



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

UNIVERSITÀ DEGLI STUDI DI PADOVA
Department of Statistical Sciences

PhD course in Statistics
Ciclo xxxiv
Coordinator: Prof Nicola Sartori

DIFFERENTIAL GEOMETRY OF SYMMETRIC
AND POSITIVE DEFINITE MATRICES FOR
STATISTICAL APPLICATIONS

Supervisor: Prof. Antonio Canale
Co-Supervisor: Prof. Mauro Bernardi
Co-Supervisor: Prof. Livio Finos

Candidate:
Jacopo Schiavon

Jacopo Schiavon: *Differential Geometry of symmetric and positive definite matrices for statistical applications*. © Padova, 2022

Supervisors: Prof. Antonio Canale, Prof. Mauro Bernardi, Prof. Livio Finos

Template by Jacopo Schiavon: github.com/jschiavon/clean-thesis.

*A Eleonora,
compagna di avventure
compagna di vita*

ACKNOWLEDGEMENTS

As usual, getting to the end of such a journey and looking back, it is really difficult to express all the feelings and emotions that have filled it. In particular, the last couple of years have been very challenging, for me and for almost everyone else, and many people have helped me to get through it.

To start, I have to thank Antonio, Mauro, and Livio, my supervisors: without them this work would have never been done. Their patience when I felt overwhelmed and their stern motivation when I needed a push are something that I am very grateful for.

I also want to thank the reviewers, Prof. Solari and Dr. Pigoli, for their insightful observations. A hearth-felt thank goes also to Prof. Sartori and the rest of the doctoral committee for listening to our problems.

I am really grateful for my colleagues Anam, Dung, Emanuele, Fabio, Jacopo, Laura, and Mattia, that have made sharing the Uggè room a real pleasure despite the uncomfortable chairs. In particular, I will always remember the bond I built with the red devils Anna and Silvia.

But I would never have pulled this off without my 3 families: the first one was born in *Pollaio*, and that chalky and noisy room with its weird fauna will always have a place in my hearth. The second has been built over dices and proletarian sodas: we might be far, but I know that each of you will always be just a Discord call away. In particular, I am pretty sure without the support of my *Mistici*, nothing would have worked the way it did, and I would be a much worse person now.

Finally, my real family: Lorenzo and Linda, Mom and Dad. I know that you don't get to choose the family in which you are born, but I am pretty sure I would have chosen mine anyway!

And, of course, the deepest of thanks goes to Eleonora, my travel companion, always there to challenge and comfort me. I wouldn't be here without you.

ABSTRACT

Differential geometry is the set of tools that allows to perform the usual mathematical tasks of algebra and calculus on spaces that do not behave like Euclidean vector spaces, for instance points on a curved surface. This field of mathematics is becoming more and more relevant in multiple fields, statistics and machine learning among those, due to the enormous availability of data belonging to increasingly complex domains. An example among many of such complex domains is the set of Symmetric and Positive Definite matrices, *i.e.* the set of covariance matrices, that appears frequently in medical imaging but is also used often as parameter space in statistical modeling scenarios.

The aim of this thesis is to collect and organize the scattered knowledge on the Riemannian geometry of the symmetric and positive definite matrices, and to build practical techniques using the tools of differential geometry that can be readily applied within a pipeline of statistical analysis. This has been achieved with two different methods: the first is a quasi-Newton algorithm for Riemannian optimization that can be plugged in any situation in which maximization of a function of symmetric and positive definite matrices is required, such as those that arise in the context of likelihood inference and variational approximation. The second is a Riemannian registration algorithm to perform a pre-processing of symmetric and positive definite data such as those arising from medical imaging or brain computer interface. This algorithm, among other properties, provides a theoretical framework to focus the analysis on the eigenvalues of the analyzed matrices, allowing the employment of Euclidean methods for statistical inference

also in a Riemannian context.

SOMMARIO

La geometria differenziale è un insieme di strumenti che permette di compiere le tipiche operazioni di algebra e calcolo anche in spazi che non seguono le normali regole Euclidee degli spazi vettoriali, ad esempio come i punti di una superficie curva. Questo campo della matematica sta assumendo sempre maggiore rilevanza in vari ambiti, fra cui statistica e machine learning, a causa dell'enorme disponibilità di dati che appartengono a domini sempre più complessi. Un esempio di dominio di questo tipo è l'insieme delle matrici simmetriche e definite positive, ovvero le matrici di covarianza, che compaiono frequentemente nella diagnostica medica per immagini e sono spesso usate come spazio parametrico nei modelli statistici.

Lo scopo di questa tesi è quello di raccogliere e organizzare la conoscenza sparsa sulla geometria Riemanniana delle matrici simmetriche e definite positive e di costruire delle tecniche pratiche, usando gli strumenti della geometria differenziale, che possano essere applicate direttamente in contesti di analisi statistica. Questo obiettivo è stato perseguito attraverso lo sviluppo di due metodi: il primo è un algoritmo di tipo quasi-Newton per l'ottimizzazione Riemanniana che può essere utilizzato in qualsiasi situazione in cui sia necessaria la massimizzazione di funzioni di matrici simmetriche e definite positive, come quelle che emergono nel contesto della inferenza di verosimiglianza e nella approssimazione variazionale. Il secondo è un algoritmo di registrazione Riemanniana per eseguire il pre-processamento di dati simmetrici e definiti positivi come quelli che si ottengono nella diagnostica medica per immagini o nelle interfacce cervello-computer. Questo algoritmo, fra le altre sue proprietà, fornisce una struttura teorica che con-

sente di concentrare l'analisi sugli autovalori delle matrici analizzate, permettendo l'utilizzo di metodi Euclidei per l'inferenza statistica anche in un contesto Riemanniano.

CONTENTS

LIST OF FIGURES	xiii	
LIST OF TABLES	xv	
INTRODUCTION	1	
Overview	1	
Main contributions of the thesis	3	
1 RIEMANNIAN GEOMETRY OF THE SPD MANIFOLD	7	
I Manifold notation and Riemannian geometry	7	
II The SPD manifold and its properties	13	
2 RIEMANNIAN OPTIMIZATION	19	
I Optimization algorithms	20	
II Simulation study	25	
III Skew-Normal variational approximation	32	
3 RIEMANNIAN REGISTRATION OF MATRIX DATA	39	
I Riemannian registration algorithm	42	
II Simulation study	49	
III Applications to medical imaging data	52	
4 DISCUSSION	61	
A QUASI-NEWTON OPTIMIZATION METHODS: IMPLEMENTATION DETAILS	65	
I Strong Wolfe line-search procedure	65	
II Limited memory BFGS algorithm	67	
B TECHNICAL DETAILS ON SNVA	71	
I Some properties of the skew-normal distribution	71	

II	Computations for the normal sample	73
	ACRONYMS	75
	BIBLIOGRAPHY	77
	CURRICULUM VITAE	85

LIST OF FIGURES

1.1	Charts on a smooth manifold	8
1.2	Exponential and logarithm map on a manifold	9
1.3	Parallel transport	12
1.4	The cone of Symmetric and Positive-Definite (SPD) matrices	14
2.1	Simulation of the multivariate normal MLE	28
2.2	Simulation of the multivariate skew-normal MLE	31
3.1	Simulation results for registration method: deviations	51
3.2	Simulation results for registration method: computational time	53
3.3	Original and registered empirical correlations for EEG subjects	55
3.4	Karcher center of mass and reference matrix for EEG subjects	55
3.5	Eigenvalue distribution for fMRI data	57
3.6	Eigenvalues significance for fMRI data	57
3.7	MDS projection of the response group for NKI1 data before and after the registration procedure	59

LIST OF TABLES

1.1	Operators in linear vector spaces and manifold	10
2.1	Simulation of the multivariate normal MLE	27
2.2	Simulation of the multivariate skew-normal MLE	30

INTRODUCTION

OVERVIEW

Differential geometry was born as a mathematical description of curved spaces and smooth shapes, and it can be dated back to the ancient Greeks and their study of the geometry of the Earth. Modern differential geometry, introduced mainly after the surge of the new challenges brought by Einstein's General Relativity and by Quantum Field Theory, came around in the first half of the XX century and focused on a more general discussion of geometric structures on differentiable manifolds, usually described as smooth, curved subspaces of the Euclidean space. In modern mathematics, differential geometry lives at the intersection between algebra, calculus, group theory and geometry, merging elements from each of these fields to build a more general framework.

The main idea of differential geometry (and Riemannian geometry specifically, which is the main topic of this work) is that a smooth but not flat surface can still locally resemble an Euclidean space, and thus the standard tools of linear algebra and calculus can still be employed *locally* to perform the necessary computations. The main point, thus, becomes how to extend this local behavior to the global scale, and this is made through the introduction of concepts such as geodesics, affine connections and curvature.

Since the early '60s, statisticians have started to look at differential geometry with interest, spearheaded by the work of Rao (RAO 1945) and others that has culminated in the creation of the field called *information geometry* and mostly developed by Amari (in the monographies AMARI and NAGAOKA

2000; AMARI 2016). After this first intuition, many others have started applying concepts from differential geometry to applied statistical or machine learning problems, with examples ranging from techniques for dimensionality reduction to inference methods and deep learning architectures: a short overview of this broad interest is given in the following paragraphs, that details some examples of interests.

In general, differential geometric techniques and tools can be applied to the statistical problem from different point of view, identified by what space between those of interest exhibit a non-Euclidean behavior. For instance, information geometry deals with the consequences of considering the space of parametric distributions as a smooth manifold, and thus using results from differential geometry to rethink the process of statistical inference (see NIELSEN 2020, for a modern review). Similar in principle is the recently born field of Geometric Deep Learning (BRONSTEIN et al. 2021), where the similarity between deep learning architectures is investigated with techniques from the differential geometry of graphs. Similar principles are applied also in recent works from MALAGÒ and PISTONE 2015 to perform optimization over the exponential family.

