentation procedure to the elements in $\mathcal{A}$ so that they become $d_{\mathcal{B}}$-dimensional SPD matrices. Finally, we use RPA to match the statistics of both datasets. Note that this situation is very similar to the one described for data imputation in Section 5.4.1, however, here we assume that there are enough data points in $\mathcal{B}$ for estimating its statistics.

A slightly more complicated case is when $d_{\mathcal{A}} \leq d_{\mathcal{B}}$ and $\overline{\mathcal{E}_{\mathcal{A}} \cap \mathcal{E}_{\mathcal{B}}} \neq \{\}$, which corresponds to when $\mathcal{A}$ has electrodes that are not present in $\mathcal{B}$ and vice-versa. To match the datasets in this case, first we define a new set $\mathcal{E} = \mathcal{E}_{\mathcal{A}} \cup \mathcal{E}_{\mathcal{B}}$ and use permutation matrices to assure that the order of the dimensions of the trials in $\mathcal{A}$ and

$\mathcal{B}$ are the same as that for an arbitrarily chosen order for $\mathcal{E}$. Then, we augment the data points in $\mathcal{A}$ and $\mathcal{B}$ so that they become d-dimensional SPD matrices ($d = |\mathcal{E}|$) and apply RPA to match their statistical distributions.

An example. Suppose we have two datasets, $\mathcal{A}$ and $\mathcal{B}$, consisting of 4-dimensional and 3-dimensional EEG epochs, respectively. The electrode sets for these two datasets are $\mathcal{E}_{\mathcal{A}} = \{\texttt{Fz}, \texttt{C3}, \texttt{C4}, \texttt{Pz}\}$ and $\mathcal{E}_{\mathcal{B}} = \{\texttt{C4}, \texttt{C3}, \texttt{Cz}\}$, with the ordering of the names of the electrodes corresponding exactly to the ordering of the dimensions of the EEG epochs. Our dimension matching procedure starts by defining a new set $\mathcal{E} = \mathcal{E}_{\mathcal{A}} \cup \mathcal{E}_{\mathcal{B}} = \{\texttt{C3}, \texttt{C4}, \texttt{Fz}, \texttt{Cz}, \texttt{Pz}\}$ whose order is considered as fixed. Then, we apply the dimensionality augmentation step explained in Section 5.3.2 to the data points in both $\mathcal{A}$ and $\mathcal{B}$, so to have new SPD matrices defined in $\mathcal{P}(d)$, with $d = |\mathcal{E}| = 5$. If necessary, we change the ordering of the dimensions of the augmented matrices so that they correspond to the order imposed by $\mathcal{E}$ (this ensures that the dimensions from the expanded versions of $\mathcal{A}$ and $\mathcal{B}$ are comparable). Finally, we use the Riemannian Procrustes analysis to match the statistics of the dimension-augmented datasets.

5.5 Numerical illustrations

In this section, we illustrate the two pratical problems presented in the previous section on real EEG recordings. For data imputation, we consider a motor-imagery BCI dataset and simulate different situations where the signals on one (or several) of the electrodes are discarded. We compare the performance of dimensionality transcending with that of *spherical spline interpolation* [Per+89], the state of the art method for replacing missing values in problematic EEG channels, which takes appropriately weighted linear combinations of signals from electrodes located near to the problematic channel. We illustrate the case of datasets defined on different sets of electrodes using recordings from the motor imagery and P300 paradigms. All datasets used in this section are publicly available on the `MOABB` framework [JB18].

5.5.1 Data imputation

The dataset. We use the database `Cho2017` containing electroencephalographic (EEG) recordings of an experiment with a brain-computer interface (this database has already been used in previous chapters of this thesis; see [Cho+17] for its full reference). The database contains recordings on 23 electrodes (selected out of 64) of 52 subjects executing a left-hand/right-hand motor imagery paradigm. The signals are bandpass filtered between 8 Hz and 35 Hz (sampling frequency is 512 Hz) and epoched into one hundred 3-second trials: 50 trials on the left-hand class and 50

trials on the right-hand class. Such pre-processing yields for each subject a set of EEG epochs

$$\mathcal{A}' = \{\boldsymbol{X}_1^{\mathcal{A}}, \dots, \boldsymbol{X}_{K_{\mathcal{A}}}^{\mathcal{A}}\} \subset \mathbb{R}^{d \times T} \tag{5.53}$$

where $d = 23$, $T = 1536$, and $K_{\mathcal{A}} = 100$. For each element in $\mathcal{A}'$ we estimate a spatial covariance matrix using Ledoit-Wolf shrinkage [LW04], which helps controlling the numerical conditioning of the estimated matrix. The set of spatial covariances forms the dataset

$$\mathcal{A} = \{\boldsymbol{C}_1^{\mathcal{A}}, \dots, \boldsymbol{C}_{K_{\mathcal{A}}}^{\mathcal{A}}\} \subset \mathcal{P}(d) \, . \tag{5.54}$$

An epoch with problems on p electrodes is a data point in $\mathbb{R}^{(d-p)\times T}$. Without loss of generality, we will consider that the dimensions related to these discarded electrodes correspond to the p last dimensions of the data points in $\mathcal{A}'$. The spatial covariance matrix estimated from the problematic epoch is an element of $\mathcal{B} \subset \mathcal{P}(d-p)$.

Classification procedure. In this (and the next) sub-section, every classification task is performed using the *minimum distance to mean* classifier (MDM), which is a generalization of the nearest-centroid classifier to the space of SPD matrices [Bar+12]. It works by first estimating the geometric mean of the elements of each class in the *training* dataset (the class means). Then, it assigns to each unlabeled data point in the *testing* dataset the label of the nearest class mean according to the geodesic distance in the SPD manifold. We use the area under the ROC curve (AUC) [Bis07] as score for the classifier and report its average over 10 cross-validation folds. We denote the *training* dataset on each fold by $\mathcal{A}_{\text{train}}$ and the *testing* dataset by $\mathcal{A}_{\text{test}}$.

We consider three situations:

(1) Firstly, we assume there is no problem on any electrode and call this the **full** method. The MDM is trained on $\mathcal{A}_{\text{train}}$ and tested on $\mathcal{A}_{\text{test}}$.

(2) Then, we emulate the case when the data points in $\mathcal{A}_{\text{test}}$ have problems on a set of p arbitrarily chosen electrodes (see Table 5.1 for some examples of sets of electrodes considered as problematic). For this, we discard the problematic channels from the time series epochs whose statistics are described by the data points in $\mathcal{A}_{\text{test}}$. Then, we apply the imputation method explained in Section 5.4.1 to the SPD data points that describe the statistics of the problematic epochs. This yields a new dataset $\tilde{\mathcal{A}}_{\text{test}}$. We train the MDM classifier on $\mathcal{A}_{\text{train}}$ and test it on $\tilde{\mathcal{A}}_{\text{test}}$. We call this the **imputation** method.

(3) Finally, we proceed similarly to what was done in (2) but augment the dimensions of the problematic time series epochs via spherical spline interpolation. The SPD data points describing their statistics are then estimated and form the *testing* dataset. We call this the **interpolation** method.

We use the cross-validated AUC of the MDM classifier on each of these cases as proxy for evaluating the relevance of using dimensionality transcending for data imputation.

Results. In the results described below, we have used knowledge of the neurophysiology of BCI experiments in the motor imagery paradigm to consider settings with different combinations of EEG electrodes as problematic. We chose channels located in the motor cortex, which are known to carry important information for classifying the trials (`C3` and `C4`), as well as electrodes which are not relevant for this kind of paradigm (`Fz` and `Pz`) [Con13]. See Figure 5.1 for a representation of the spatial disposition of the 23 electrodes used for the recordings in the database.

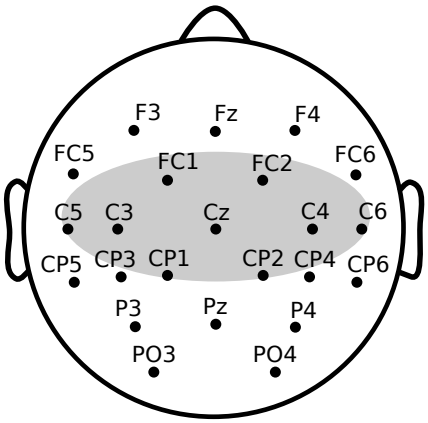

Fig. 5.1: Diagram with the electrodes configuration. The gray area indicates where the sensory motor cortex is approximately located, which is the region mostly involved in motor imagery tasks.

Table 5.1 displays the classification scores for the **imputation** and **interpolation** methods when different electrodes are considered as problematic. The score obtained with the **full** method is 0.663 and serves as a reference for our comparisons.

Tab. 5.1: Average accuracy scores for the **imputation** and **interpolation** methods over the 52 subjects in the database (standard deviation inside parenthesis). The missing electrodes column indicates which electrodes were discarded in each case. The average accuracy for the **full** method was 0.66.

missing electrodes	imputation	interpolation
{Fz}	0.66 (0.11)	0.64 (0.10)
{Pz}	0.66 (0.10)	0.63 (0.10)
{Fz, Pz}	0.66 (0.10)	0.61 (0.10)
{C3}	0.65 (0.10)	0.63 (0.10)
{C4}	0.65 (0.10)	0.61 (0.08)
{C3, C4}	0.64 (0.09)	0.61 (0.09)

We observe that the scores with the **imputation** method when only the `Fz` and/or the `Pz` electrodes are missing is not very different from that of the **full** method. In fact, a paired t-test comparing the average score for each subject of the database indicates

no evidence for rejecting the null hypothesis of equality for the two methods. Such a result is not surprising, since the referred electrodes were not expected to carry relevant information to discriminate between the classes of the experiment. However, when the `C3` and/or the `C4` are missing, the important discriminative information provided by these channels can not be replaced by our imputation method, so the average classification score decreases.

We also note that the **imputation** method consistently yields better results, on the average, as compared to the **interpolation** method. We performed paired t-tests to compare the results of the two methods and the null hypothesis of equal average scores was always rejected with p-values smaller than 10^{-3} (corrected for the multiple comparisons problem via the Bonferroni method). A possible explanation for this could be the diversity of information used by our imputation procedure as compared to the interpolation method, since it adds new dimensions to the problematic $(d-p)$-dimensional $\boldsymbol{C}^{\mathcal{B}}$ matrix using information from the rest of the dataset $\mathcal{A}$, whereas spherical spline interpolation uses only information from the time series $\boldsymbol{X}^{\mathcal{B}}$ from which $\boldsymbol{C}^{\mathcal{B}}$ is estimated. Furthermore, because the p dimensions added to $\boldsymbol{X}^{\mathcal{B}}$ are simply linear combination of its $d-p$ time series, the rank of $\boldsymbol{X}^{\mathcal{B}^{\uparrow}} \in \mathbb{R}^{d\times T}$ is just $d-p$. As a consequence, although the estimated $\boldsymbol{C}^{\mathcal{B}^{\uparrow}}$ has no zero eigenvalues (because of the Ledoit-Wolf shrinkage), some of its eigenvectors point to directions which are not descriptive and may prejudice the classification procedure.

It should be mentioned that the matrix augmentation scheme provided by our imputation method is purely based on the distribution of the spatial covariance matrices of each trial. This means that there is no physiological interpretation for the time series obtained on the p added dimensions. However, one could try to determine a physiologically plausible $\mathbf{x}_p$ in (5.50) with the statistical properties required by the imputation method. Such extension remains an open question and is one of the future perspectives for this work.

5.5.2 Matching datasets

In this second example, we consider situations involving EEG data from two BCI paradigms : motor imagery and P300. We demonstrate the relevance of dimensionality transcending through the performance of a MDM classifier trained on data from one dataset and used to classify trials from another dataset with a different set of electrodes (different number of electrodes as well as different positions over the subject's scalp).

Datasets. For the examples on the motor imagery paradigm, we use the `Zhou2016` [Zho+16] and `BNCI2015001` [Fal+12] datasets. The first dataset consists of recordings on 14 electrodes from 4 subjects executing either a left-hand/right-hand or a feet/right-

hand motor imagery task; we denote the datasets by `Zhou2016-LR` (`LR` for left-hand/right-hand) and `Zhou2016-FR` (`FR` for feet/right-hand). Dataset `BNCI2015001` is composed of EEG signals from 13 electrodes and 12 subjects (from which we have selected 7 with the best self-scores, e.g., the score of a classifier trained and tested on the same dataset), all executing a feet/right-hand motor imagery task; the two classes on both datasets are balanced. The smallest set containing the names of all electrodes from both MI datasets is of size 18. See Figure 5.2 for an illustration showing where the electrodes of each dataset are placed.