Instead, if the data are defined on a non-Euclidean manifold, techniques such as manifold learning (TENENBAUM et al. 2000; LIN and ZHA 2008) or manifold classification are usually employed (BHATTACHARYA and DUNSON 2010; BARACHANT et al. 2012; LI and DUNSON 2020). In a similar fashion, manifold-based methods are employed when the parameter space is non Euclidean, such as the Markov Chain Monte Carlo (MCMC) methods developed in HOLBROOK et al. 2018 and similar works.

Such particular techniques are a particular examples of a larger scenario that concerns the development of statistical methods on manifold-valued data. Indeed, much work have been done in recent years on this topic, both from a non-parametric (DRYDEN et al. 2009; PATRANGENARU and ELLINGSON 2016) and from a more distributional (SAID et al. 2017) point of view.

A last example of the use of differential geometry for statistical applications is again related to the class of MCMC methods, with the development

of the Riemannian manifold Hamiltonian Monte Carlo algorithm by GIROLAMI and CALDERHEAD 2011; BETANCOURT et al. 2017, which exploits a metric on the parameter space induced by the *Fisher information metric*, as defined in the setting of information geometry, to improve the sampling efficiency of the usual Hamiltonian sampler.

In this work, we will deal with a single example of manifold, *i.e.* the manifold of Symmetric and Positive-Definite (SPD) matrices, widely used in statistics as it is the space of full-rank covariance matrices. Indeed, the manifold of SPD matrices is relevant both as a parameter space (for instance in HOLBROOK et al. 2018; LAN et al. 2020) and as a sample space (mainly in medical imaging, such as in PENNEC et al. 2006; BARACHANT et al. 2012).

After a technical introduction to review and organize the formal notation of differential geometry and the geometry of SPD matrices in Chapter 1, we will explore both scenarios respectively in Chapters 2 and 3.

MAIN CONTRIBUTIONS OF THE THESIS

The two main contributions of this thesis are related to two of the main uses of differential geometry for statistical applications. In Chapter 2, we focus on the problem of optimization over the space of SPD manifold, showing an example of use in the context of Bayesian Variational Approximation (VA). Chapter 3 instead is devoted to the proposal of a pre-processing method for SPD data, similar in spirit to Procrustes analysis, that focus on the underlying eigenvalue structure of a matrix. This method, named Riemannian Registration (RR) algorithm for SPD matrices, is then applied to examples with Electroencephalography (EEG) and functional Magnetic Resonance Imaging (fMRI) data.

Riemannian optimization

Section 2.1 is devoted to the development of a novel algorithm to perform optimization over the space of SPD matrices. In short, the focus of this chapter is to obtain an elegant and efficient solution to the problem

$$\min_{\Sigma \in \mathcal{S}^+} f(\Sigma),$$

where the symbol \mathcal{S}^+ denotes the manifold of SPD matrices.

To do this, instead of performing a constrained optimization or trying to parameterize the function f in order to remove the constraints, we exploit the notions of geodesics and parallel transport to construct an implementation of standard Newton and quasi-Newton optimization algorithm that search the optimal matrix directly over the manifold \mathcal{S}^+ . This allows to radically improve the computational performances with respect to a Cholesky-based Euclidean method as the problem's dimension increases. Indeed, as the efficiency of traversing the manifold when using a geodesic path is approximately constant, the use of a Cholesky decomposition produces a $\mathcal{O}(n^2)$ Euclidean optimization problem, which quickly incurs in the curse of dimensionality. This gain in performances is shown through simulations discussed in Section 2.11.

Finally, an example of use of this method in the context of Bayesian VA is shown in Section 2.13. Specifically, we discuss the Skew-Normal Variational Approximation (SNVA) method presented in ORMEROD 2011, where optimization over the covariance parameter of the approximating distribution is required and is particularly challenging.

Riemannian registration

Statistical analysis of SPD data has diverse applications ranging from medical diagnostics (PENNEC et al. 2006) to computer vision (HUANG and VAN GOOL 2017) and brain-computing interfaces (BARACHANT et al. 2012). Pre-

processing these data, despite being crucial for almost all statistical purposes, remains a delicate issue that is typically tackled with domain specific techniques that may depend on the data collection mechanism or the goal of the statistical analysis.

Thus, in Chapter 3 we present a novel registration method based on geometric considerations on the SPD manifold that is inspired by the orthogonal Procrustes analysis of matrices (GOODALL 1991). In short, our method consists in finding a reference matrix that minimizes the quantity

$$\operatorname{argmin}_{M \in \mathcal{S}^+} \sum_{i=1}^n d_{\text{AI}}(M, T(\Sigma_i))$$

where $T(\cdot)$ is a transformation that depends on the reference matrix M . We show that, under an appropriate choice of the transformation (the equivalent for the SPD manifold of the orthogonal transformation for usual matrices, discussed in Remark 1.2 in Chapter 1), this equation has an analytic solution for the eigenvalues of M . Moreover, we discuss how to implement this transformation such that it preserves the *spatial* structure of the matrices through the introduction of a rotational effort (introducing a sort of variational principle, HAMILTON 1835). To observe the implications of the RR algorithm, we perform a simulation and discuss two applications to EEG and fMRI data.

1

RIEMANNIAN GEOMETRY OF THE SYMMETRIC POSITIVE DEFINITE MANIFOLD

In this chapter we will review some results on differential geometry and, specifically, on the riemannian properties of the manifold of Symmetric and Positive-Definite (*SPD*) matrices. The results presented here are well established, and can be found in any book that tackles the subject (for instance BHATIA 2007). The notation that we are going to use (and the general framework employed to present the concepts) is mainly taken from MOAKHER 2005, slightly tweaked for consistency and convenience. Results not available in any of this two sources are cited specifically.

There are many beautiful and rich textbooks on differential geometry, such as the more classics LEE 2003; LEE 2009 or the modern SUSSMAN and WISDOM 2013, and thus we direct the interested reader to any of these for a solid background.

The first section deals with a general introduction to differential geometry and to Riemannian metrics. Section 1.II precise these notations with respect to the *SPD* manifold, introducing all the quantities required for the following chapters of this work.

I MANIFOLD NOTATION AND RIEMANNIAN GEOMETRY

In brief, a manifold \mathcal{M} is a topological space that *locally resembles* Euclidean space at each point, *i.e.* for a topological neighborhood of each point of

the manifold exists an isomorphism with an appropriate subset of the Euclidean space and is called smooth when, provided any pair of isomorphisms from the manifold and its appropriate Euclidean subset, if there is overlap then the map between the isomorphisms is smooth (see Figure 1.1 for a graphical explanation).

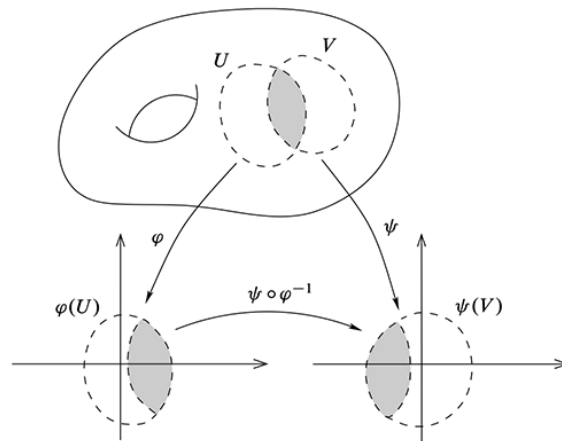


FIGURE 1.1: A schematic representation of a smooth manifold. Both U and V are topological neighborhood on the manifold which are isomorphic to a certain subset of \mathbb{R}^p for an appropriate p (through the isomorphisms $\phi: \mathcal{M} \rightarrow \mathbb{R}^p$ and $\psi: \mathcal{M} \rightarrow \mathbb{R}^p$). A *smooth* manifold is such if for any such pair U and V with a non empty intersection, the map $\psi \circ \phi^{-1}$ is smooth (differentiable with differentiable inverse).

This simple structure allows a simple study of functions defined over manifolds, thanks to the mapping to open sets of \mathbb{R}^p , but to be able to perform calculus it is necessary to introduce some quantity able to deal with derivatives. This can be done through the concepts of tangent vector, tangent space, tangent bundle and vector field, defined in the following definition (See LEE 2003).

DEFINITION 1.1 (TANGENT SPACE AND TANGENT BUNDLE): Given a point $x \in \mathcal{M}$ and the set of all smooth functions $C^\infty: \mathcal{M} \rightarrow \mathbb{R}$, a linear map $v: C^\infty(\mathcal{M}) \rightarrow \mathbb{R}$ is

a derivation at x if it satisfies

$$v(fg) = fv(g) + gv(f) \quad \forall f, g \in C^\infty(\mathcal{M}) \tag{1.1}$$

The set of all derivation at x is the tangent space of \mathcal{M} at x and is denoted by $T_x\mathcal{M}$, while an element of this space is called a tangent vector of \mathcal{M} at x . The tangent bundle $T\mathcal{M}$ is the disjoint union of tangent spaces attached to each point in \mathcal{M} , $T\mathcal{M} = \bigsqcup_x T_x\mathcal{M}$.

Finally, a vector field X is a section of the tangent bundle $T\mathcal{M}$, which can be thought as selecting a single element of the tangent space $T_x\mathcal{M}$ for every point of the manifold x .

This definition, which looks abstract, can be intuitively understood by thinking to an actual geometric vector space attached to a point of the manifold, whose elements are the tangent vector to all possible curves passing for that point, thus imagining something similar to Figure 1.2, where $w \in T_{x_1}\mathcal{M}$ is a tangent vector of \mathcal{M} at x_1 . Then, the tangent bundle is the union of the tangent spaces for each point and a vector field is the union of a single vector for each point.

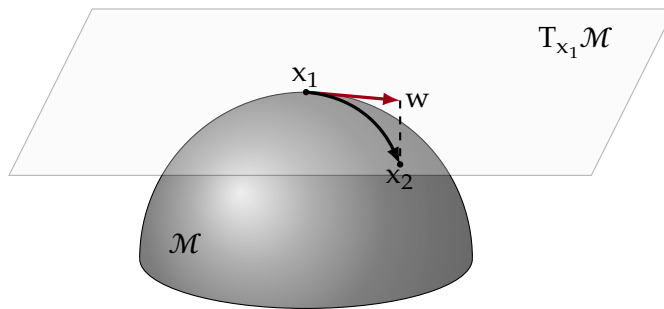


FIGURE 1.2: Depiction of the exponential and logarithm map on a manifold \mathcal{M} . The exponential map $(\exp_{x_1}(w) = x_2)$ pulls the vector w of the tangent space on to the manifold, while the logarithm map $(\log_{x_1}(x_2) = w)$ does the inverse by pushing the element x_2 to the tangent space.

Two important mathematical operations that relates the tangent space and the manifold itself are the exponential and logarithm map. Indeed,

given a point $x \in \mathcal{M}$ on the manifold and a vector $w \in T_x\mathcal{M}$ in the tangent space of x , the exponential map $\exp_x : T_x\mathcal{M} \rightarrow \mathcal{M}$ pulls the vector w on the manifold, while the logarithm map $\log_x : \mathcal{M} \rightarrow T_x\mathcal{M}$ does the inverse (a graphical representation is provided in Figure 1.2). This mapping between the tangent space and the manifold itself is sometimes called *linearization* of the manifold and it will be quite useful when defining the relations between points on the manifold.

Indeed, following the useful approach of PENNEC et al. 2006, we can use this mapping to define some of the most useful operators when dealing with usual euclidean spaces, the subtraction and the addition operators between elements of the space

	Linear space	Manifold
Subtraction	$\vec{xy} = y - x$	$\vec{xy} = \log_x(y)$
Addition	$y = x + \vec{xy}$	$y = \exp_x(\vec{xy})$

TABLE 1.1: Comparison of standard operators as usually defined when dealing with linear vector space and as defined over manifolds.

1.1 Riemannian structure

The Riemannian structure of a manifold is given by the Riemannian metric tensor, a smooth bi-linear map over the tangent bundle that is positive definite for each point

$$g_x(v, w) \equiv \langle v, w \rangle_g \geq 0 \quad \forall v, w \in T_x\mathcal{M} \text{ and } x \in \mathcal{M} \quad (1.2)$$

Once the manifold and its tangent space are equipped with an appropriate metric, it is natural to define other concepts that derive from that. An example of such a quantity is the norm of a vector, defined in the following definition.

DEFINITION 1.2 (NORM): Let $v, w \in T_x\mathcal{M}$ be two vectors in the tangent space of

a point x and let g_x be the metric evaluated in that point. Then the norm (length) of w is

$$\|w\|_g = \langle w, w \rangle_g^{1/2}. \quad (1.3)$$

Another important quantity that comes from the definition of a metric on a manifold is the concept of length of a curve. Let $\gamma: [a, b] \rightarrow \mathcal{M}$ be a smooth curve segment on the manifold, its length with respect to the metric g is

$$L_g(\gamma) = \int_a^b \|\gamma'(t)\|_g dt, \quad (1.4)$$

where $\gamma'(t)$ is the vector in $T_{\gamma(t)}\mathcal{M}$ tangent to the curve γ in $\gamma(t)$. From this it is immediate to define a notion of distance between points on a manifold, as in the following definition.

DEFINITION 1.3 (RIEMANNIAN DISTANCE): For any $x, y \in \mathcal{M}$ the (Riemannian) distance from x to y (denoted by $d_g(x, y)$) is the infimum of $L_g(\gamma)$ over all the smooth curve segments γ from x to y .

Notice that this definition of distance actually requires the manifold \mathcal{M} to be connected, which for the SPD manifold is directly satisfied.

REMARK 1.1 (ON THE RELATION BETWEEN DISTANCE AND LOGARITHMIC MAP): It is interesting to observe that, from the notation of Table 1.1, we can directly relate the distance, the metric g and the logarithm maps as

$$d_g(x, y)^2 = \|\overline{xy}\|_g^2 = \|\log_x(y)\|_g^2 = \langle \log_x(y), \log_x(y) \rangle_g \quad (1.5)$$

From Definition 1.3 it is natural to ask if the particular γ that produces the shortest length (and thus that correspond to the concept of *distance*) can be found in a direct manner, as also the interesting fact from Remark 1.1 only provides an expression for the distance, but not for the curve.

Indeed, this quantity is the *geodesic*, which in words can be defined as the

shortest path that connects two points on the manifold. The formal definition requires the notion of affine connection and its Christoffel symbols over a manifold, which is a way to represent how to connect different tangent space, but as we will not need it for our work, we will not enter that discussion. The interested reader can look to specialized texts on the subject such as PETERSEN 2016 or the already mentioned SUSSMAN and WISDOM 2013.

Finally, the last concept required for the remaining exposition is the *parallel transport*: indeed, how is it possible to compare vectors belonging to different tangent spaces? This is exemplified by the graphical representation of Figure 1.3 and in general is based upon the concept of affine connection already named in the previous paragraph.

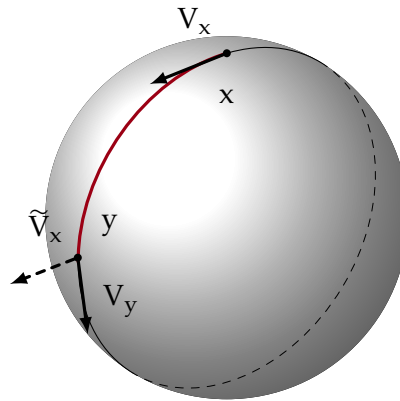


FIGURE 1.3: Depiction of the idea behind parallel transport: how to compare an element of the tangent space in y with an element of the tangent space of x .

Luckily, the manifold of SPD matrices equipped with the appropriate metric is actually one of the very few manifolds for which an easily computable expression of both the geodesic and the parallel transport exists, as we will see in the following section.

II THE SPD MANIFOLD AND ITS PROPERTIES

Let \mathcal{S}_p^+ denote the set of SPD matrices of dimension p , a subset of the set of Symmetric matrices which can be defined through one of the following two equivalent characterizations:

$$\mathcal{S}_p^+ = \{\Sigma \in \mathbb{R}^{p \times p} \mid \Sigma = \Sigma^\top, x^\top \Sigma x > 0 \forall x \in \mathbb{R}^p\}, \quad (1.6)$$

$$\mathcal{S}_p^+ = \{\Sigma \in \mathbb{R}^{p \times p} \mid \Sigma = \Sigma^\top, \lambda_{\min}(\Sigma) > 0\} \quad (1.7)$$

where $\lambda_{\min}(\Sigma)$ is the smallest eigenvalue of Σ ¹.

Yet another useful characterization is that each matrix $\Sigma \in \mathcal{S}_p^+$ can be obtained as a product GG^\top for some element $G \in GL_p$ of the General Linear group. Interestingly, this characterization can be further refined by observing that an isomorphism $\mathcal{S}_p^+ \cong GL_p/\mathbb{O}_p$ exists, as is easy to observe that $\mathcal{S}_p^+ \ni \Sigma = GG^\top$ with $G \in GL_n$ then, given any $O \in \mathbb{O}_p$, $\Sigma = GOO^\top G^\top$.

This isomorphism is important in the context of this work as it induces a Riemannian structure from the quotient group GL_p/\mathbb{O}_p (which is a Lie group). This Riemannian structure – usually referred to as the *cone of SPD matrices* and represented in Figure 1.4 for $p = 2$ – will be the main object of interest in this work.

Moreover, a final characterization (similar to the one provided in Equation (1.7)), is the one provided by the *polar* representation, obtained thanks to the eigenvalue decomposition as

$$\Sigma = \Sigma(r, U) = Ue^rU^\top \quad \text{with} \quad \begin{cases} U \in \mathbb{O}_p \\ r \in \mathbb{R}^p. \end{cases} \quad (1.8)$$

The tangent space to each point of the manifold (recall Definition 1.1) can be represented as the set of Symmetric matrices \mathcal{S}_p . This is an easy way to compute the dimension of the manifold, which appears as a sub-manifold

¹Recall that, due to the spectral theorem for real symmetric matrices, the Eigenvalue Decomposition (ED) of a $p \times p$ symmetric matrix has exactly p real eigenvalues (Axler 2015).

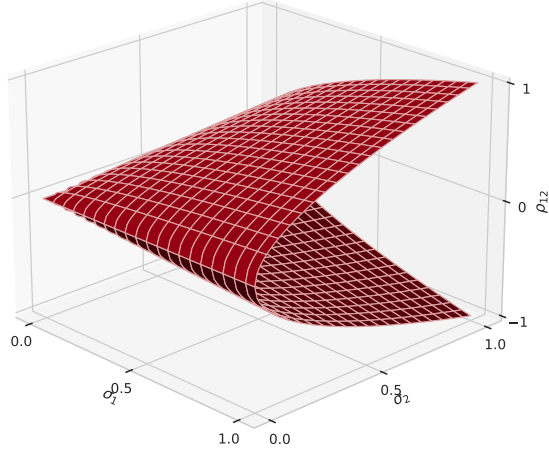


FIGURE 1.4: The cone of SPD matrices with $p = 2$ shown as an embedding in \mathbb{R}^3 . On the z axis the off-diagonal element is represented, which satisfies $|\rho_{12}| < \sqrt{\sigma_1 \sigma_2}$.

of $\mathbb{R}P^{(p+1)/2}$.