The examples on the P300 paradigm use the `BNCI2014009` [Ari+14] and `BI.2013` [Vai+18] datasets. The data in `BNCI2014009` contains EEG recordings from 16 electrodes on 10 subjects (from which we have selected the 5 with the best self-scores). The EEG signals in `BI.2013` also come from 16 electrodes but they are placed in different positions as compared to `BNCI2014009` (see Figure 5.2); we selected the best 9 subjects in terms of self-scores out of 24 available subjects. The smallest set containing the names of all electrodes from both datasets is of size 27. Note that the two datasets are from recordings on P300 experiments with a 6-by-6 grid with flashing cues, but the subjects' cognitive tasks are slightly different: in `BNCI2014009` they must concentrate on letters to spell words, whereas in `BI.2013` the subjects are asked to fix their attention on target cues representing 'aliens' to be destroyed. The classes of the trials are unbalanced, with one 'target' trial for every five 'non-target' trials.

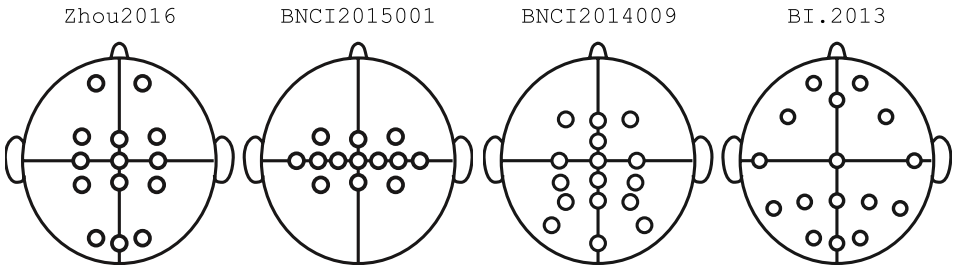

Fig. 5.2: Diagrams with the electrode configurations of the four datasets considered in this sub-section. We do not specify the names of the electrodes for visual simplicity, but the reader is referred to the references associated to each dataset for such information.

Analysis. Our goal is to show that dimensionality transcending allows a classifier to leverage from discriminative information in EEG recordings from other subjects even if they were obtained under different experimental setups. To demonstrate this, we proceed in a similar manner to what was done in Chapter 4 for illustrating the RPA method. We consider the cross-subject transfer learning case for BCI, where one wants to determine the unknown labels from a *target* dataset ($\mathcal{T}_u$) using information from a few labeled trials in the target dataset ($\mathcal{T}_\ell$), as well as the full information available from a *source* dataset ($\mathcal{S}$) containing recordings from another subject. We consider the case when both subjects are from the same database (i.e. the same experimental setup), which we call the 'intra-base case', as well as when the subjects

come from different databases (the 'inter-base' case). We compare three classification pipelines assuming that there are n_{covs} labeled data points from each class on the $\mathcal{T}_\ell$ dataset:

- **calibration**: this is when the data points in $\mathcal{T}_u$ are classified using a MDM classifier trained with only the labeled data points available in the $\mathcal{T}_\ell$ dataset,

$$\mathcal{D}_{\text{train}} = \mathcal{T}_\ell \quad \text{and} \quad \mathcal{D}_{\text{test}} = \mathcal{T}_u \; . \tag{5.55}$$

- **DT-uns**: this is when only the unsupervised steps of the RPA are used (recentering and stretching) for matching the statistics of two dimension-matched datasets. A MDM classifier is trained on a set containing the n_{covs} labeled data points from the *target* dataset as well as the dimension-matched and RPA-uns-transformed data points from a source subject,

$$\mathcal{D}_{\text{train}} = \mathcal{T}_\ell^{\uparrow(\text{rct+str})} \cup \mathcal{S}^{\uparrow(\text{rct+str})} \quad \text{and} \quad \mathcal{D}_{\text{test}} = \mathcal{T}_u^{\uparrow(\text{rct+str})} \; . \tag{5.56}$$

- **DT**: this is when the full RPA is used to match the statistics of two dimension-matched datasets. A MDM classifier is trained on a set containing the n_{covs} labeled data points from *target* dataset as well as the dimension-matched and RPA-transformed data points from a source subject,

$$\mathcal{D}_{\text{train}} = \mathcal{T}_\ell^{\uparrow(\text{RPA})} \cup \mathcal{S}^{\uparrow(\text{RPA})} \quad \text{and} \quad \mathcal{D}_{\text{test}} = \mathcal{T}_u^{\uparrow(\text{RPA})} \; . \tag{5.57}$$

We use the area under the ROC curve (AUC score) for quantifying the classification performance of the MDM classifier at each case. We randomly split the *target* dataset into labeled and unlabeled subsets five times and average the classification scores obtained in each realization. We assert that dimensionality transcending is useful for cross-subject transfer learning when the score of the **DT** pipeline is superior to that of the **calibration** pipeline, since it means that information from a *source* subject improved the classification score on a *target* dataset.

Results on motor imagery. Figure 5.3 and Figure 5.4 portray the results of the analysis described above for the two motor imagery datasets. In Figure 5.3, we used the data from subjects in `BNCI2015001` as *target* datasets and considered two different cases for the *source* datasets : data coming from subjects in the same database (the 'intra-base' case) or from the `Zhou2016-FR` database (named the 'inter-base' case). Conversely, Figure 5.4 considers subjects from `Zhou2016-FR` as *target* datasets and uses source subjects from `Zhou2016-FR` ('intra-base') or `BNCI2015001` ('inter-base') as *source* datasets. The scores of the classification pipelines on each *target* subject are displayed on different rectangular regions in which the vertical line indicates the score for the **calibration** pipeline. The four rows of scatter points in each rectangular box represent the cross-subject scores for each *source* subject and each classification pipeline (different markers indicate different classification

pipelines; see legend in the figure). The rectangular boxes are ordered according to the score of the **calibration** pipeline when the size of the labeled part of the *target* dataset is $|\mathcal{T}_\ell| = 2 \times n_{\text{covs}}$ (the datasets have two motor imagery classes). The values on the first column on the right of the axis indicate the average value of the scattered points for each *target* subject, whereas the second column indicates the score with the **calibration** pipeline.

As mentioned before, our goal is to assess whether the scores of the pipelines using dimensionality transcending are superior to that of the **calibration** pipeline. For this, we examine where the scatter points are located relative to the vertical lines indicating calibration scores. In both figures, we observe that the scores in the 'inter-base' case tend to be higher for *target* subjects where the calibration score is higher; this goes in line with what was observed in Chapter 4, where the *target* subjects with the best self-scores were also the best 'receivers' of data from *source* subjects. We also observe that, in general, the results with **DT-uns** are inferior to that of **calibration**, whereas those for **DT** are, for the most part, superior to **calibration**. Interestingly, on both figures the average performance of the classification pipelines in the 'intra-base' case (which boils down to simply using RPA to match the statistics of a *source-target* pair) is similar to that on the 'inter-base' case, which shows that transfer learning with dimensionality transcending manages to satisfactorily match datasets that a priori would be completely uncompatible.

In addition to the qualitative analysis of the results in Figure 5.3 and Figure 5.4, we also did a quantitative comparison of the pipelines' scores based on statistical hypothesis tests. The results are displayed in Table 5.2, where the average values of the classification pipelines are taken over all the cross-subject classification scores for all pairs of *source-target* subjects. We display only the comparisons for the 'inter-base' case, since the 'intra-base' case is already thoroughly discussed in Chapter 4.

To assess whether the performance of pipelines **DT-uns** and **DT** were statistically significantly different than that of pipeline **calibration**, we used paired t-tests with p-values obtained via permutation methods. The statistical procedure is similar to the one used in Chapter 4 for comparing transfer learning pipelines. To compare method **A** versus **calibration** we do :

(1) For each *target* subject i, we perform a signed paired t-test comparing the scores of method **A** to **calibration** along all *source* subjects. Each of these tests yields a statistic T_i and a p-value p_i is obtained via permutations tests [EO07].

(2) We combine the p-values of all the *target* subjects using Stouffer's Z-score method [Zay11]. This yields a single p-value for the comparison between methods as well as the direction to which the null hypothesis has been rejected (i.e., whether method **A** is better than **calibration** or vice-versa).

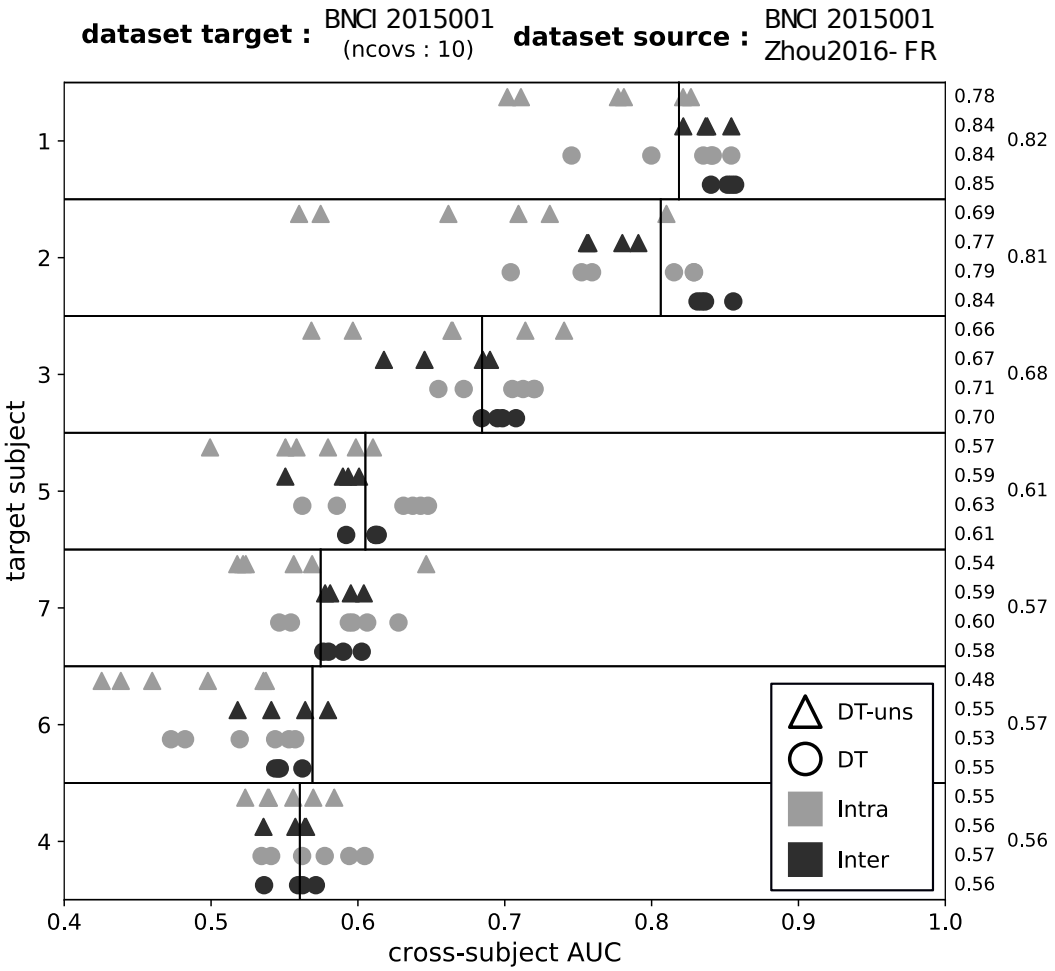

Fig. 5.3: AUC scores for cross-subject classification considering different *target* subjects in `BNCI2015001` (always with $n_{\text{covs}} = 10$ and $|\mathcal{T}_\ell| = 2 \times n_{\text{covs}} = 20$) and *source* subjects from `BNCI2015001` ('intra-base' case, represented in gray) and `Zhou2016-FR` ('inter-base' case, represented in black). The cross-subject scores for each *target* subject are represented inside rectangular boxes and the different scatter points inside them indicate the scores obtained for each *source* subject; different markers indicate different classification pipelines (triangles for **DT-uns** and circles for **DT**; see text for a description of each pipeline). The vertical line inside each rectangular box indicates the **calibration** score for the *target* subject when n_{covs} matrices are available in $\mathcal{T}_\ell$. The values on the first column to the right of the axis represent the mean AUC scores for each line of scatter points, whereas the second column contains the scores with **calibration**. Take *target* subject 1 as example: the AUC for **calibration** is 0.82 and we see that the cross-subject scores for the **DT** pipeline with *source* subjects in both the `Zhou2016-FR` database ('inter-base', black circles and average score of 0.85) and `BNCI2015001` database ('intra-base', gray circles and average score of 0.84) are almost always superior to **calibration**; the pipeline **DT-uns** performs better in the 'inter-case' (black triangles and average score of 0.84) as compared to the 'intra-base' case (gray triangles and average score of 0.78).