In the following we will omit the subscript p that indicate the dimension of the matrix, as it is usually clear from the context, unless necessary.

11.1 The Affine-Invariant metric

The Riemannian invariant metric on S^+ , *i. e.* the inner product between vectors on the tangent space $T_\Sigma S^+$, is called Affine-Invariant (AI) metric and can be defined as (ATKINSON and MITCHELL 1981; FÖRSTNER and MOONEN 2003)

$$g_\Sigma(W_1, W_2) = \langle W_1, W_2 \rangle_\Sigma = \text{Tr}(\Sigma^{-1} W_1 \Sigma^{-1} W_2) \quad (1.9)$$

$$\|W\|_\Sigma = \langle W, W \rangle_\Sigma. \quad (1.10)$$

Note that this metric is quite different than the euclidean (and flat) one: indeed, the euclidean metric for the general linear manifold is simply defined by the Frobenius product $\langle A, B \rangle_X = \text{Tr}(A^\top B)$, clearly independent on the point X on whose tangent space the vectors A and B resides.

Before continuing, let's recall that the matrix exponential (Exp) and matrix logarithm (Log), for an SPD matrix, can be defined starting from the ED and, thanks to the spectral theorem and to characterization (1.7), both definitions

$$\text{Exp}[\Sigma] = U \text{diag}(\exp \lambda) U^\top \quad \text{Log}[\Sigma] = U \text{diag}(\log \lambda) U^\top$$

are well behaved, where $\exp \lambda$ and $\log \lambda$ are the component-wise natural exponential and logarithm of the vector of eigenvalues. Moreover, thanks to the same properties of SPD matrices it can be proved that the square root exists and is the unique matrix

$$\Sigma^{1/2} = U \text{diag}(\lambda^{1/2}) U^\top.$$

Following PENNEC et al. 2006, to define the geodesic starting from a point on the manifold and that moves along the direction provided by a vector in the tangent space we require the invariance of the metric under congruent transformation. This enable us to use a result from the classical differential geometry (LEE 2003) that allows us to skip the computation of the Christoffel symbols for the metric, permitting to obtain

$$\gamma_{\Sigma, W}(t) = \Sigma^{1/2} \text{Exp} [t \Sigma^{-1/2} W \Sigma^{-1/2}] \Sigma^{1/2}. \quad (1.11)$$

i.e. the matrix $\Sigma(t) \in \mathcal{S}^+$ obtained by evolving the matrix $\Sigma \in \mathcal{S}^+$ along the direction $W \in T_\Sigma \mathcal{S}^+$ for a time $t \in \mathbb{R}$. The geodesic path that evolves from a point along a tangent direction is also called *retraction*.

REMARK 1.2 (ON THE CONGRUENT TRANSFORMATION): Notice that the congruent transformation that we are exploiting to obtain the geodesic Equation (1.11) are of

the form

$$A \star \Sigma = A \Sigma A^\top \quad \forall A \in \text{GL} \text{ and } \Sigma \in \mathcal{S}^+.$$

This kind of transformation makes sense from a statistical point of view as it is related to the affine transformation of a random variable. Indeed, consider a random variable $X \in \mathbb{R}^p$ and let the covariance matrix $\Sigma_{xx} = \mathbb{E} \left[(X - \bar{X})(X - \bar{X})^\top \right] \in \mathcal{S}_p^+$. If we apply a transformation $Y = AX + b$ with $A \in \text{GL}_p$ and $b \in \mathbb{R}^p$, then the covariance matrix for Y is

$$\Sigma_{yy} = \mathbb{E} \left[(Y - \bar{Y})(Y - \bar{Y})^\top \right] = A \Sigma_{xx} A^\top = A \star \Sigma_{xx}.$$

As there is a well-known relation between the geodesic and the exponential map, we are able to obtain the expression for the exponential map as

$$\begin{aligned} \exp_\Sigma : T_\Sigma \mathcal{S}^+ &\rightarrow \mathcal{S}^+ \\ W &\mapsto \exp_\Sigma W = \Sigma^{1/2} \text{Exp} \left[\Sigma^{-1/2} W \Sigma^{-1/2} \right] \Sigma^{1/2}. \end{aligned} \quad (1.12)$$

Finally, this enables also an explicit expression for the inverse of this map, *i.e.* the logarithm map

$$\begin{aligned} \log_\Sigma : \mathcal{S}^+ &\rightarrow T_\Sigma \mathcal{S}^+ \\ \Omega &\mapsto \log_\Sigma \Omega = \Sigma^{1/2} \text{Log} \left[\Sigma^{-1/2} \Omega \Sigma^{-1/2} \right] \Sigma^{1/2}, \end{aligned} \quad (1.13)$$

which, in turn, finally enable an explicit expression for the distance between two geodesics, using the observation from Remark 1.1:

$$\begin{aligned} d_{\text{ai}}(\Sigma_1, \Sigma_2)^2 &= \left\| \log_{\Sigma_1}(\Sigma_2) \right\|_{\text{ai}}^2 \\ &= \text{Tr} \left[\Sigma_1^{-1} \log_{\Sigma_1} \Sigma_2 \Sigma_1^{-1} \log_{\Sigma_1} \Sigma_2 \right] \\ &= \text{Tr} \left[\text{Log}^2 \Sigma_1^{-1/2} \Sigma_2 \Sigma_1^{-1/2} \right] \\ &= \sum_i^p \log^2 \lambda_i \left(\Sigma_1^{-1/2} \Sigma_2 \Sigma_1^{-1/2} \right). \end{aligned} \quad (1.14)$$

REMARK 1.3 (ON THE INTERPRETATION OF THE AI METRIC): *It is useful, for an interpretation of the AI metric, to observe that Equation (1.14) implies that the distance of any matrix from the identity matrix increase more and more when one of the eigenvalues of the matrix gets close to zero. Getting back to Figure 1.4, this means that the matrices on the shown border that have at least one eigenvalue equal to zero are infinitely far from all the matrix in the interior of the manifold.*

II.II Other useful quantities

We discuss here briefly some other Riemannian quantities of the SPD manifold. First, it is simple to write the geodesic between two points $\Sigma_1, \Sigma_2 \in \mathcal{S}^+$

$$\begin{aligned} \tilde{\gamma}: [0, 1] &\rightarrow \mathcal{S}^+ \\ t &\mapsto \tilde{\gamma}_{\Sigma_1 \Sigma_2}(t) = \Sigma_1^{1/2} \left(\Sigma_1^{-1/2} \Sigma_2 \Sigma_1^{-1/2} \right)^t \Sigma_1^{1/2} \end{aligned} \quad (1.15)$$

Another important quantity is the parallel transport, explained intuitively in Figure 1.3 and of which we provide here two alternative definitions. The first one is related conceptually to Equation (1.11) and provide the expression to transport a vector $V \in T_{\Sigma} \mathcal{S}^+$ from the tangent space to the point Σ along the geodesic with initial direction $W \in T_{\Sigma} \mathcal{S}^+$ and length t :

$$\Pi_{\Sigma, W}(t)(V) = \gamma_{\Sigma W} \left(\frac{t}{2} \right) \Sigma^{-1} V \Sigma^{-1} \gamma_{\Sigma W} \left(\frac{t}{2} \right). \quad (1.16)$$

Similarly, one can define the parallel transport of $V \in T_{\Sigma_1} \mathcal{S}^+$ from the tangent space at the point Σ_1 to the tangent space at the point Σ_2 as

$$\tilde{\Pi}_{\Sigma_1, \Sigma_2}(V) = \left(\Sigma_1^{-1/2} \Sigma_2 \Sigma_1^{-1/2} \right)^{\frac{1}{2}} V \left(\Sigma_1^{-1/2} \Sigma_2 \Sigma_1^{-1/2} \right)^{\frac{1}{2}}. \quad (1.17)$$

2 | RIEMANNIAN OPTIMIZATION ON THE SPD MANIFOLD

In different modern statistical applications, mostly enabled by the recent availability of new data sources, the main difficulties are not only related to the large scale of the problem but also from the complex constraints that the data or the model's parameters require. A primer example is the Symmetric and Positive-Definite (SPD) manifold, discussed extensively from a geometric point of view in Chapter 1 and that appears in statistics in various context, both when dealing with covariance matrix-valued data (see for instance the data in BARACHANT et al. 2012) or when tackling problems with covariance matrix parameters.

In this chapter we will focus on the latter, and more specifically we will focus on how to solve the general problem of optimization

$$\min_{\Sigma \in \mathcal{S}^+} f(\Sigma) \tag{2.1}$$

where \mathcal{S}^+ is the SPD manifold and $f: \mathcal{S}^+ \rightarrow \mathbb{R}$ is an at least once-differentiable, convex function. Notice that the concept of convex function has to be interpreted in the sense of geodesic convexity such as described by RAPCSÁK 1991.

To solve this problem, an Euclidean approach is to transform the matrix Σ with a Cholesky decomposition such that $\Sigma = LL^T$ and, after a vectorization procedure to create a $p(p + 1)$ dimensional vector by stacking the columns below the diagonal, to perform a standard Euclidean optimization. Of course, depending on the specific problem to be solved other procedures

might be used, such as the Expectation-Maximization algorithm (DEMPSTER et al. 1977) or in specific examples a fixed point procedure (BURDEN et al. 1985).

In this chapter, instead, we will describe and implement a general alternative approach that leverages the properties of the Riemannian manifold \mathcal{S}^+ . This procedure is comparable (and an alternative to) the Cholesky-Euclidean approach broadly described above, which is generally applicable whenever it is required to optimize a function of a covariance matrix. In the next section we describe some implementation detail of the algorithms, while in Section 2.11 a simulation study on the performance of this algorithm compared to the Cholesky-Euclidean one is presented.

Our approach is inserted in the general field of manifold and specifically matrix manifold optimization (GABAY 1982; ABSIL et al. 2008), which have been in development for several years. Various attempts have been made to solve this specific problem (see for instance PENNEC et al. 2006; BINI and IANNAZZO 2013), but in our opinion a thorough investigation and formalization is still lacking.

I OPTIMIZATION ALGORITHMS

Line search optimization methods are basically constructed by means of an optimization iterative scheme usually constructed as shown in Algorithm 2.1. More details on how this kind of numerical optimization schemes works in general can be found on the widely known NOCEDAL and WRIGHT 2006.

The function `FINDDIRECTION` in Line 3 is used to find the appropriate direction for the evolution of the algorithm, and is in general related to the gradient of the objective function f evaluated in x_k , while the actual functional form depends on the specific version of the algorithm. The basic *Steepest Descent* algorithm, indeed, uses simply $d_k = -\text{Grad}_x f(x_k) = -\nabla f_k$ as the descent direction, thus updating along the direction of maximal de-

ALGORITHM 2.1: Standard Newton-like optimization procedure.

Input: $f: \mathbb{R}^p \rightarrow \mathbb{R}$ ▷ Objective Function
▷ Starting point

```

1: function NEWTONLIKE( $x_0$ )
2:   for  $k = 0, \dots$  do
3:      $d_k \leftarrow$  FINDDIRECTION( $x_k$ )
4:      $\eta_k \leftarrow$  FINDSTEPLENGTH( $x_k, d_k$ )
5:      $x_{k+1} \leftarrow x_k + \eta_k d_k$ 
6:     if Convergence then Break
7:   return  $x_k$ 

```

crease of the function.

The function FINDSTEPLENGTH of Line 4, instead, is used to select an appropriate step length. In its most naïve version, it is a fixed length (for instance $\eta_k = 1$) for every step. In our work we used an implementation of algorithms 3.5 and 3.6 from NOCEDAL and WRIGHT 2006, which provide a step length that satisfies the *strong Wolfe conditions*

$$f(x_k + \eta_k d_k) \leq f_k + c_1 \eta_k \nabla f_k^\top d_k, \quad (2.2a)$$

$$|\nabla f(x_k + \eta_k d_k)^\top d_k| \leq c_2 |\nabla f_k^\top d_k|, \quad (2.2b)$$

where $0 < c_1 < c_2 < 1$ are two hyperparameters of the algorithm.

This general structure, though, is only applicable when the parameter space to be explored by the optimization procedure corresponds to the flat Euclidean vector space, as the Riemannian structure of a non-flat manifold introduces some changes.

The first is the fact that, as we have discussed in Section 1.1 and more specifically as shown in Table 1.1, the translation of an element of the manifold along the direction provided by an element of the tangent space to the manifold in that point is not simply the sum, but the exponential map has to be used.

Thus, Line 5 of Algorithm 2.1 (and also every time that a translation appears, such as in Equation (2.2)) has to become $x_{k+1} \leftarrow \exp_{x_k}(\eta_k d_k)$ for

an appropriate definition of the exponential map, which for the Affine-Invariant (AI) metric is provided in Equation (1.12)

$$\exp_{x_k}(\eta_k d_k) = x_k^{1/2} \text{Exp} \left[\eta_k x_k^{-1/2} d_k x_k^{-1/2} \right] x_k^{1/2} \quad (2.3)$$

where $\text{Exp}[M]$ is the matrix exponential which for SPD matrices can be expressed simply by obtaining the exponential of the eigenvalues from the Eigenvalue Decomposition (ED).

Moreover, in Equation (2.2) and in many of the computation of the descent direction, inner products between derivations of the function f appears, such as $\nabla f_k^\top d_k$. In a Riemannian context, as discussed in Section 1.1 and specifically in Definition 1.1, this objects are elements of the tangent space $T_{x_k} \mathcal{M}$, and thus their inner product is induced by the metric g that equips that manifold. In the context of the SPD manifold, as defined in Equation (1.9), the inner product becomes

$$\langle \nabla f_k, d_k \rangle_{x_k} = \text{Tr} (x_k^{-1} \nabla f_k x_k^{-1} d_k) \quad (2.4)$$

Finally, also the gradient has to be adapted, as its definition is on the Euclidean space: following the definition of AMARI 2016, we introduce the Natural Gradient on the SPD manifold as

$$\underline{\nabla} f_k = x_k \frac{(\nabla f_k + \nabla f_k^\top)}{2} x_k. \quad (2.5)$$

In the following, as we will only use the natural gradient to perform our computations, we will drop the underline and reserve the symbol ∇f_k to define the natural gradient of the function f evaluated in x_k .

1.1 Newton and quasi-Newton methods

A first generalization of this method is to consider a more elaborate expression for d_k . Indeed, the steepest descent algorithm discussed earlier is most of the time very slow to converge for complex problems and a more suitable

descent direction is required. The most important direction to be considered is the Newton direction, that requires the computation of the Hessian function $\text{Hess}_x f(x_k) = \text{Hf}_k$ and is expressed as

$$\mathbf{d}_k = -(\text{Hf}_k)^{-1} \nabla f_k.$$

In most cases, though, computing explicitly the full Hessian matrix is a daunting task, both analytical and computational. In our scenario, where the function is matrix valued, computing the Hessian matrix without resorting to complicated parametrization is almost impossible, and thus an appealing solution is offered by the so called *quasi-Newton* family of algorithm, of which we will discuss only the Limited memory BFGS (**L-BFGS**) algorithm.

In almost all the algorithms of this family, the inverse hessian factor is approximated using information on the gradient at past steps. To do so, though, one need to compare vectors that, in the case of a non-Euclidean manifold, are defined in different vector spaces (*i.e.* in the tangent space to different point of the manifold). To do so, as discussed in Sections 1.1 and 1.11.11, we define the vector transport for a generic vector $V \in T_{x_k} \mathcal{S}^+$ as

$$\Pi_{x_k, \mathbf{d}_k}(\eta_k)(V) = \exp_{x_k} \left(\frac{\eta_k}{2} \mathbf{d}_k \right) x_k^{-1} V x_k^{-1} \exp_{x_k} \left(\frac{\eta_k}{2} \mathbf{d}_k \right) \quad (2.6)$$

The general theory behind such algorithms are found in many classical textbook on numerical optimization (we followed NOCEDAL and WRIGHT 2006 for our implementations), while in Algorithm 2.2 we report only the general scheme adopted. In Appendix A details on the procedures FINDDIRECTION, FINDSTEPLENGTH and UPDATEHISTORY for our Riemannian Limited memory BFGS (**R-L-BFGS**).

1.11 Approximate retraction and vector transport

Notice that, from a numerical standpoint, both Equations (2.3) and (2.6) present the repeated computation of a matrix exponential which requires

ALGORITHM 2.2: Riemannian quasi-Newton optimization procedure.

Input: $f: \mathcal{S}^+ \rightarrow \mathbb{R}$ \triangleright Objective Function
 \triangleright Starting point

```

1: function R-QUASINEWTON( $x_0$ )
2:   for  $k = 0, \dots$  do
3:      $d_k \leftarrow$  FINDDIRECTION( $x_k$ , History)
4:      $\eta_k \leftarrow$  FINDSTEPLENGTH( $x_k, d_k$ )
5:      $x_{k+1} \leftarrow$   $\exp_{x_k}(\eta_k d_k)$ 
6:     History  $\leftarrow$  UPDATEHISTORY( $x_k, d_k, \eta_k, \dots$ )
7:     if Convergence then Break
8:   return  $x_k$ 

```

around $\mathcal{O}(n^3)$ operations and, moreover, is highly unstable for long step-size or large gradient values.

A workaround is to use an approximate version of these two expressions, based on the second order Taylor expansion of the exponential

$$\exp_{x_k}(\eta_k d_k) \approx x_k + \eta_k d_k + \frac{\eta_k^2}{2} d_k x_k^{-1} d_k. \quad (2.7)$$

It is important the use of the second order term as it ensures that the resulting matrix is still an element of the SPD manifold (indeed, using only the first term we would recover the euclidean step which, as d_k is not positive definite, may have undesired consequences).

The same approximation can be performed also for Equation (2.6), obtaining

$$\begin{aligned} \Pi_{x_k, d_k}(\eta_k)(V) &\approx V + \eta_k d_k x_k^{-1} V \\ &\quad + \frac{\eta_k^2}{2} \left(d_k x_k^{-1} V + \frac{1}{2} V x_k^{-1} d_k \right) x_k^{-1} d_k. \end{aligned} \quad (2.8)$$

1.iii Euclidean and product manifold

Notice that Algorithm 2.2, aside from the specific expression of the geometric quantities such as the inner product or the exponential map, is inde-

pendent from the particular manifold of interest. As such, we defined two other variants that we will now briefly discuss. The first one is an Euclidean version of the algorithm: indeed, the flat Euclidean space is actually a Riemannian manifold by itself and, as such, can be described within the same framework. In this case, the geodesic expression is the simple sum between elements, the inner product is the usual dot product and so on. Of course, this results in the exact same algorithm as the one defined in NOCEDAL and WRIGHT 2006.