Tab. 5.2: Mean values of the area under the ROC curve (AUC) score for cross-subject classification using three pipelines, all described in the text; we consider only the 'inter-base' case. For each database being used as *target*, we consider a list with three values (labeled 1, 2, 3 in the table) for the size of the labeled part of the *target* dataset, $\mathcal{T}_\ell$. For `BNCI2015001` this list is $[10, 20, 50]$ and, for both `Zhou2016-FR` and `Zhou2016-LR`, it is $[5, 10, 15]$. For the datasets in the P300 paradigm the lists are $[12, 36, 48]$. Parameter i_{covs} indicates to which element of these lists the value in the grid corresponds to. The fontstyle of the average scores represented in the table are determined from the statistical tests that compare their values with that of **calibration**; see text for an explanation on the statistical procedure that we used. When the score of a pipeline is in bold, it means that it is better than **calibration** in average, whereas a classification score that is underlined indicates that the pipeline's performance is inferior to **calibration** in average; a score with no fontstyle is one that is not statistically significantly different as compared to **calibration**. The letters 'T' and 'S' on the left of the table indicate which database is used as *target* and *source* in each comparison.

		DT-uns			DT			calibration			
	icovs	1	2	3	1	2	3	1	2	3	
T:	`Zhou2016-FR`	0.66	0.71	0.76	**0.77**	0.81	0.84	0.75	0.79	0.84	Motor Imagery
S:	`BNCI2015001`										
T:	`BNCI2015001`	0.61	0.65	0.69	**0.65**	**0.67**	**0.70**	0.63	0.66	0.69	
S:	`Zhou2016-FR`										
T:	`Zhou2016-LR`	0.49	0.53	0.56	0.65	0.68	0.71	0.67	0.72	0.74	
S:	`BNCI2015001`										
T:	`BNCI2015001`	0.59	0.64	0.68	0.61	0.65	0.69	0.62	0.67	0.69	
S:	`Zhou2016-LR`										
T:	`BI.2013`	0.62	0.75	0.78	**0.75**	**0.83**	**0.84**	0.66	0.81	0.83	P300
S:	`BNCI2014009`										
T:	`BNCI2014009`	0.57	0.68	0.72	**0.69**	**0.77**	**0.78**	0.62	0.74	0.76	
S:	`BI.2013`										

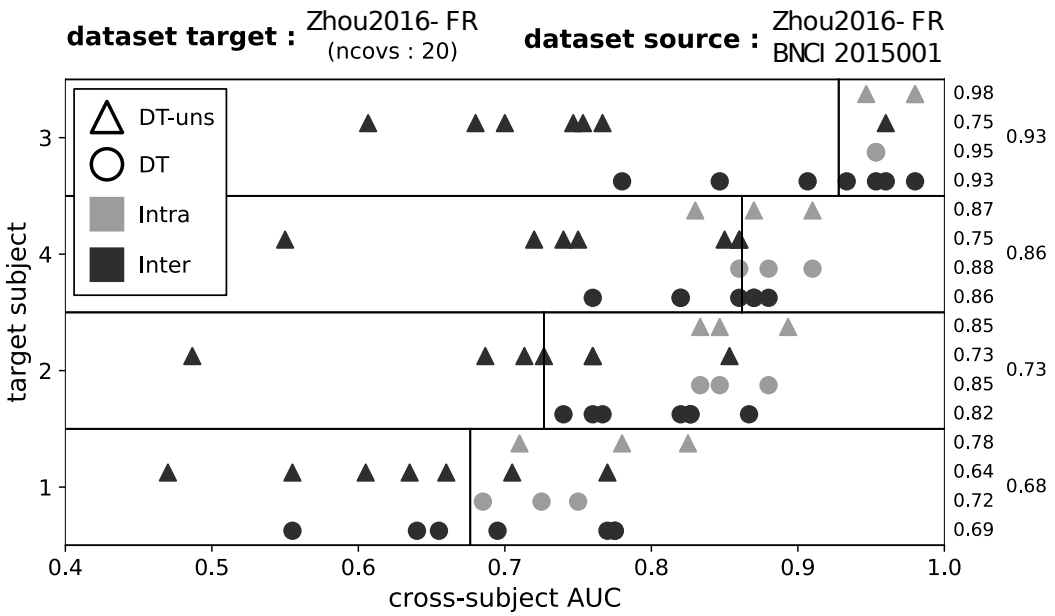

Fig. 5.4: AUC scores for cross-subject classification considering different *target* subjects in `Zhou2016-FR` (always with $n_{\text{covs}} = 20$ and $|\mathcal{T}_\ell| = 2 \times n_{\text{covs}} = 40$) and *source* subjects from `BNCI2015001` ('intra-base' case) and `Zhou2016-FR` ('inter-base' case). See the caption of Figure 5.3 for more details about the structure of the plot.

(3) We adjust the p-values of each pairwise comparison using Holm's step-down procedure [Hol79] to account for the multiple comparison problem.

Table 5.2 shows that pipeline **DT-uns** yields inferior results to that of **calibration** for different sizes of the labeled part of the *target* dataset (determined by the values of n_{covs}), being statistically equivalent in a few instances. This is not surprising, since the supervised step in the RPA method (which is missing in **DT-uns**) is closely related to how the electrodes on two datasets compare to each other. For instance, if the *source* and *target* datasets contain exactly the same data, but have the names of their electrodes in different order (and so the dimensions of the SPD matrices in different order), RPA generates a permutation matrix for correcting this mismatch. On the other hand, pipeline **DT** is better than **calibration** on most situations (or at least equivalent), showing that it is indeed a good option for leveraging discriminative information from other datasets.

We have also considered the slightly different situation where the databases do not correspond to the same cognitive task: `BNCI2015001` has trials for right-hand/feet motor imagery tasks, whereas `Zhou2016-LR` has classes left-hand/right-hand. The results in Table 5.2 show that dimensionality transcending yields poorer results as compared to before, being always inferior (or sometimes equivalent) to the pipeline **calibration**. This is an interesting result, since it shows that datasets containing information which are not physiologically comparable can always go through the dimension-expanding-RPA-transfomation procedure, but their discriminative information remains uncompatible and can not be used for transfer learning.

Results on P300. We redid the same analysis from above to the case with EEG recordings in the P300 paradigm. Figure 5.5 and Figure 5.6 show that dimensionality transcending works very well when using the full RPA procedure (pipeline **DT**), yielding cross-subject classification scores that are much higher than **calibration**; Table 5.2 shows that **DT** is superior to **calibration** also for different values of n_{covs}. As with the motor imagery data, pipeline **DT-uns** yields inferior results as compared to **calibration** in all instances (see Figure 5.5, Figure 5.6 and Table 5.2), demonstrating that the supervised step from RPA is indeed essential in the dimensionality transcending procedure.

An interesting aspect to note is that although the EEG signals were recorded during experiments where the subjects were oriented to do slightly different cognitive tasks, they both relied on the idea of asking the subject to concentrate on a given target cue and, then, detect a P300 wave in the EEG when the cue flashes. This explains why the dimensionality transcending works in this case, since the discriminative aspects between the classes are the same for both datasets.

5.6 Conclusion

In this chapter, we have considered the problem of working with datasets that describe the same phenomenon but contain samples with different dimensionalities and/or different features. We have been mainly interested in the case where the data points are multivariate time series, so that having different dimensionalities come from, for example, the fact of having recordings with different number and/or placement of sensors. Using the Riemannian geometric framework described in Chapter 2, the time series have been parametrized via SPD matrices and all data manipulations respected the intrinsic geometry of the manifold where they are defined. We presented a mathematical formulation for the problem and proposed a solution consisting of two steps: dimensionality matching followed by statistical distribution matching. The dimensionality matching part uses isometric transformations to map data points defined in SPD manifolds of different dimensionality into a common manifold where they can be naturally compared. The matching of statistical distributions is done via the Riemannian Procrustes analysis presented in Chapter 4. Because our method surpasses the usual limitations due to dimensionality mismatch between data points, we named it *dimensionality transcending* (DT).

We applied DT to two practical situations where multivariate recordings from EEG experiments may have different dimensionalities. The first example considered the case where one (or more) electrode presents a problem and the signal it records has to be rejected. When this happens, one may simply discard the problematic epoch or try to fill the missing values associated to the malfunctioning electrode. Dimensionality transcending is a method for the latter option. We compared the

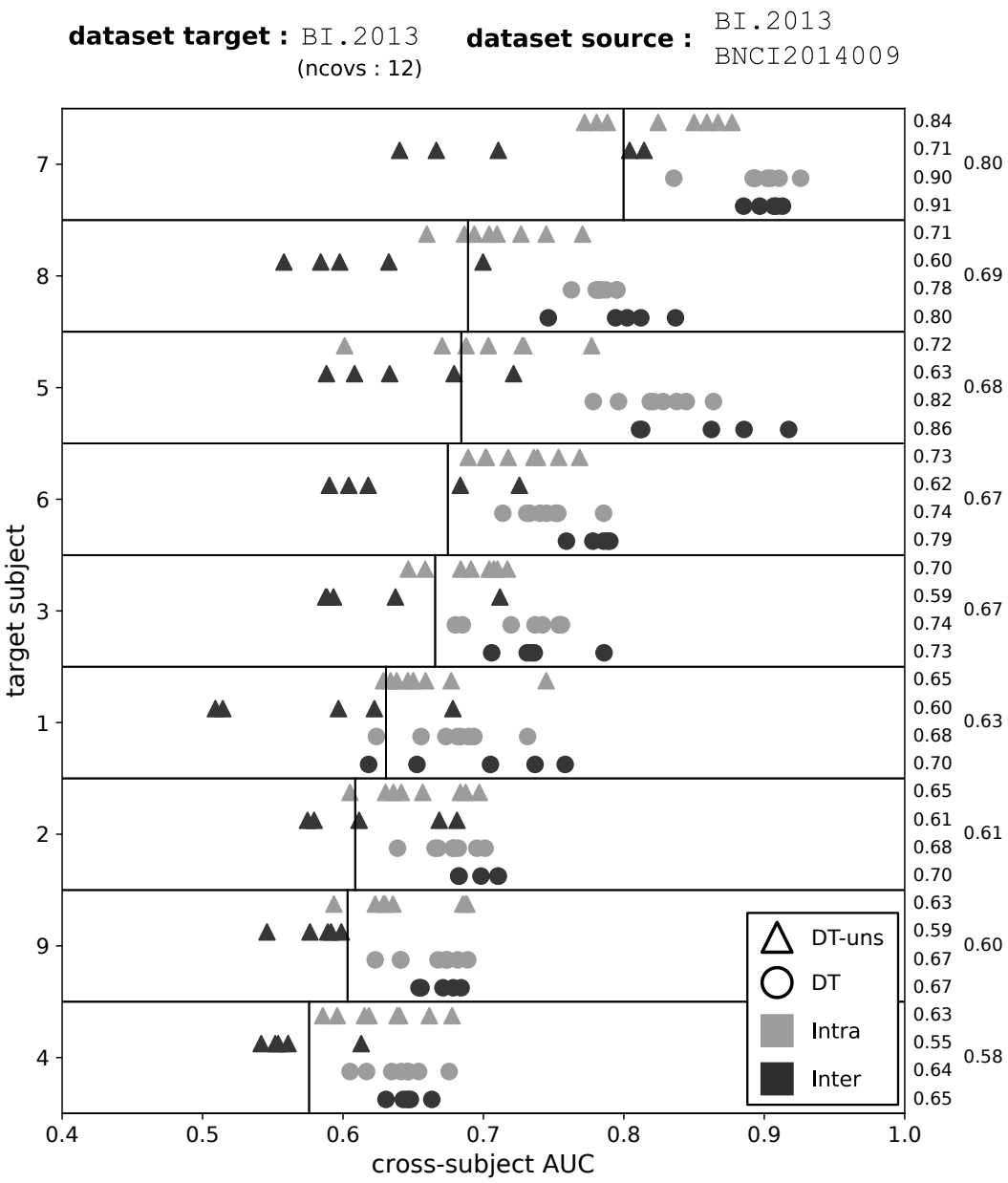

Fig. 5.5: AUC scores for cross-subject classification considering different *target* subjects in `BI.2013`, always with $n_{\text{covs}} = 12$ and $|\mathcal{T}_\ell| = 1 \times n_{\text{covs}} + 5 \times n_{\text{covs}} = 72$ (this comes from the fact that for each Target label in a P300 experiment, there are five other Non-Target labels), and *source* subjects from `BI.2013` ('intra-base' case) and `BNCI2014009` ('inter-base' case). See the caption of Figure 5.3 for more details about the structure of the plot.