The second variant, slightly more interesting, regards the *product manifold*: in many practical cases, the actual parameter space is not a single manifold but a combination of different ones. For instance, in the simple case of a multivariate normal $\mathcal{N}_p(\mu, \Sigma)$ the parameter space is $\mathcal{M} = \mathbb{R}^p \times \mathcal{S}_p^+$. To tackle this situation, we defined a suitable class of manifolds that allows to flexibly specify such a product. Then, each one of the quantities discussed before is defined as the appropriate combination of the single manifold quantities. Continuing in the example of \mathcal{M} defined earlier, the exponential map would be computed component-wise as a simple vector sum for the real component and as Equation (2.3) for the SPD component. The same happens for the parallel transport, while the inner product is computed as the sum of the inner product of each component, as the tangent space to the product manifold is canonically isomorph to the direct sum of the tangent spaces. Assuming then that $\xi, \zeta \in T_x \mathbb{R}^p$ and $\eta, \nu \in T_x \mathcal{S}_p^+$ are the components of two tangent vectors, their inner product can be written as

$$\langle \xi \oplus \eta, \zeta \oplus \nu \rangle_{\mathcal{M}} = \langle \xi, \zeta \rangle_{\mathbb{R}} + \langle \eta, \nu \rangle_{\mathcal{S}^+}.$$

II SIMULATION STUDY

We performed a simulation study to test the performance of this algorithm and to compare to standard methods for the optimization of a function defined on the SPD manifold.

As an objective function, a natural choice is to define a log-likelihood function and to perform optimization in order to obtain a Maximum Likelihood Estimator (MLE) estimation of the parameters of this likelihood.

In order to compare our method to a classical one, we implemented a simple Cholesky-Euclidean approach: given a function $f(\Sigma)$ of a SPD variable, we first use the Cholesky decomposition to obtain a triangular (with unconstrained entries) matrix and we then flatten the non-zero entries in a column vector with dimension $p(p+1)/2$. Finally, a standard Euclidean algorithm is employed to optimize this new function.

We first illustrate the performance of the proposed method using a toy example. Specifically, we simulated a simple multivariate normal sample $\{y_i\}_i^n$ with $y_i \sim \mathcal{N}(\mu, \Sigma)$. The log-likelihood for the parameters μ and Σ is

$$\mathcal{L}(\mu, \Sigma | y) = \kappa - \log |\Sigma| - \frac{1}{2} \sum_i^n (y_i - \mu)^\top \Sigma^{-1} (y_i - \mu). \quad (2.9)$$

To improve the performance and stability of the optimization we employ a re-parametrization of Equation (2.9) that ensures geodesic convexity to the function (VISHNOI 2018). We augment the sample vectors $\bar{y}_i = [y_i^\top, 1]^\top$. This allows us to consider an extended log-likelihood function

$$\bar{\mathcal{L}}(S | \bar{y}) = \kappa - \log |S| - \frac{1}{2} \sum_i^n \bar{y}_i^\top S^{-1} \bar{y}_i$$

where $S \in \mathcal{S}_{p+1}^+$. Via simple algebra, it can be shown that the MLE estimator \hat{S} can be expressed as a function of $\hat{\mu}$ and $\hat{\Sigma}$ as

$$\hat{S} = \begin{pmatrix} \hat{\Sigma} + \hat{\mu} \hat{\mu}^\top & \hat{\mu} \\ \hat{\mu}^\top & 1 \end{pmatrix}.$$

We repeated the simulation for varying p (the dimension of the sample space), generating $r = 50$ independent datasets with sample size $n = 1000$ for each p . To generate the *true parameters* we draw each component of the mean vector from independent normal distributions, while the variance-

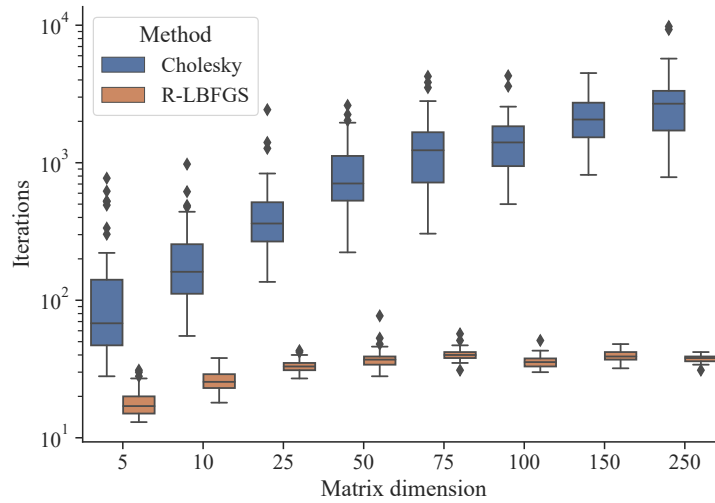
covariance matrix is drawn from a Wishart distribution with large degree of freedom ($\nu = 10p$) and location parameter \mathbb{I}/ν , to keep its expected value equal to \mathbb{I} .

The results obtained via this simulation are reported in Table 2.1, and a graphical representation is also shown in Figure 2.1. As in this example the analytical expression of the MLE is known, we are able to accurately check the convergence of each algorithm.

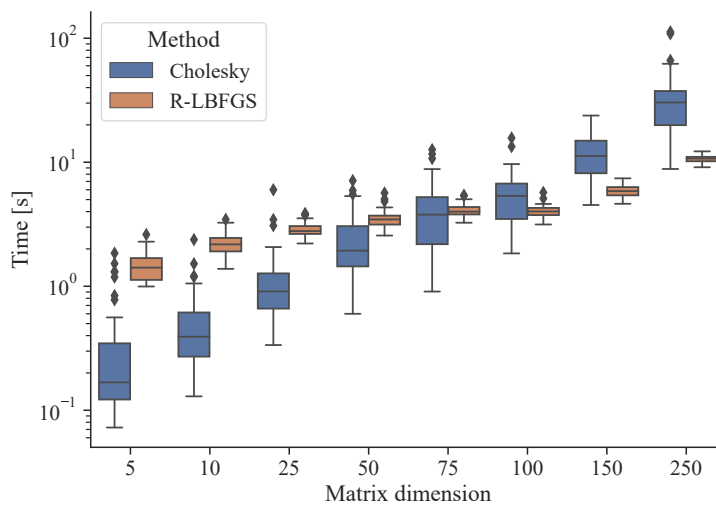
TABLE 2.1: Results for the time and iterations required to find the MLE of a multivariate normal distribution, comparing the Cholesky-Euclidean method with the R-L-BFGS algorithm described in Appendix A. The median and the 95% range are provided.

p	Algorithm	Iterations		Time [s]	
		Median	95% range	Median	95% range
5	Cholesky	68.0	[30.2, 602.2]	0.17	[0.07, 1.48]
	R-L-BFGS	17.0	[13.2, 29.5]	1.41	[1.01, 2.28]
10	Cholesky	161.0	[58.7, 587.4]	0.39	[0.14, 1.45]
	R-L-BFGS	25.5	[20.0, 36.3]	2.18	[1.59, 3.18]
25	Cholesky	361.5	[188.1, 1374.3]	0.93	[0.47, 3.36]
	R-L-BFGS	33.0	[28.2, 41.6]	2.79	[2.35, 3.78]
50	Cholesky	706.5	[253.7, 2214.6]	2.38	[0.89, 5.87]
	R-L-BFGS	37.0	[31.0, 52.0]	3.45	[2.79, 5.01]
75	Cholesky	1232.0	[391.2, 3776.2]	4.69	[1.46, 11.46]
	R-L-BFGS	40.0	[35.2, 50.1]	3.99	[3.36, 5.29]
100	Cholesky	1405.0	[553.6, 3393.0]	6.74	[2.67, 12.64]
	R-L-BFGS	35.5	[30.0, 42.8]	4.01	[3.28, 4.99]
150	Cholesky	2062.0	[901.9, 3753.2]	13.99	[6.08, 20.61]
	R-L-BFGS	39.0	[34.0, 47.5]	5.85	[4.85, 7.21]
250	Cholesky	2687.0	[1303.0, 8951.5]	30.36	[14.97, 104.46]
	R-L-BFGS	38.0	[34.0, 42.0]	10.71	[9.54, 11.89]

Notice how the number of iterations required by our algorithm does not show an exponential growth with the matrix size, a behavior instead clearly shown by the Cholesky-Euclidean algorithm. The reason for this behavior is



(A) Iterations



(B) Time

FIGURE 2.1: Results for the time and iterations required to find the MLE of a multivariate normal distribution, comparing the Cholesky-Euclidean method with the R-LBFGS algorithm described in Appendix A. The upper panel shows the number of needed iteration to reach convergence while the lower one shows the time in seconds spent by the algorithm.

related to the more efficient exploration of the parameter space when using the geodesic step: while the dimension of the manifold (and thus the euclidean space obtained via Cholesky decomposition) grows quadratically with the dimension p , the Cholesky-Euclidean method becomes quickly affected by the curse of dimensionality, while our **R-L-BFGS**, by using the natural gradient and the geodesic expression to perform the update (which is, recalling from Chapter 1, the shortest path available) maintains a high efficiency and keeps the “effective” dimension low.