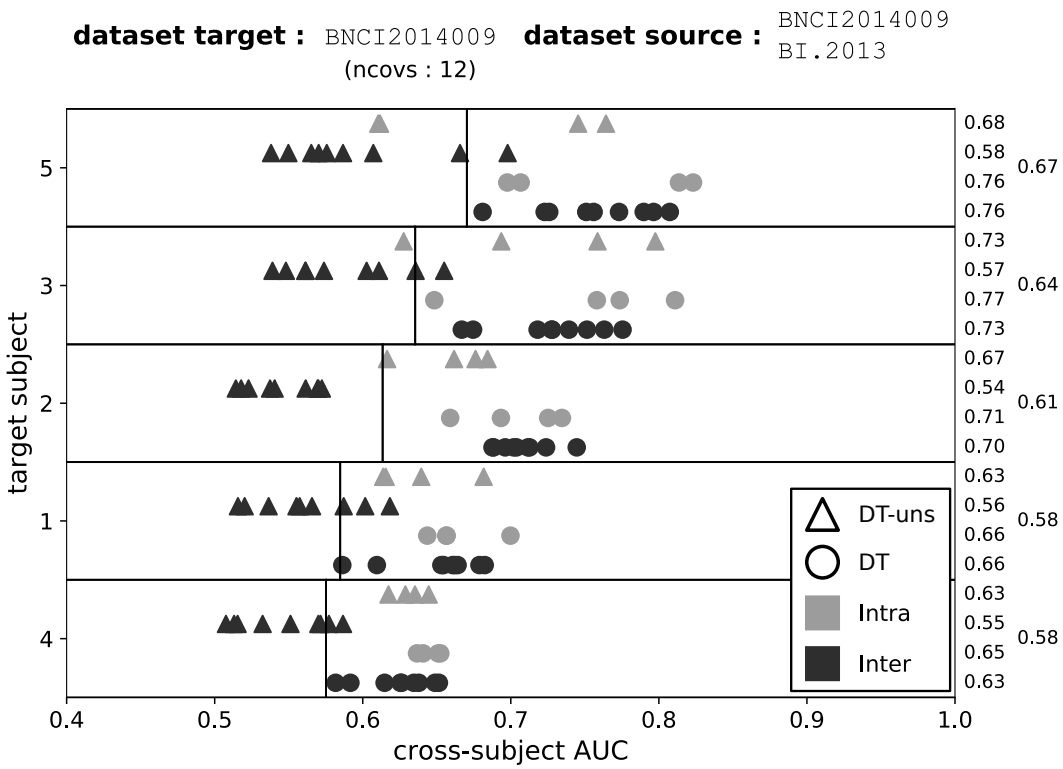

Fig. 5.6: AUC scores for cross-subject classification considering different *target* subjects in `BNCI2014009`, always with $n_{\text{covs}} = 12$ and $|\mathcal{T}_\ell| = 1 \times n_{\text{covs}} + 5 \times n_{\text{covs}} = 72$ (this comes from the fact that for each Target label in a P300 experiment, there are five other Non-Target labels), and *source* subjects from `BNCI2014009` ('intra-base' case) and `BI.2013` ('inter-base' case). See the caption of Figure 5.3 for more details about the structure of the plot.

performance of a classification pipeline using this approach to that of spherical spline interpolation, the standard approach in the EEG literature for replacing missing values from malfunctioning electrodes. The pipeline with DT always performed better.

The second example concerned datasets with EEG recordings from BCI experiments using different electrode configurations. We considered two different BCI paradigms (motor imagery and P300) and investigated whether DT could be used for doing cross-subject classification, that is, train a classifier with data from a *source* subject in one database and apply it to classify unlabeled data points from a *target* subject in a different database. As explained in the text, the supervised distribution matching step in DT (that is, the rotation part in RPA) depends on a few labeled data points from the *target* dataset in order to find an adequate transformation to match the datasets. Our goal was to show that a classifier trained on a set containing data points from a DT-transformed *source* dataset plus a few labeled data points from the *target* dataset had superior performance as compared to that of a pipeline using only the labeled points from the *target* dataset at training time. The latter approach was named 'calibration'. The results with the datasets considered in the text showed that a classification pipeline using DT always attained superior (or at least equivalent) performance as compared to calibration, which is a remarkable result.

A natural question to ask regarding *dimensionality transcending* (DT) is whether it would not be better to simply reduce the dimensionality of the data points into a common space (using, for example, the methods presented in Chapter 3) and then apply a procedure for statistical matching on the new data points (using, for instance, RPA). Although this would avoid increasing the dimensionality of the data points, it would have the risk of losing important discriminative information from the datasets, since the dimensionality is chosen to satisfy datasets whose intrinsic dimensionalities are not necessarily the same. Another relevant question is regarding the 'transition point' for deciding whether two recording sessions should be matched with DT or if we can apply directly RPA. For example, consider two recording sessions where the electrodes are slightly moved. Should these electrodes be considered as different features and, therefore, different dimensions? Should DT be used to match the datasets or can we apply RPA directly? These questions remain unanswered and shall be investigated in the future.

The topics considered in this chapter pave the way to many other interesting questions. For instance, one may consider the case when datasets describe the same phenomena but are recorded using different recording modalities (e.g. fMRI and EEG); the time series still share common features but the matching between them may prove to be more involved as compared to what has been done in this work. Other interesting line of work would be to consider pooling and ensembling strategies which gather the EEG recordings from many different databases recorded with

different electrode configurations and combine them to form a single robust classifier. Finally, one could also envision different transformations in DT's dimensionality matching step. For instance, by relaxing the constraint of isometry one could craft more involved transformations that would play in favor of a better separability between the dataset's classes.

6 Conclusion

The role of invariances in statistical data analysis is analogous to that of a teacher with a student. Although a student may learn the contents of a given course reading a textbook, the insight provided by a teacher often allows the student to understand the concepts faster. In science, invariances convey information about the underlying process that generates a set of observed data points. This information can be used to define statistical models that require fewer training samples to attain good generalization behavior on unseen data points. In this thesis, we have explored invariant properties of multivariate time series. These invariances reflect characteristics of the physical activity represented by the time series and may be used to study different practical problems. To perform such analysis, we have used a geometric framework in which the statistical behavior of the multivariate time series is parametrized by Hermitian positive definite (HPD) matrices. Under this setting, we manipulate time series as points in a metric space and compare them using concepts borrowed from Riemannian geometry (RG).

In Chapter 3, we have considered the invariances of multivariate time series in terms of their dimensionalities. We have used a linear dimensionality reduction technique that extends the classical PCA to reduce the dimensionality of multivariate time series in a geometry-aware fashion. Our results show that statistical classifiers applied to reduced time series attain, in average, the same classification performance as compared to when they are applied to the original time series. This result reflects the existence of some intrinsic information in the physical phenomenon recorded by the time series that is invariant to the number of dimensions used to represent it.

In Chapter 4, we have presented an original transfer learning approach for multivariate time series called *Riemmanian Procrustes analysis* (RPA). Our proposal uses the fact that, although time series recorded from different experimental recordings may have different statistical distributions, if they represent the same physical phenomenon, they are likely to share some latent information that can be exploited. We have illustrated our method on EEG data from BCI experiments carried out with different subjects and have observed promising results in cross-subject classification with pipelines that use RPA to match the statistics of the datasets. Figure 6.1 gives a visual depiction of the three transformations involved in RPA for matching the distribution of the data points from two subjects; the data is the same as the one used for Figure 1.3.

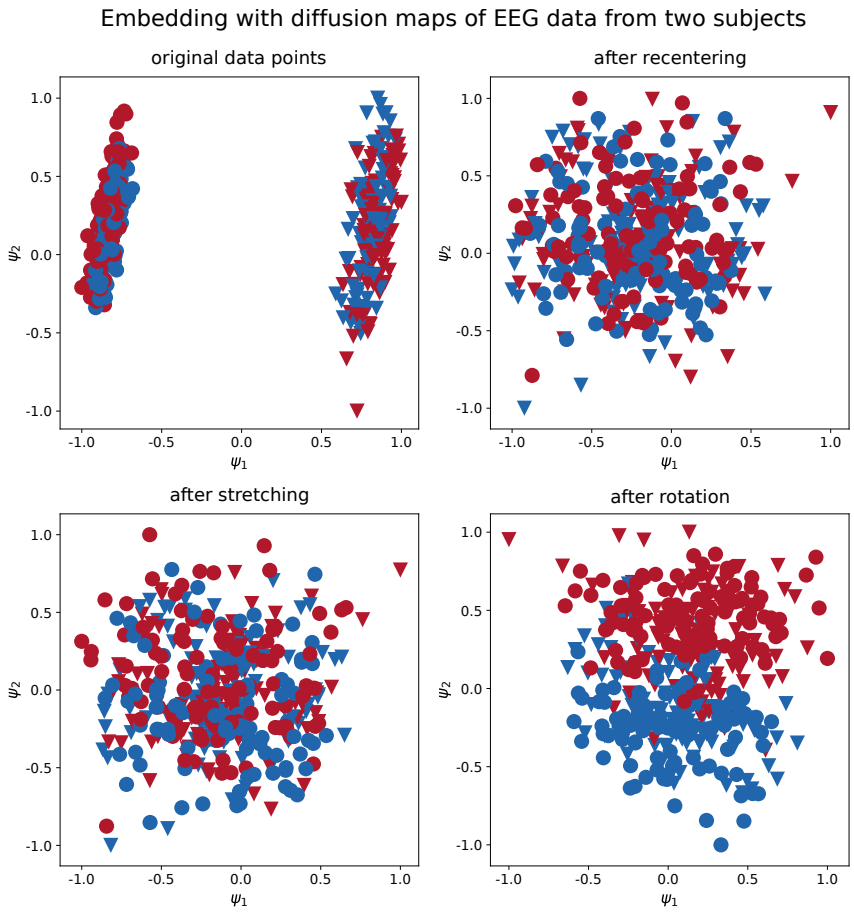

Fig. 6.1: Two-dimensional representation of the embedding obtained via the diffusion maps algorithm applied to the recordings of two subjects in the `Cho2017` database [Cho+17]; the axis ψ_1 and ψ_2 are eigenvectors of the Laplacian matrix estimated from the data points with the diffusion maps algorithm [CL06]. Each subplot is related to one step of the RPA procedure. Each point corresponds to the EEG signal of an experimental trial and the distances between the data points were calculated using the geodesic distance of the SPD manifold. The colors of the scattered points indicate the classes of the EEG epochs, 'left-hand' in red and 'right-hand' in blue, and the different markers indicate whether an epoch is from subject one (circles) or subject two (triangles).

Finally, in Chapter 5, we have enlarged the scope of the invariant property discussed in Chapter 4 and have considered the case when the experimental setup used for recording the samples may also vary. This covers situations where the number and/or the position of sensors in two recording sessions are different, leading to multivariate time series with different dimensionalities and whose dimensions may be associated to activities in different places. Using once again the fact that if the recordings represent the same physical phenomenon, then they must share some latent information, we have proposed an original method called *dimensionality transcending* (DT). DT works by first applying isometric transformations to the elements of each dataset so that they are taken into a space where all data points have the same dimensionality. Then, it uses RPA to match the statistics of these

transformed data points. We have used DT to perform cross-subject classification on BCI data from experiments where the subjects performed the same set of cognitive tasks, but with different electrode setups. Our results show the possibility of sharing information between datasets that were, until then, incompatible.

It is worth mentioning that our original contributions, RPA and DT, are part of a much larger effort in the research community with the goal of designing algorithms capable of extracting information shared between datasets with different dimensionalities, different statistical distributions, etc. The aim of such methods is to go against the current state of affairs of the 'big data era', where large amounts of experimental data are gathered by different laboratories with total disregard to whether they can be jointly used for performing statistical tasks. On a societal point of view, such methods may be seen as 'ecological', since they try to reuse information that already exists and for which some effort has already been put into its generation, the ultimate goal being to avoid the consumption of unnecessary energy for obtaining new data points as well as for storing them.

Future perspectives

In the following paragraphs, we list a few perspectives for the works developed in this thesis. We split the discussion in 'short-term' and 'long-term' perspectives, in the sense that some ideas are rather well posed and easy to tackle, whereas other proposals would need further investigation and reflection.

'Short-term' perspectives. Most machine learning algorithms have a set of parameters that can be adjusted to adapt their behavior to the nuances of the datasets to which they are applied. In its original form, RPA does not have such flexibility, but we believe that adding some hyper-parameters could lead to better results in practice. An example would be to add a variable weight to the contribution of the *source* dataset when training a classifier to label the data points from the *target* dataset. Note that this parameter could also be used in an online implementation, where labeled *target* data points arrive sequentially and the information from the *source* dataset becomes less useful. It would also be advantageous to add weights to each term of the cost function used to obtain the rotation matrix of RPA; such weights could, for instance, reflect the quality of the estimation of the class means on the *target* dataset.

Another interesting line of work would be to investigate how to combine the information from several *source* datasets to classify data points from a *target* dataset using RPA and DT. In the context of cross-subject classification in BCI, this is done using pooling and ensembling strategies, where the contribution of each *source* subject is weighted by an adaptive algorithm such as the one proposed in [Way+16]. By

adding a step for matching the dimensionalities and statistics of the datasets, we can expect that such pooling/ensembling strategies will yield better and more robust classifiers.

Finally, it would be interesting to study the 'transition point' for deciding whether two recording sessions should be matched with DT or if RPA can be directly applied. For example, consider two recording sessions where the electrodes have slightly moved. Should these electrodes be considered as different features and, therefore, different dimensions? Should DT be used to match the datasets or can we apply RPA directly?