As a second example, we consider the skew-normal density function from **AZZALINI and DALLA VALLE 1996**, defined as

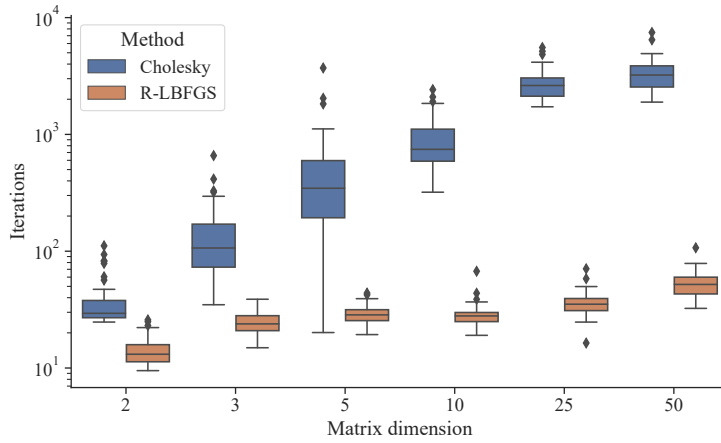
$$\mathcal{L}(\mu, \Sigma, \xi | y) = 2\phi_{\Sigma}(y - \mu)\Phi(\xi^{\top}(y - \mu)) \quad (2.10)$$

where $\phi_{\Sigma}(y - \mu)$ is the multivariate normal with mean μ and variance-covariance Σ , while $\Phi(x)$ is the cumulative distribution of the univariate standard normal distribution. This definition follows the parametrization (Ω, α) discussed in **AZZALINI and CAPITANIO 1999; AZZALINI 2013**, where Σ plays the role of Ω and α is represented by ξ . As in the previous example, we generated $r = 50$ repetitions for every p (problem’s dimension). To generate the *true parameters* we simulate each component of the location and α parameter as independent identically distributed standard normal variables, while the Ω parameter is obtained from a Wishart with large number of degree of freedom ($v = 10p$) and mean \mathbb{I}/v . Then, a sample of size of $n = 1000$ is generated following the implementation of R package `sn` (**AZZALINI 2021**), described in **AZZALINI and CAPITANIO 1999** and rewritten in Python.

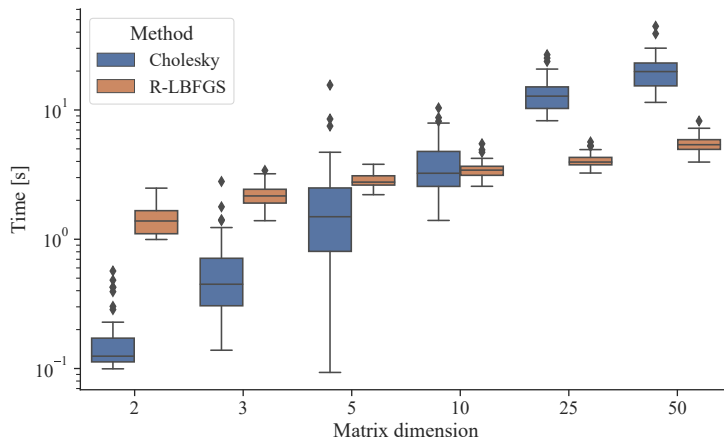
Notice that, differently from the re-parametrized normal case, in this example it is necessary to perform the optimization over the product manifold $\mathcal{M} = \mathbb{R}^p \times \mathcal{S}_p^+ \times \mathbb{R}^p$, as described in Section 2.1.III. The result of this simulation are recorded in Table 2.2, and they show the same characteristics as those in the multivariate normal case.

TABLE 2.2: Results for the time and iterations required to reach convergence searching the MLE of a multivariate skew-normal distribution, comparing the Cholesky-Euclidean method with the R-L-BFGS algorithm described in Appendix A. The median and the 95% range are provided.

p	Algorithm	Iterations		Time [s]	
		Median	95% range	Median	95% range
2	Cholesky	29.4	[25.0, 91.5]	0.12	[0.10, 0.47]
	R-L-BFGS	13.1	[9.7, 24.5]	1.38	[1.01, 2.28]
3	Cholesky	106.5	[37.3, 395.1]	0.44	[0.15, 1.70]
	R-L-BFGS	23.8	[17.3, 36.8]	2.16	[1.59, 3.18]
5	Cholesky	345.9	[64.5, 1990.2]	1.49	[0.27, 8.29]
	R-L-BFGS	28.5	[21.2, 41.7]	2.77	[2.35, 3.78]
10	Cholesky	745.0	[347.1, 2070.5]	3.24	[1.56, 8.67]
	R-L-BFGS	27.9	[22.0, 42.7]	3.43	[2.79, 4.87]
25	Cholesky	2622.5	[1807.4, 5089.2]	12.79	[8.71, 24.91]
	R-L-BFGS	35.2	[25.2, 56.3]	3.96	[3.36, 5.29]
50	Cholesky	3221.8	[1969.2, 6146.7]	19.82	[11.95, 37.16]
	R-L-BFGS	51.9	[32.4, 77.8]	5.38	[4.18, 7.22]



(A) Iterations



(B) Time

FIGURE 2.2: Results for the time and iterations required to find the MLE of a multivariate skew-normal distribution, comparing the Cholesky-Euclidean method with the R-LBFGS algorithm described in Appendix A. The upper panel shows the number of needed iteration to reach convergence while the lower one shows the time in seconds spent by the algorithm.

III SKEW-NORMAL VARIATIONAL APPROXIMATION

As an example of a use case in which our algorithm might be applied, we discuss now the use of our method within the context of Variational Approximation (VA) (ORMEROD and WAND 2010; BLEI et al. 2017).

As a short introduction to fix the notation, we recall that in this context we consider a Bayesian model parametrized through θ and with observed data y and we are interested in the posterior distribution

$$p(\theta | y) = \frac{p(y, \theta)}{p(y)}. \quad (2.11)$$

As known, marginalization over the parameter space $p(y)$ is often difficult to compute or downright intractable, and thus alternative approaches abound, the most common by far being the class of Monte-Carlo based techniques, of which the Markov Chain Monte Carlo is a primer example. Starting from the field of Computer Science and recently seeing a wider adoption also in the statistics literature, the *variational approximations* is a framework of deterministic techniques to obtain an approximate version of the posterior distribution (2.11).

If we define an arbitrary density q over the space of θ , the marginalized log-likelihood satisfies

$$\begin{aligned} \log p(y) &= \int q(\theta) \log p(y) d\theta = \int q(\theta) \log \frac{p(y, \theta)/q(\theta)}{p(\theta | y)/q(\theta)} d\theta \\ &= \int q(\theta) \log \frac{p(y, \theta)}{q(\theta)} d\theta + \int q(\theta) \log \frac{q(\theta)}{p(\theta | y)} d\theta \\ &\geq \int q(\theta) \log \frac{p(y, \theta)}{q(\theta)} d\theta, \end{aligned}$$

with the inequality arising from the properties of the Kullback-Leibler divergence (KULLBACK and LEIBLER 1951) between $q(\cdot)$ and $p(\cdot | y)$.

The essence of the variational approximation method is then to approximate the exact posterior $p(\theta | y)$ with a more tractable approximation $q(\theta)$

obtained by restricting in some way the class of available density function q and then finding the most *similar* of this densities to the target posterior. Notice that maximizing the variational lower bound

$$p_q(y) = \exp \int q(\theta) \log \frac{p(y, \theta)}{q(\theta)} d\theta \quad (2.12)$$

over the chosen class of function q is equivalent to minimizing the Kullback-Liebr divergence between the target posterior and the approximating function.

The most common restriction that are imposed on q are of two types.

PRODUCT DENSITY TRANSFORMS: where $q(\theta)$ factorizes as $\prod_i q_i(\theta_i)$ for some partition of θ . It is also known as *mean field* approximations in the statistical physics literature (PARISI 1988).

PARAMETRIC DENSITY TRANSFORMS: $q(\theta)$ is a member of a parametric family of density function $q(\theta; \zeta)$.

The second approach is less widely adopted (at least in the Computer Science literature) albeit it is recently gaining popularity due to its superior applicability to Generalized Linear Mixed Models (GLMMs). It is based on finding a flexible and at the same time tractable enough parametric family of densities. The main family used in this context is the multivariate normal family, which then constitutes the so called Gaussian Variational Approximation (GVA) method (ORMEROD and WAND 2012).

Observe that maximizing Equation (2.12), even if q is a known and simple density, might be difficult or expensive if the dimension of the parameter space is large or the parameters have complex constrains. A classical example, is the situation (typical for instance in the GVA method) where one of the parameter is a SPD matrix, for instance the covariance matrix parameter of a multivariate normal distribution. In that case, the computational costs of the optimization scales badly with the dimension of the parameter space, as we have seen from the simulation results reported Section 2.II. For the GVA, one can derive a tailor made update step that exploits the structure

of the multivariate normal density function to simplify the computations (WAND 2014), but changing the approximating density to something more flexible loses this kind of optimizations.

An example of such situation is the one that arises in the context of the Skew-Normal Variational Approximation (SNVA) method, discussed in the unpublished work by ORMEROD 2011. In this case, optimization is slowed compared to other methods by the fact that one of the parameter is an (in principle large) SPD matrix. As such, the Riemannian optimization technique discussed in Section 2.1 can be used to stabilize and enhance the performances of the method.

Once the maximization of the lower bound (2.12) has provided an estimated parameter $\hat{\zeta}$, the optimal approximation to the posterior function $p(\theta | y)$ is then given by $q(\theta; \hat{\zeta})$.