'Long-term' perspectives. When working with a d-dimensional multivariate time series $\boldsymbol{x}(t)$, the covariance matrix that describes its statistics have dimension $d \times d$ and its ij-th coordinate describes the statistical correlation between the signals at the i-th and j-th time series of $\boldsymbol{x}(t)$. An interesting generalization would be to consider this matrix as the discretization of a infinite-dimensional covariance operator that describes the correlation between the activity at any two points in space. Such abstraction could be used, for instance, to model how the covariance of the EEG signals recorded over a subject's scalp change when the position of the electrodes change; a relevant first reference for this investigation would be [HQM16].

Another interesting line of work is the study of the dynamical behavior of multivariate time series using the RG framework. For this, we could choose a time scale during which $\boldsymbol{x}(t)$ is approximately stationary and use a sliding window to examine how its statistical behavior evolves in time; an even better method would be to detect automatically the moments when the statistics of the multivariate time series change and must be parametrized by a different covariance matrix. Then, having a sequence of covariance matrices that parametrize the statistics of the multivaraite time series of each window, we may study the characteristics of the trajectory that they engender in the SPD manifold and better understand the phenomena described by $\boldsymbol{x}(t)$. Notice, however, that the estimation of covariance matrices with a sliding window may be challenging, since the limited number of available samples may yield bad estimators. A very relevant line of research would be to study the statistical behavior of these poorly estimated covariance matrices and come up with a corrected version for the expression of the geodesic distance between them in the small-sample regime. The works in [Tio+19] and [CBG16] would be a good place to start.

Finally, we believe that a deeper understanding of the connections between RPA and optimal transport on the SPD manifold could be very profitable. Such investigation could lead, for instance, to an extension of RPA that would transform the data points adaptively using local information instead of doing them in a global fashion.

Bibliography

[Abs+09] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009 (cit. on pp. 19, 22, 53, 92, 93).

[Ama16] Shun ichi Amari. *Information Geometry and Its Applications*. Springer, 2016 (cit. on pp. 2, 20, 24).

[Ari+14] P Aricò, F Aloise, F Schettini, et al. "Influence of P300 latency jitter on event related potential-based brain–computer interface performance". In: *Journal of Neural Engineering* 11.3 (May 2014), p. 035008 (cit. on pp. 40, 138).

[Arj+17] Martín Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein Generative Adversarial Networks". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. 2017, pp. 214–223 (cit. on p. 24).

[Ars+07] Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. "Geometric Means in a Novel Vector Space Structure on Symmetric Positive-Definite Matrices". In: *SIAM Journal on Matrix Analysis and Applications* 29.1 (Jan. 2007), pp. 328–347 (cit. on p. 21).

[AS66] S. M. Ali and S. D. Silvey. "A General Class of Coefficients of Divergence of One Distribution from Another". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 28.1 (1966), pp. 131–142 (cit. on p. 24).

[Ast+08] L. Astolfi, F. Cincotti, D. Mattia, et al. "Tracking the time-varying cortical connectivity patterns by adaptive multivariate estimators". In: *IEEE Transactions on Biomedical Engineering* 55.3 (2008), pp. 902–913 (cit. on p. 13).

[Bar+12] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten. "Multiclass Brain–Computer Interface Classification by Riemannian Geometry". In: *IEEE Transactions on Biomedical Engineering* 59.4 (2012), pp. 920–928 (cit. on pp. 17, 21, 27, 28, 36, 55, 135).

[Bar12] Alexandre Barachant. "Robust control of an actuator by EEG based asynchronous BCI". Theses. Université de Grenoble, Mar. 2012 (cit. on p. 37).

[Bas88] Michèle Basseville. "Detecting changes in signals and systems—A survey". In: *Automatica* 24.3 (May 1988), pp. 309–326 (cit. on p. 13).

[Bas89] Michèle Basseville. "Distance measures for signal processing and pattern recognition". In: *Signal Processing* 18.4 (1989), pp. 349 –369 (cit. on pp. 16, 24).

[BB83] Michèle Basseville and Albert Benveniste. “Sequential segmentation of nonstationary digital signals using spectral analysis”. In: *Information Sciences* 29.1 (1983). Institute of Electrical and Electronics Engineers Workshop Applied Time Series Analysis, pp. 57 –73 (cit. on p. 14).

[BC14] Alexandre Barachant and Marco Congedo. “A plug and play P300 BCI using information geometry”. In: *arXiv* 1104.1 (2014), pp. 1–9. arXiv: 1409.0107v1 (cit. on pp. 33, 102).

[BC19] Rajendra Bhatia and Marco Congedo. “Procrustes problems in Riemannian manifolds of positive definite matrices”. In: *Linear Algebra and its Applications* 563 (2019), pp. 440–445 (cit. on p. 92).

[BD+09] Shai Ben-David, John Blitzer, Koby Crammer, et al. “A theory of learning from different domains”. In: *Machine Learning* 79.1-2 (2009), pp. 151–175 (cit. on pp. 79, 80, 101).

[Bel03] Mikhail Belkin. “Problems of Learning on Manifolds”. AAI3097083. PhD thesis. 2003 (cit. on pp. 60, 67).

[Ben+04] Yoshua Bengio, Olivier Delalleau, Nicolas Le Roux, et al. “Learning Eigenfunctions Links Spectral Embedding and Kernel PCA”. In: *Neural Computation* 16.10 (2004), pp. 2197–2219 (cit. on p. 61).

[Ber+13] T. Berry, J. R. Cressman, Z. Gregurić-Ferenček, and T. Sauer. “Time-Scale Separation from Diffusion-Mapped Delay Coordinates”. In: *SIAM Journal on Applied Dynamical Systems* 12.2 (Jan. 2013), pp. 618–649 (cit. on p. 72).

[Ber+15] Tyrus Berry, Dimitrios Giannakis, and John Harlim. “Nonparametric forecasting of low-dimensional dynamical systems”. In: *Physical Review E* 91.3 (Mar. 2015) (cit. on p. 65).

[Ber+18] Dimitris Bertsimas, Colin Pawlowski, and Ying Daisy Zhuo. “From Predictive Methods to Missing Data Imputation: An Optimization Approach”. In: *Journal of Machine Learning Research* 18.196 (2018), pp. 1–39 (cit. on p. 122).

[Bha+18] Rajendra Bhatia, Tanvi Jain, and Yongdo Lim. “On the Bures–Wasserstein distance between positive definite matrices”. In: *Expositiones Mathematicae* (Jan. 2018) (cit. on p. 21).

[Bha09] Rajendra Bhatia. *Positive definite matrices*. Princeton university press, 2009 (cit. on pp. 3, 15, 19–22).

[Bis07] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2007 (cit. on pp. 25–27, 36, 70, 135).

[BK86] D.S. Broomhead and Gregory P. King. “Extracting qualitative dynamics from experimental data”. In: *Physica D: Nonlinear Phenomena* 20.2-3 (June 1986), pp. 217–236 (cit. on p. 71).

[Bla+08] Benjamin Blankertz, Motoaki Kawanabe, Ryota Tomioka, et al. “Invariant Common Spatial Patterns: Alleviating Nonstationarities in Brain-Computer Interfacing”. In: *Advances in Neural Information Processing Systems 20*. 2008, pp. 113–120 (cit. on p. 84).

[Bor+06] K. M. Borgwardt, A. Gretton, M. J. Rasch, et al. “Integrating structured biological data by Kernel Maximum Mean Discrepancy”. In: *Bioinformatics* 22.14 (July 2006), e49–e57 (cit. on p. 24).

[Buz06] György Buzsáki. *Rhythms of the Brain*. Oxford University Press, Oct. 2006 (cit. on p. 30).

[BV00] J. . Bercher and C. Vignat. “Estimating the entropy of a signal with applications”. In: *IEEE Transactions on Signal Processing* 48.6 (2000), pp. 1687–1694 (cit. on p. 12).

[Cat+18] Gregoire Cattan, Pedro Luiz Coelho Rodrigues, and Marco Congedo. *EEG Alpha Waves Dataset*. Research Report. GIPSA-LAB, Dec. 2018 (cit. on pp. 1, 31, 71).

[CB01] George Casella and Roger Berger. *Statistical Inference*. Duxbury Resource Center, 2001 (cit. on p. 110).

[CB64] Robert M. Chapman and Henry B. Bragdon. “Evoked Responses to Numerical and Non-Numerical Visual Stimuli while Problem Solving”. In: *Nature* 203.4950 (Sept. 1964), pp. 1155–1157 (cit. on p. 30).

[CBG16] Romain Couillet and Florent Benaych-Georges. “Kernel spectral clustering of large dimensional data”. In: *Electronic journal of statistics* 10.1 (2016), pp. 1393–1454 (cit. on p. 152).

[CG11] H Cecotti and A Graser. “Convolutional Neural Networks for P300 Detection with Application to Brain-Computer Interfaces”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.3 (Mar. 2011), pp. 433–445 (cit. on p. 35).

[CG15] John P. Cunningham and Zoubin Ghahramani. “Linear Dimensionality Reduction: Survey, Insights, and Generalizations”. In: *Journal of Machine Learning Research* 16 (2015), pp. 2859–2900 (cit. on p. 47).

[CH91] C. S. Chen and K. . Huo. “Karhunen-Loeve method for data compression and speech synthesis”. In: *IEE Proceedings I - Communications, Speech and Vision* 138.5 (1991), pp. 377–380 (cit. on p. 51).

[Cho+17] Hohyun Cho, Minkyu Ahn, Sangtae Ahn, Moonyoung Kwon, and Sung Chan Jun. “EEG datasets for motor imagery brain–computer interface”. In: *GigaScience* 6.7 (2017), pp. 1–8 (cit. on pp. 4, 102, 134, 150).

[Chu96] Fan Chung. *Spectral Graph Theory*. American Mathematical Society, Dec. 1996 (cit. on p. 64).

[CL06] Ronald R. Coifman and Stéphane Lafon. “Diffusion maps”. In: *Applied and Computational Harmonic Analysis* 21.1 (July 2006), pp. 5–30 (cit. on pp. 4, 60, 61, 63, 98, 150).

[CM14] Romain Couillet and Matthew McKay. “Large dimensional analysis and optimization of robust shrinkage covariance matrix estimators”. In: *Journal of Multivariate Analysis* 131 (2014), pp. 99 –120 (cit. on p. 17).

[Con+11] Marco Congedo, M. Goyat, N. Tarrin, et al. “Brain Invaders: a prototype of an open-source P300-based video game working with the OpenViBE platform”. In: *5th International BCI Conference, Graz, Austria, 280-283* 2011.Bci (2011), pp. 1–6 (cit. on pp. 31, 102).

[Con+17] Marco Congedo, Alexandre Barachant, and Rajendra Bhatia. “Riemannian geometry for EEG-based brain-computer interfaces; a primer and a review”. In: *Brain-Computer Interfaces* (2017), pp. 1–20 (cit. on pp. 3, 36).

[Con13] Marco Congedo. “EEG Source Analysis ”. Habilitation à diriger des recherches. Université de Grenoble, Oct. 2013 (cit. on pp. 16, 17, 22, 44, 102, 136).

[Cor+10] Corinna Cortes, Yishay Mansour, and Mehryar Mohri. “Learning Bounds for Importance Weighting”. In: *Advances in Neural Information Processing Systems 23*. Ed. by J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta. Curran Associates, Inc., 2010, pp. 442–450 (cit. on p. 80).

[Cou+17] Nicolas Courty, Remi Flamary, Devis Tuia, and Alain Rakotomamonjy. “Optimal Transport for Domain Adaptation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.9 (2017), pp. 1853–1865 (cit. on pp. 81, 96, 100, 106).

[Das+98] Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan, and Padhraic Smyth. “Rule Discovery from Time Series”. In: *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*. KDD’98. New York, NY: AAAI Press, 1998, pp. 16–22 (cit. on p. 14).

[Dav+17] Alireza Davoudi, Saeed Shiry Ghidary, and Khadijeh Sadatnejad. “Dimensionality reduction based on distance preservation to local mean for symmetric positive definite matrices and its application in brain–computer interfaces”. In: *Journal of Neural Engineering* 14.3 (Apr. 2017), p. 036019 (cit. on p. 52).

[DG03] D. L. Donoho and C. Grimes. “Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data”. In: *Proceedings of the National Academy of Sciences* 100.10 (Apr. 2003), pp. 5591–5596 (cit. on p. 60).

[Din+15] Shifei Ding, Nan Zhang, Xinzheng Xu, Lili Guo, and Jian Zhang. “Deep Extreme Learning Machine and Its Application in EEG Classification”. In: *Mathematical Problems in Engineering* 2015 (2015), pp. 1–11 (cit. on p. 35).

[Don00] David L. Donoho. “High-dimensional data analysis: The curses and blessings of dimensionality”. In: *AMS Conference on Math Challenges of the 21st Century*. 2000 (cit. on p. 47).

[DPm99] Roberto Domingo Pascual-marqui. “Review of Methods for Solving the EEG Inverse Problem”. In: *Int. J. Biomagn.* 1 (Oct. 1999) (cit. on p. 29).

[Eng+18] Denis A Engemann, Federico Raimondo, Jean-Rémi King, et al. “Robust EEG-based cross-site and cross-protocol classification of states of consciousness”. In: *Brain* 141.11 (Oct. 2018), pp. 3179–3192 (cit. on p. 123).

[EO07] Eugene Edgington and Patrick Onghena. *Randomization Tests*. Chapman and Hall CRC, 2007 (cit. on pp. 105, 112, 140).

[Fal+12] J. Faller, C. Vidaurre, T. Solis-Escalante, C. Neuper, and R. Scherer. “Autocalibration and Recurrent Adaptation: Towards a Plug and Play Online ERD-BCI”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 20.3 (May 2012), pp. 313–319 (cit. on pp. 37, 102, 137).

[Faz+09] Siamac Fazli, Florin Popescu, Márton Danóczy, et al. “Subject-independent mental state classification in single trials”. In: *Neural Networks* 22.9 (2009), pp. 1305 –1312 (cit. on pp. 84, 115).

[FD88] L.A. Farwell and E. Donchin. “Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials”. In: *Electroencephalography and Clinical Neurophysiology* 70.6 (Dec. 1988), pp. 510–523 (cit. on p. 35).

[Fow+04] C. Fowlkes, S. Belongie, Fan Chung, and J. Malik. “Spectral grouping using the nystrom method”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.2 (Feb. 2004), pp. 214–225 (cit. on p. 61).

[FR+12] Reza Fazel-Rezai, Brendan Z. Allison, Christoph Guger, et al. “P300 brain computer interface: current challenges and emerging trends”. In: *Frontiers in Neuroengineering* 5 (2012) (cit. on p. 35).

[Gal13] Robert G. Gallager. *Stochastic Processes*. Cambridge University Press, Dec. 2013 (cit. on p. 62).

[Gay+17] Nathalie T. H. Gayraud, Alain Rakotomamonjy, and Maureen Clerc. “Optimal Transport Applied to Transfer Learning For P300 Detection”. In: *BCI 2017 - 7th Graz Brain-Computer Interface Conference*. Graz, Austria, Sept. 2017, p. 6 (cit. on p. 84).

[GC07] A. Gramfort and M. Clerc. “Low Dimensional Representations of MEG/EEG Data Using Laplacian Eigenmaps”. In: *2007 Joint Meeting of the 6th International Symposium on Noninvasive Functional Source Imaging of the Brain and Heart and the International Conference on Functional Biomedical Imaging*. 2007, pp. 169–172 (cit. on p. 66).

[GD04] John C Gower and Garmt B Dijksterhuis. *Procrustes Problems*. Oxford University Press, 2004 (cit. on pp. 83, 85, 87).

[Gib40] A. Gibert. “Cosmic Rays and Poisson Law”. In: *Nature* 146.3693 (Aug. 1940), pp. 198–198 (cit. on p. 14).

[GM12] D. Giannakis and A. J. Majda. “Nonlinear Laplacian spectral analysis for time series with intermittency and low-frequency variability”. In: *Proceedings of the National Academy of Sciences* 109.7 (Jan. 2012), pp. 2222–2227 (cit. on p. 72).

[GM76] A. Gray and J. Markel. “Distance measures for speech processing”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 24.5 (1976), pp. 380–391 (cit. on p. 16).

[Gol+00] A. L. Goldberger, L. A. N. Amaral, L. Glass, et al. “PhysioBank, PhysioToolkit, and PhysioNet : Components of a New Research Resource for Complex Physiologic Signals”. In: *Circulation* 101.23 (2000), e215–e220 (cit. on pp. 40, 55, 69).

[Goo+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, et al. “Generative Adversarial Nets”. In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger. Curran Associates, Inc., 2014, pp. 2672–2680 (cit. on p. 24).

[Gra13] Alexandre Gramfort. “MEG and EEG data analysis with MNE-Python”. In: *Frontiers in Neuroscience* 7 (2013) (cit. on p. 11).

[Gug+09] Christoph Guger, Shahab Daban, Eric Sellers, et al. “How many people are able to control a P300-based brain–computer interface (BCI)?” In: *Neuroscience Letters* 462.1 (Sept. 2009), pp. 94–98 (cit. on p. 40).

[GV08] A. Goh and R. Vidal. “Clustering and dimensionality reduction on Riemannian manifolds”. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. 2008, pp. 1–7 (cit. on p. 66).

[GW+09] M. Grosse-Wentrup, C. Liefhold, K. Gramann, and M. Buss. “Beamforming in Noninvasive Brain–Computer Interfaces”. In: *IEEE Transactions on Biomedical Engineering* 56.4 (2009), pp. 1209–1219 (cit. on p. 102).

[HA85] Lawrence Hubert and Phipps Arabie. “Comparing partitions”. In: *Journal of Classification* 2.1 (1985), pp. 193–218 (cit. on p. 70).

[Hal+17] David Hallac, Sagar Vare, Stephen Boyd, and Jure Leskovec. “Toeplitz Inverse Covariance-Based Clustering of Multivariate Time Series Data”. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’17. Halifax, NS, Canada: ACM, 2017, pp. 215–223 (cit. on p. 14).

[Ham+04] Jihun Ham, Daniel D. Lee, Sebastian Mika, and Bernhard Schölkopf. “A Kernel View of the Dimensionality Reduction of Manifolds”. In: *Proceedings of the Twenty-first International Conference on Machine Learning*. ICML ’04. Banff, Alberta, Canada: ACM, 2004, pp. 47– (cit. on p. 61).

[Har+14] Mehrtash T. Harandi, Mathieu Salzmann, and Richard Hartley. “From Manifold to Manifold: Geometry-Aware Dimensionality Reduction for SPD Matrices”. In: *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, pp. 17–32 (cit. on p. 52).

[Har+17] Mehrtash Harandi, Mathieu Salzmann, and Richard Hartley. “Joint Dimensionality Reduction and Metric Learning: A Geometric Take”. In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. International Convention Centre, Sydney, Australia: PMLR, 2017, pp. 1404–1413 (cit. on pp. 49, 52, 73).

[Har+18] Mehrtash Harandi, Mathieu Salzmann, and Richard Hartley. “Dimensionality Reduction on SPD Manifolds: The Emergence of Geometry-Aware Methods”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.1 (2018), pp. 48–62 (cit. on pp. 17, 27, 28, 52, 53).

[Has+09] Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The elements of statistical learning: data mining, inference, and prediction, 2nd Edition*. Springer series in statistics. Springer, 2009 (cit. on pp. 38, 50, 66).

[Hay+05] Akira Hayashi, Yuko Mizuhara, and Nobuo Suematsu. “Embedding Time Series Data for Classification”. In: *Machine Learning and Data Mining in Pattern Recognition*. Springer Berlin Heidelberg, 2005, pp. 356–365 (cit. on p. 66).

[Hol79] Sture Holm. “A Simple Sequentially Rejective Multiple Test Procedure”. In: *Scandinavian Journal of Statistics* 6.2 (1979), pp. 65–70 (cit. on pp. 105, 106, 143).

[Hor+16] Inbal Horev, Florian Yger, and Masashi Sugiyama. “Geometry-aware principal component analysis for symmetric positive definite matrices”. In: *Machine Learning* 106.4 (Nov. 2016), pp. 493–522 (cit. on pp. 52, 59).

[HQM16] Minh Ha Quang and Vittorio Murino. “From Covariance Matrices to Covariance Operators: Data Representation from Finite to Infinite-Dimensional Settings”. In: Oct. 2016, pp. 115–143 (cit. on p. 152).

[HSJ10] Sullivan Hidot and Christophe Saint-Jean. “An Expectation-Maximization algorithm for the Wishart mixture model. Application to movement clustering”. In: *Pattern Recognition Letters* 31.14 (July 2010), pp. 2318–2324 (cit. on p. 23).

[Hüb+18] D. Hübner, T. Verhoeven, K. Müller, P. Kindermans, and M. Tangermann. “Unsupervised Learning for Brain-Computer Interfaces Based on Event-Related Potentials: Review and Online Comparison [Research Frontier]”. In: *IEEE Computational Intelligence Magazine* 13.2 (2018), pp. 66–77 (cit. on pp. 84, 85).

[Jay+15] V Jayaram, M Alamgir, Y Altun, B Schölkopf, and M Grosse-Wentrup. “Transfer Learning in Brain-Computer Interfaces”. In: *IEEE Computational Intelligence Magazine* February (2015), pp. 20–31. arXiv: 1512.00296 (cit. on pp. 84, 113, 115).

[JB18] Vinay Jayaram and Alexandre Barachant. “MOABB: trustworthy algorithm benchmarking for BCIs”. In: *Journal of Neural Engineering* 15.6 (Sept. 2018), p. 066011 (cit. on pp. 11, 37, 78, 102, 120, 134).

[JK93] Xing-Qi Jiang and Genshiro Kitagawa. “A time varying coefficient vector AR modeling of nonstationary covariance time series”. In: *Signal Processing* 33.3 (1993), pp. 315 –331 (cit. on p. 13).

[Kal+16] Emmanuel K. Kalunga, Sylvain Chevallier, Quentin Barthélemy, et al. “Online SSVEP-based BCI using Riemannian geometry”. In: *Neurocomputing* 191 (2016), pp. 55–68 (cit. on p. 102).

[Kal+96] J. Kalcher, D. Flotzinger, Ch. Neuper, S. Gölly, and G. Pfurtscheller. “Graz brain-computer interface II: towards communication between humans and computers based on online classification of three different EEG patterns”. In: *Medical & Biological Engineering & Computing* 34.5 (Sept. 1996), pp. 382–388 (cit. on p. 34).

[Kan+91] Eric R. Kandel, James H. Schwartz, and Thomas M. Jessell, eds. *Principles of Neural Science*. Third. New York: Elsevier, 1991 (cit. on p. 28).

[Kar77] H. Karcher. “Riemannian center of mass and mollifier smoothing”. In: *Communications on Pure and Applied Mathematics* 30.5 (Sept. 1977), pp. 509–541 (cit. on p. 22).

[Ken89] David G. Kendall. “A Survey of the Statistical Theory of Shape”. In: *Statistical Science* 4.2 (1989), pp. 87–99 (cit. on pp. 6, 83).

[Keo+93] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. “Segmenting Time Series: A Survey and Novel Approach”. In: *In an Edited Volume, Data mining in Time Series Databases. Published by World Scientific*. Publishing Company, 1993, pp. 1–22 (cit. on p. 14).

[Kin+14] Pieter-Jan Kindermans, Martijn Schreuder, Benjamin Schrauwen, Klaus-Robert Muller, and Michael Tangermann. “True Zero-Training Brain-Computer Interfacing ? An Online Study”. In: *PLOS ONE* 9.7 (July 2014), pp. 1–13 (cit. on p. 84).

[KL51] S. Kullback and R. A. Leibler. “On Information and Sufficiency”. In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86 (cit. on p. 24).

[Kol+90] Zoltan J. Koles, Michael S. Lazar, and Steven Z. Zhou. “Spatial patterns underlying population differences in the background EEG”. In: *Brain Topography* 2.4 (1990), pp. 275–284 (cit. on p. 52).

[Kor+19] Louis Korczowski, Ekaterina Ostaschenko, Anton Andreev, et al. *Brain Invaders 2014a*. 2019 (cit. on p. 40).

[Laf04] S. Lafon. “Diffusion Maps and Geometric Harmonics”. PhD thesis. 2004 (cit. on p. 60).

[Leb05] Guy Lebanon. “Riemannian Geometry and Statistical Machine Learning”. AAI3159986. PhD thesis. Pittsburgh, PA, USA, 2005 (cit. on p. 2).

[Lee+07] R. Leeb, F. Lee, C. Keinrath, et al. “Brain–Computer Communication: Motivation, Aim, and Impact of Exploring a Virtual Apartment”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 15.4 (Dec. 2007), pp. 473–482 (cit. on p. 37).

[LEE+94] J. S. LEE, M. R. GRUNES, and R. KWOK. “Classification of multi-look polarimetric SAR imagery based on complex Wishart distribution”. In: *International Journal of Remote Sensing* 15.11 (July 1994), pp. 2299–2311 (cit. on p. 23).

[Li+09] Li, and H. deBruin. “EEG signal classification based on a Riemannian distance measure”. In: *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*. 2009, pp. 268–273 (cit. on pp. 21, 27).

[Li+12] Y. Li, K.M. Wong, and H. de Bruin. “Electroencephalogram signals classification for sleep-state decision - a Riemannian geometry approach”. In: *IET Signal Processing* 6.4 (2012), pp. 288–299 (cit. on pp. 16, 36).

[Lii10] Innar Liiv. “Seriation and matrix reordering methods: An historical overview”. In: *Statistical Analysis and Data Mining* (2010), n/a–n/a (cit. on p. 103).

[LL06] S. Lafon and A.B. Lee. “Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.9 (2006), pp. 1393–1403 (cit. on pp. 60, 83, 99).

[Lot+07] F Lotte, M Congedo, A Lécuyer, F Lamarche, and B Arnaldi. “A review of classification algorithms for EEG-based brain–computer interfaces”. In: *Journal of Neural Engineering* 4.2 (Jan. 2007), R1–R13 (cit. on p. 35).

[Lot+18] F Lotte, L Bougrain, A Cichocki, et al. “A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update”. In: *Journal of Neural Engineering* 15.3 (2018), p. 031005 (cit. on pp. 2, 3, 34, 35, 84).

[Lot14] Fabien Lotte. “A Tutorial on EEG Signal-processing Techniques for Mental-state Recognition in Brain–Computer Interfaces”. In: *Guide to Brain-Computer Music Interfacing*. Springer London, 2014, pp. 133–161 (cit. on p. 52).

[Lov+14] Miodrag Lovric, Marina Milanovic, and Milan Stamenkovic. “Algorithmic methods for segmentation of time series: An overview”. In: *Journal of Contemporary Economic and Business Issues (JCEBI)* 1 (Jan. 2014), pp. 31–53 (cit. on p. 14).

[Low04] David G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *Int. J. Comput. Vision* 60.2 (Nov. 2004), pp. 91–110 (cit. on p. 3).

[LQ00] Richard M. Leahy and Jinyi Qi. “Statistical approaches in quantitative positron emission tomography”. In: *Statistics and Computing* 10.2 (2000), pp. 147–165 (cit. on p. 14).

[LR02] Roderick J. A. Little and Donald B. Rubin. *Statistical Analysis with Missing Data*. Wiley-Interscience, 2002 (cit. on pp. 122, 131).

[LS97] A. Lopes and F. Sery. “Optimal speckle reduction for the product model in multilook polarimetric SAR imagery and the Wishart distribution”. In: *IEEE Transactions on Geoscience and Remote Sensing* 35.3 (1997), pp. 632–647 (cit. on p. 23).

[Luc14] Steven J. Luck. *An Introduction to the Event-Related Potential Technique*. A Bradford Book, 2014 (cit. on p. 30).

[Lun+03] Stefan Lundbergh, Timo Teräsvirta, and Dick van Dijk. “Time-Varying Smooth Transition Autoregressive Models”. In: *Journal of Business and Economic Statistics* 21.1 (2003), pp. 104–121 (cit. on p. 13).

[Lut07] Helmut Lutkepohl. *New introduction to multiple time series analysis*. New York City, US: Springer, 2007 (cit. on pp. 12, 14).

[Lux07] Ulrike von Luxburg. “A tutorial on spectral clustering”. In: *Statistics and Computing* 17.4 (Aug. 2007), pp. 395–416 (cit. on p. 67).

[LW04] Olivier Ledoit and Michael Wolf. “A well-conditioned estimator for large-dimensional covariance matrices”. In: *Journal of Multivariate Analysis* 88.2 (Feb. 2004), pp. 365–411 (cit. on pp. 55, 135).

[Maa+08] L.J.P. van der Maaten, E. O. Postma, and H. J. van den Herik. *Dimensionality Reduction: A Comparative Review*. 2008 (cit. on p. 60).

[Man+09] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. “Domain Adaptation: Learning Bounds and Algorithms”. In: *Proceedings of The 22nd Annual Conference on Learning Theory (COLT 2009)*. Montreal, Canada, 2009 (cit. on p. 80).

[Mar87] S. Lawrence Marple. *Digital spectral analysis with applications*. New Jersey, US: Prentice Hall, 1987 (cit. on p. 14).

[Mas+18] Estelle M. Massart, Julien M. Hendrickx, and P.-A. Absil. “Matrix geometric means based on shuffled inductive sequences”. In: *Linear Algebra and its Applications* 542 (2018). Proceedings of the 20th ILAS Conference, Leuven, Belgium 2016, pp. 334 –359 (cit. on p. 22).

[May06] Stephen J. Maybank. “Application of the Fisher-Rao Metric to Structure Detection”. In: *Journal of Mathematical Imaging and Vision* 25.1 (June 2006), pp. 49–62 (cit. on pp. 17, 21).

[McC01] R.J. McCann. “Polar factorization of maps on Riemannian manifolds”. In: *Geometric & Functional Analysis GAFA* 11.3 (2001), pp. 589–608 (cit. on p. 96).

[McI+18] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. “UMAP: Uniform Manifold Approximation and Projection”. In: *J. Open Source Software* 3.29 (2018), p. 861 (cit. on p. 60).

[MG15] Ran Manor and Amir B. Geva. “Convolutional Neural Network for Multi-Category Rapid Serial Visual Presentation BCI”. In: *Frontiers in Computational Neuroscience* 9 (Dec. 2015) (cit. on p. 35).

[MH08] Laurens van der Maaten and Geoffrey Hinton. “Visualizing Data using t-SNE”. In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605 (cit. on p. 60).

[Moa05] Maher Moakher. “A Differential Geometric Approach to the Geometric Mean of Symmetric Positive-Definite Matrices”. In: *SIAM J. Matrix Anal. Appl.* 26.3 (Mar. 2005), pp. 735–747 (cit. on pp. 19, 20, 93).

[Mor+96] S. T. Morgan, J. C. Hansen, and S. A. Hillyard. “Selective attention to stimulus location modulates the steady-state visual evoked potential.” In: *Proceedings of the National Academy of Sciences* 93.10 (May 1996), pp. 4770–4774 (cit. on p. 30).

[MS00] P. Marriott and M. Salmon. *Applications of Differential Geometry to Econometrics*. Cambridge University Press, 2000 (cit. on p. 21).

[MS04] Facundo Mémoli and Guillermo Sapiro. “Comparing point clouds”. In: *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing - SGP 04*. ACM Press, 2004 (cit. on p. 122).

[Mém11] Facundo Mémoli. “A spectral notion of Gromov–Wasserstein distance and related methods”. In: *Applied and Computational Harmonic Analysis* 30.3 (May 2011), pp. 363–401 (cit. on p. 61).

[NM93] C. L. Nikias and J. M. Mendel. “Signal processing with higher-order spectra”. In: *IEEE Signal Processing Magazine* 10.3 (1993), pp. 10–37 (cit. on p. 12).

[NS05] Paul L. Nunez and Ramesh Srinivasan. *Electric Fields of the Brain - The Neurophysics of EEG*. Oxford University Press, 2005 (cit. on p. 29).

[OS94] M. Omologo and P. Svaizer. “Acoustic event localization using a crosspower-spectrum phase based technique”. In: *Proceedings of ICASSP ’94. IEEE International Conference on Acoustics, Speech and Signal Processing*. Vol. ii. 1994, II/273–II/276 vol.2 (cit. on p. 12).

[Par+13] Sangwoo Park, Erchin Serpedin, and Khalid Qaraqe. “Gaussian Assumption: The Least Favorable but the Most Useful [Lecture Notes]”. In: *IEEE Signal Processing Magazine* 30.3 (May 2013), pp. 183–186 (cit. on p. 14).

[PC19] Gabriel Peyré and Marco Cuturi. “Computational Optimal Transport”. In: *Foundations and Trends® in Machine Learning* 11.5-6 (2019), pp. 355–206 (cit. on pp. 24, 81–83).

[Pea01] Karl Pearson. “LIII. On lines and planes of closest fit to systems of points in space”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (Nov. 1901), pp. 559–572 (cit. on p. 50).

[Ped+11] F. Pedregosa, G. Varoquaux, A. Gramfort, et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (cit. on p. 11).

[Pen06] Xavier Pennec. “Intrinsic statistics on Riemannian manifolds: Basic tools for geometric measurements”. In: *Journal of Mathematical Imaging and Vision* 25.1 (2006), p. 127 (cit. on pp. 17, 20, 23, 36).

[Per+89] F. Perrin, J. Pernier, O. Bertrand, and J.F. Echallier. “Spherical splines for scalp potential and current density mapping”. In: *Electroencephalography and Clinical Neurophysiology* 72.2 (Feb. 1989), pp. 184–187 (cit. on p. 134).

[PN01] G. Pfurtscheller and C. Neuper. “Motor imagery and direct brain-computer communication”. In: *Proceedings of the IEEE* 89.7 (2001), pp. 1123–1134 (cit. on p. 34).

[PP02] Athanasios Papoulis and S. Unnikrishna Pillai. *Probability, Random Variables, and Stochastic Processes*. Fourth. Boston: McGraw Hill, 2002 (cit. on p. 14).

[Pri83] M. B. Priestley. *Spectral Analysis and Time Series, Two-Volume Set, Volume 1-2: Volumes I and II*. Academic Press, 1983 (cit. on pp. 1, 13, 14, 17).

[PW93] Donald B. Percival and Andrew T. Walden. *Spectral analysis for physical applications*. Cambridge, US: Cambridge University Press, 1993 (cit. on pp. 13, 15).

[PY10] Sinno Jialin Pan and Qiang Yang. “A Survey on Transfer Learning”. In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2010), pp. 1345–1359 (cit. on pp. 77, 80, 86).

[Ram+00] H. Ramoser, J. Muller-Gerking, and G. Pfurtscheller. “Optimal spatial filtering of single trial EEG during imagined hand movement”. In: *IEEE Transactions on Rehabilitation Engineering* 8.4 (Dec. 2000), pp. 441–446 (cit. on pp. 35, 37).

[RB15] Pedro L. C. Rodrigues and Luiz A. Baccala. “A new algorithm for neural connectivity estimation of EEG event related potentials”. In: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, Aug. 2015 (cit. on p. 13).

[Red+17] Ievgen Redko, Amaury Habrard, and Marc Sebban. “Theoretical Analysis of Domain Adaptation with Optimal Transport”. In: *Machine Learning and Knowledge Discovery in Databases*. Springer International Publishing, 2017, pp. 737–753 (cit. on p. 80).

[Rei+08] David Reich, Alkes L Price, and Nick Patterson. “Principal component analysis of genetic data”. In: *Nature Genetics* 40.5 (May 2008), pp. 491–492 (cit. on p. 51).

[Ric+13] Angela Riccio, Luca Simione, Francesca Schettini, et al. “Attention and P300-based BCI performance in people with amyotrophic lateral sclerosis”. In: *Frontiers in Human Neuroscience* 7 (2013) (cit. on p. 40).

[Riv+09] B. Rivet, A. Souloumiac, V. Attina, and G. Gibert. “xDAWN Algorithm to Enhance Evoked Potentials: Application to Brain–Computer Interface”. In: *IEEE Transactions on Biomedical Engineering* 56.8 (Aug. 2009), pp. 2035–2043 (cit. on pp. 35, 38).

[RS00] Sam T. Roweis and Lawrence K. Saul. “Nonlinear dimensionality reduction by locally linear embedding”. In: *SCIENCE* 290 (2000), pp. 2323–2326 (cit. on p. 60).

[Sai+17] Salem Said, Lionel Bombrun, Yannick Berthoumieu, and Jonathan H. Manton. "Riemannian Gaussian Distributions on the Space of Symmetric Positive Definite Matrices". In: *IEEE Transactions on Information Theory* 63.4 (2017), pp. 2153–2170 (cit. on p. 23).

[Sau+91] Tim Sauer, James A. Yorke, and Martin Casdagli. "Embedology". In: *Journal of Statistical Physics* 65.3-4 (Nov. 1991), pp. 579–616 (cit. on p. 71).

[SB11] Petre Stoica and Prabhu Babu. "The Gaussian Data Assumption Leads to the Largest Cramér-Rao Bound [Lecture Notes]". In: *IEEE Signal Processing Magazine* 28.3 (May 2011), pp. 132–133 (cit. on p. 14).

[SC07] Saeid Sanei and J.A. Chambers. *EEG Signal Processing*. John Wiley and Sons Ltd, 2007 (cit. on pp. 12, 29, 42).

[Sch+04] G. Schalk, D.J. McFarland, T. Hinterberger, N. Birbaumer, and J.R. Wolpaw. "BCI2000: A General-Purpose Brain-Computer Interface (BCI) System". In: *IEEE Transactions on Biomedical Engineering* 51.6 (2004), pp. 1034–1043 (cit. on p. 102).

[Sch+98] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. "Nonlinear Component Analysis as a Kernel Eigenvalue Problem". In: *Neural Computation* 10.5 (July 1998), pp. 1299–1319 (cit. on p. 61).

[Shi00] Hidetoshi Shimodaira. "Improving predictive inference under covariate shift by weighting the log-likelihood function". In: *Journal of Statistical Planning and Inference* 90.2 (2000), pp. 227–244 (cit. on p. 83).

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. New York, NY, USA: Cambridge University Press, 2014 (cit. on pp. 26, 27).

[Ste+16] David Steyrl, Reinhold Scherer, Josef Faller, and Gernot R. Müller-Putz. "Random forests in non-invasive sensorimotor rhythm brain-computer interfaces: a practical and convenient non-linear classifier". In: *Biomedical Engineering / Biomedizinische Technik* 61.1 (Feb. 2016), pp. 77–86 (cit. on pp. 37, 102).

[Sug+07] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. "Covariate Shift Adaptation by Importance Weighted Cross Validation". In: *J. Mach. Learn. Res.* 8 (Dec. 2007), pp. 985–1005 (cit. on p. 83).

[Tal+13] Ronen Talmon, Israel Cohen, Sharon Gannot, and Ronald R. Coifman. "Diffusion Maps for Signal Processing: A Deeper Look at Manifold-Learning Techniques Based on Kernels and Graphs". In: *IEEE Signal Processing Magazine* 30.4 (July 2013), pp. 75–86 (cit. on p. 66).

[Tan+12] Michael Tangermann, Klaus-Robert MŸller, Ad Aertsen, et al. "Review of the BCI Competition IV". In: *Frontiers in Neuroscience* 6 (2012) (cit. on p. 102).

[Ter+01] Mario Giovanni Terzano, Liborio Parrino, Adriano Sherieri, et al. "Atlas, rules, and recording techniques for the scoring of cyclic alternating pattern (CAP) in human sleep". In: *Sleep Medicine* 2.6 (2001), pp. 537–553 (cit. on pp. 40, 69).

[Tio+19] Malik Tiomoko, Romain Couillet, Florent Bouchard, and Guillaume Ginolhac. "Random Matrix Improved Covariance Estimation for a Large Class of Metrics". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. Long Beach, California, USA: PMLR, 2019, pp. 6254–6263 (cit. on pp. 17, 152).

[Tow+16] James Townsend, Niklas Koep, and Sebastian Weichwald. "Pymanopt: A Python Toolbox for Optimization on Manifolds using Automatic Differentiation". In: *Journal of Machine Learning Research* 17.137 (2016), pp. 1–5 (cit. on pp. 54, 93).

[TP91] Matthew Turk and Alex Pentland. "Eigenfaces for Recognition". In: *J. Cognitive Neuroscience* 3.1 (Jan. 1991), pp. 71–86 (cit. on p. 51).

[Tuc95] Marco P. Tucci. "Time-varying parameters: a critical introduction". In: *Structural Change and Economic Dynamics* 6.2 (1995), pp. 237 –260 (cit. on p. 13).

[Tuz+06] Oncel Tuzel, Fatih Porikli, and Peter Meer. "Region covariance: A fast descriptor for detection and classification". In: *European conference on computer vision*. Springer. 2006, pp. 589–600 (cit. on pp. 17, 21, 27, 52).

[Vai+18] Erwan Vaineau, Alexandre Barachant, Anton Andreev, et al. *Brain Invaders Adaptive versus Non-Adaptive P300 Brain-Computer Interface dataset*. en. 2018 (cit. on pp. 40, 138).

[Vil09] Cédric Villani. *Optimal Transport*. Springer Berlin Heidelberg, 2009 (cit. on p. 81).

[VV+19] Gijsbrecht Franciscus Petrus Van Veen, Alexandre Barachant, Anton Andreev, et al. *Building Brain Invaders: EEG data of an experimental validation*. 2019 (cit. on p. 40).

[Way+16] Nicholas R. Waytowich, Vernon J. Lawhern, Addison W. Bohannon, Kenneth R. Ball, and Brent J. Lance. "Spectral Transfer Learning Using Information Geometry for a User-Independent Brain-Computer Interface". In: *Frontiers in Neuroscience* 10 (2016) (cit. on pp. 84, 113, 115, 151).

[WH00] A. Weingessel and K. Hornik. "Local PCA algorithms". In: *IEEE Transactions on Neural Networks* 11.6 (2000), pp. 1242–1250 (cit. on p. 52).

[Wie+18] Piotr Wierzgała, Dariusz Zapała, Grzegorz M. Wojcik, and Jolanta Masiak. "Most Popular Signal Processing Methods in Motor-Imagery BCI: A Review and Meta-Analysis". In: *Frontiers in Neuroinformatics* 12 (Nov. 2018) (cit. on p. 34).

[Wis28] John Wishart. "The generalised product moment distribution in samples from a normal multivariate population". In: *Biometrika* 20A.1-2 (1928), pp. 32–52 (cit. on p. 23).

[WM08] Chang Wang and Sridhar Mahadevan. "Manifold alignment using Procrustes analysis". In: *Proceedings of the 25th international conference on Machine learning - ICML 2008*. ACM Press, 2008 (cit. on p. 83).

[Wol+91] Jonathan R. Wolpaw, Dennis J. McFarland, Gregory W. Neat, and Catherine A. Forneris. "An EEG-based brain-computer interface for cursor control". In: *Electroencephalography and Clinical Neurophysiology* 78.3 (Mar. 1991), pp. 252–259 (cit. on p. 34).

[WV05] Zhizhou Wang and B.C. Vemuri. “DTI segmentation using an information theoretic tensor dissimilarity measure”. In: *IEEE Transactions on Medical Imaging* 24.10 (Oct. 2005), pp. 1267–1277 (cit. on p. 21).

[Yai+19] Or Yair, Felix Dietrich, Ronen Talmon, and Ioannis G. Kevrekidis. “Optimal Transport on the Manifold of SPD Matrices for Domain Adaptation”. In: *arXiv e-prints*, arXiv:1906.00616 (June 2019), arXiv:1906.00616. arXiv: 1906.00616 [cs.HC] (cit. on pp. 96, 100).

[Yge+17] Florian Yger, Maxime Berar, and Fabien Lotte. “Riemannian Approaches in Brain-Computer Interfaces: A Review”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 25.10 (2017), pp. 1753–1762 (cit. on pp. 3, 36).

[Yi+14] Weibo Yi, Shuang Qiu, Kun Wang, et al. “Evaluation of EEG Oscillatory Patterns and Cognitive Process during Simple and Compound Limb Motor Imagery”. In: *PLoS ONE* 9.12 (Dec. 2014). Ed. by Natasha M. Maurits, e114853 (cit. on p. 37).

[Zan+17] Paolo Zanini, Marco Congedo, Christian Jutten, Salem Said, and Yannick Berthoumieu. “Transfer Learning: a Riemannian geometry framework with applications to Brain-Computer Interfaces”. In: *IEEE Transactions on Biomedical Engineering* (2017), pp. 1–1 (cit. on pp. 83–85, 88, 100, 106, 114, 115).

[Zay11] D. V. Zaykin. “Optimally weighted Z-test is a powerful method for combining probabilities in meta-analysis”. In: *Journal of Evolutionary Biology* 24.8 (2011), pp. 1836–1841 (cit. on pp. 105, 140).

[Zho+16] Bangyan Zhou, Xiaopei Wu, Zhao Lv, Lei Zhang, and Xiaojin Guo. “A Fully Automated Trial Selection Method for Optimization of Motor Imagery Based Brain-Computer Interface”. In: *PLOS ONE* 11.9 (Sept. 2016). Ed. by Bin He, e0162657 (cit. on pp. 37, 137).

[Zhu+10] Danhua Zhu, Jordi Bieger, Gary Garcia Molina, and Ronald M. Aarts. “A Survey of Stimulation Methods Used in SSVEP-Based BCIs”. In: *Computational Intelligence and Neuroscience* 2010 (2010), pp. 1–12 (cit. on p. 34).

[ZZ04] Zhenyue Zhang and Hongyuan Zha. “Principal Manifolds and Nonlinear Dimensionality Reduction via Tangent Space Alignment”. In: *SIAM Journal on Scientific Computing* 26.1 (Jan. 2004), pp. 313–338 (cit. on p. 60).

[Arn+13] M. Arnaudon, F. Barbaresco, and L. Yang. “Riemannian Medians and Means With Applications to Radar Signal Processing”. In: *IEEE Journal of Selected Topics in Signal Processing* 7.4 (2013), pp. 595–604 (cit. on p. 22).

[Bar08] F. Barbaresco. “Innovative tools for radar signal processing Based on Cartan?s geometry of SPD matrices Information Geometry”. In: *2008 IEEE Radar Conference*. 2008, pp. 1–6 (cit. on pp. 16, 21, 22, 27).

[Bro+17] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. “Geometric Deep Learning: Going beyond Euclidean data”. In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 18–42 (cit. on p. 2).

[Cha+18] S. Chambon, M. N. Galtier, and A. Gramfort. “Domain adaptation with optimal transport improves EEG sleep stage classifiers”. In: *2018 International Workshop on Pattern Recognition in Neuroimaging (PRNI)*. 2018, pp. 1–4 (cit. on p. 84).

[Che+10] Y. Chen, A. Wiesel, Y. C. Eldar, and A. O. Hero. “Shrinkage Algorithms for MMSE Covariance Estimation”. In: *IEEE Transactions on Signal Processing* 58.10 (2010), pp. 5016–5029 (cit. on p. 17).

[Che+19] Y. Chen, T. T. Georgiou, and A. Tannenbaum. “Optimal Transport for Gaussian Mixture Models”. In: *IEEE Access* 7 (2019), pp. 6269–6278 (cit. on p. 96).

[Coi+08] R. R. Coifman, Y. Shkolnisky, F. J. Sigworth, and A. Singer. “Graph Laplacian Tomography From Unknown Random Projections”. In: *IEEE Transactions on Image Processing* 17.10 (2008), pp. 1891–1899 (cit. on p. 65).

[Con+13] Marco Congedo, Alexandre Barachant, and Anton Andreev. “A New Generation of Brain-Computer Interface Based on Riemannian Geometry”. In: *arXiv e-prints*, arXiv:1310.8115 (Oct. 2013), arXiv:1310.8115. arXiv: 1310.8115 [cs.HC] (cit. on p. 84).

[Con+17] M. Congedo, A. Barachant, and E. K. Koopaei. “Fixed Point Algorithms for Estimating Power Means of Positive Definite Matrices”. In: *IEEE Transactions on Signal Processing* 65.9 (2017), pp. 2211–2220 (cit. on p. 22).

[Con+19] M. Congedo, P. L. C. Rodrigues, and C. Jutten. “The Riemannian Minimum Distance to Means Field Classifier”. In: *Graz BCI Conference 2019*. 2019 (cit. on p. 22).

[Fer+13] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. “Unsupervised Visual Domain Adaptation Using Subspace Alignment”. In: *2013 IEEE International Conference on Computer Vision*. 2013, pp. 2960–2967 (cit. on p. 81).

[Fer+15] A. Feragen, F. Lauze, and S. Hauberg. “Geodesic exponential kernels: When curvature and linearity conflict”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3032–3042 (cit. on p. 66).

[Lim+19] L. Lim, R. Sepulchre, and K. Ye. “Geometric Distance Between Positive Definite Matrices of Different Dimensions”. In: *IEEE Transactions on Information Theory* 65.9 (2019), pp. 5401–5405 (cit. on p. 123).

[Mal+18] M. Malfante, M. Dalla Mura, J. Metaxian, et al. “Machine Learning for Volcano-Seismic Signals: Challenges and Perspectives”. In: *IEEE Signal Processing Magazine* 35.2 (2018), pp. 20–30 (cit. on p. 13).

[Mam+19] G. Maman, O. Yair, D. Eytan, and R. Talmon. “Domain Adaptation Using Riemannian Geometry of Spd Matrices”. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, pp. 4464–4468 (cit. on p. 84).

[Pan+11] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. “Domain Adaptation via Transfer Component Analysis”. In: *IEEE Transactions on Neural Networks* 22.2 (2011), pp. 199–210 (cit. on p. 121).

[Rod+18] P. L. C. Rodrigues, M. Congedo, and C. Jutten. “Multivariate Time-Series Analysis Via Manifold Learning”. In: *2018 IEEE Statistical Signal Processing Workshop (SSP)*. 2018, pp. 573–577 (cit. on pp. 66, 83, 84, 123).

[Tuz+08] O. Tuzel, F. Porikli, and P. Meer. “Pedestrian Detection via Classification on Riemannian Manifolds”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.10 (2008), pp. 1713–1727 (cit. on pp. 17, 28, 52).

[Vin+06] E. Vincent, R. Gribonval, and C. Fevotte. “Performance measurement in blind audio source separation”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 14.4 (2006), pp. 1462–1469 (cit. on pp. 16, 44).

[Yai+19] O. Yair, M. Ben-Chen, and R. Talmon. “Parallel Transport on the Cone Manifold of SPD Matrices for Domain Adaptation”. In: *IEEE Transactions on Signal Processing* 67.7 (2019), pp. 1797–1811 (cit. on pp. 22, 84, 85, 100, 106, 114).

[Yge+15] F. Yger, F. Lotte, and M. Sugiyama. “Averaging covariance matrices for EEG signal classification based on the CSP: An empirical study”. In: *2015 23rd European Signal Processing Conference (EUSIPCO)*. 2015, pp. 2721–2725 (cit. on p. 52).