III.1 The Skew-Normal variational approximation

The SNVA is defined as the VA when the parametric family of densities $q(\theta; \zeta)$ is the multivariate skew-normal distribution $\text{SN}(\mu, \Sigma, \zeta)$ discussed in AZZALINI and DALLA VALLE 1996, using the same parametrization of Equation (2.10)

$$q(\theta; \mu, \Sigma, \zeta) = 2\phi_{\Sigma}(\theta - \mu)\Phi(\zeta^{\top}(\theta - \mu)).$$

Notice that the SNVA is guaranteed to have a tighter lower bound (2.12) on $p(y)$ than the simple GVA as

$$p(y) \geq \sup_{\mu, \Sigma, \zeta} [p_{\text{sn}}(y; \mu, \Sigma, \zeta)] \geq \sup_{\mu, \Sigma, \zeta=0} [p_{\text{sn}}(y; \mu, \Sigma, \zeta)] = \sup_{\mu, \Sigma} [p_{\text{n}}(y; \mu, \Sigma)],$$

a consequence of the fact that when $\zeta = 0$ the multivariate Skew-Normal reduces to the usual multivariate normal distribution.

The explicit expression of $p_{\text{sn}}(y | \mu, \Sigma, \zeta)$ from Equation (2.12) for the skew-normal distribution can be readily computed using the expression

of the entropy of the skew-normal distribution as obtained for instance in CONTRERAS-REYES and ARELLANO-VALLE 2012, obtaining the following result for the log-lower bound

$$\begin{aligned} \log p_{\text{sn}}(y \mid \mu, \Sigma, \xi) &= \frac{1}{2} \log |2\pi\Sigma| + \frac{m}{2} - \log 2 \\ &\quad - \Psi(\xi^\top \Sigma \xi) + f_{\text{SN}}(y \mid \mu, \Sigma, \xi) \end{aligned} \quad (2.13)$$

where

$$\begin{aligned} f_{\text{sn}}(y \mid \mu, \Sigma, \xi) &= \mathbb{E}_{q(\theta)} [\log p(y, \theta)] \\ &= \int_{\mathbb{R}^m} 2\phi_{\Sigma}(\theta - \mu) \Phi(\xi^\top (\theta - \mu)) \log p(y, \theta) d\theta \\ \Psi(\sigma^2) &= \int_{\mathbb{R}} 2\phi_{\sigma}(z) \Phi(z) \log \Phi(z) dz. \end{aligned} \quad (2.14)$$

III.II An actual example: the normal sample

As a toy example, we now discuss a simple case of a normal sample with very small sample size and suitable conjugate prior distributions. Specifically, assume

$$y_i \sim \mathcal{N}(v, \sigma^2) \quad (2.15)$$

for $i = 1, \dots, n$ and prior distribution on v and σ^2 given by

$$v \sim \mathcal{N}(0, \sigma_v^2), \quad \sigma^2 \sim \text{IG}(a, b).$$

To use the [SNVA](#) method we use the transformation $\sigma^2 = e^\gamma$ and we write

the posterior distribution density for the parameter vector $\theta = [v, \gamma]$

$$\begin{aligned} \log p(\theta | y) = & - \left(a + \frac{n}{2} \right) \langle t, \theta \rangle - \frac{\langle r, \theta \rangle^2}{2\sigma_v^2} \\ & - e^{-\langle t, \theta \rangle} \left(b + \frac{\sum_i^n (y_i - \langle r, \theta \rangle)^2}{2} \right) + \kappa \end{aligned} \quad (2.16)$$

where $t = (0, 1)^\top$ and $r = (1, 0)^\top$ allows to select the single components of the parameter vector.

Appendix B provides details on the computations required to obtain the full expression for f_{sn} in this case, given by

$$\begin{aligned} f_{\text{sn}}(y | \mu, \Sigma, \xi) = & - \left(a + \frac{n}{2} \right) \left(\langle t, \mu \rangle + \sqrt{\frac{2}{\pi}} \langle t, \delta \rangle \right) - M \left(b + \frac{1}{2} T \right) \\ & - \frac{1}{2\sigma_v^2} \left(r^\top \Sigma r + \langle r, \mu \rangle^2 + 2\sqrt{\frac{2}{\pi}} \langle r, \mu \rangle \langle r, \delta \rangle \right) \end{aligned} \quad (2.17)$$

where $\delta = \frac{\Sigma \xi}{\sqrt{1 + \xi^\top \Sigma \xi}}$ and

$$\begin{aligned} M &= 2 \exp \left[- \langle t, \mu \rangle + \frac{t^\top \Sigma t}{2} \right] \Phi \left(- \langle t, \delta \rangle \right) \\ T &= \left[\sum_i (y_i - \tilde{\mu})^2 + n \left(r^\top \Sigma r + \zeta_2 \left(- \langle t, \delta \rangle \right) \langle r, \delta \rangle^2 \right) \right] \\ \tilde{\mu} &= \langle r, \mu \rangle - r^\top \Sigma t + \zeta_1 \left(- \langle t, \delta \rangle \right) \langle r, \delta \rangle \\ \zeta_k(x) &= \frac{d^k \log \Phi(x)}{dx^k}. \end{aligned}$$

In order to perform the maximization of the lower bound p_{sn} of Equation (2.13) an optimization over the parameters of the skew-normal distribution is required, which is difficult due to the presence of $\Sigma \in \mathcal{S}^+$.

In this scenario the optimal variational parameter $\hat{\Sigma}$ can be estimated by applying the R-L-BFGS algorithm described in Section 2.1. The use of such an algorithm provides an advantage in stability and performance over the

standard Euclidean procedure as we have seen in Section 2.ii. Moreover, the internals of our implementation provide an automatic differentiation approach to the gradient computation, leveraging the excellent performance of the JAX library (BRADBURY et al. 2018). This allows the user to simply specify the function (in this case p_{sn}) without needing to worry about explicitly writing the vectorization function and its derivatives.

We performed a small simulation, generating $r = 50$ datasets from Equation (2.15) with $n = 6$ (to keep a high degree of skewness, see ORMEROD 2011) and simulating ν and σ^2 from their prior distributions. Our algorithm always reach the same result (within a 10^{-3} relative error) of the standard euclidean Cholesky algorithm and does so in a stable number of iterations (33 ± 2), differently from the Euclidean version which varies widely (the minimum number of iterations is as low as 42 but it can go as high as 562 in one case, with an average of 112 ± 69).

3

RIEMANNIAN REGISTRATION OF MATRIX

DATA

Data in the form of Symmetric and Positive-Definite (SPD) matrices are collected in a variety of areas, ranging from medical imaging (PENNEC et al. 2006) to computer vision (HUANG and VAN GOOL 2017). Before data analysis and inference, it is of paramount importance to focus on data registration and alignment, consistently with successful data registration approaches introduced for functional data (RAMSAY and SILVERMAN 2002) or geomics data (McCARTHY et al. 2017).

As illustrative application, consider Electroencephalography (EEG) data reported in CATTAN et al. 2018. This dataset is just an example of the data used in a modern and interesting field, and collects the EEG data of a number of subjects performing some simple tasks. One of its most modern and interesting applications is in the field of Brain Computing Interfaces (BARACHANT et al. 2012), in which the observer is interested in detecting the response of the subject to a particular stimulus and vice-versa. In this field, the use of the properties of the Riemannian manifold of SPD matrices is widely adopted, and has brought numerous important results.

Many applications to Brain Computing Interfaces reveal, however, a weak generalizability of the predictions (i.e. of the results) to other subjects. The similarities among subjects are most often hidden by the individual specific brain patterns. The Procrustes analysis attempts to solve this problem. It aims to match matrices using similarity transformations by minimizing their Frobenius distance. A few seminal papers (GOODALL 1991; DRYDEN et

al. 1997) consolidated the theory, then paving the road to its applications to wide range of fields such as ecology (SAITO et al. 2015), biology (ROHLF and SLICE 1990), analytical chemometrics (ANDRADE et al. 2004), psychometry (GREEN 1952; McCRAE et al. 1996) and neuroscience (HAXBY et al. 2001).

Despite the paramount importance of reproducible and theoretically solid registration tools for SPD data, no gold standard approach is uniquely recognized in the literature. In this context, ZANINI et al. 2018 and BHATIA and CONGEDO 2019 firstly worked in the direction of developing a Procrustes approach into a Riemannian framework. In spite of the admirable results all the above-mentioned methods are problem-specific and often depend on the data collection and on the dependence between variables. Motivated by this and by considerations on Riemannian geometry, we propose a registration method that hypothesize the existence of a latent set of common traits between the subjects, that is hidden through the single subject specific spatial allocations of signals on the brain surface.

Our proposal is based on orthogonal transformations of the data (thus preserving their eigenvalue structure) and on the identification of a suitable reference matrix minimizing a notion of *rotational effort*, which allows us to maintain as much of the original information as possible. In our illustrative example, for instance, this implies that the transformed matrices are still representable in the original brain space.

Before describing our Riemannian Registration (RR) method, in the next section, we recall some notations and well-known concepts in Riemannian geometry in the space of SPD matrices from Chapter 1. Let \mathcal{S}_p^+ be the space of symmetric and positive definite matrices of fixed dimension p . We will in general omit the subscript p unless it is needed. The set \mathcal{S}^+ is a subset of the general linear group and can be seen as a Riemannian manifold when endowed with a suitable metric. Herein, we will use the affine-invariant metric, discussed in Section 1.11.1, that can be represented in terms of the usual Frobenius norm and that we report here for sake of convenience. For

$\Sigma, \Sigma_1, \Sigma_2 \in \mathcal{S}^+$, the affine-invariant distance is defined as

$$d_{\text{AI}}(\Sigma_1, \Sigma_2)^2 = \left\| \Sigma_1^{-1/2} \Sigma_2 \Sigma_1^{-1/2} \right\|_{\text{AI}}^2, \quad (3.1)$$

$$\|\Sigma\|_{\text{AI}}^2 = \|\text{Log } \Sigma\|_{\text{F}}^2 = \sum_{j=1}^p \log^2 \lambda_j(\Sigma), \quad (3.2)$$

where $\lambda_j(\Sigma)$ is the j -th ordered eigenvalue of the matrix Σ . We will often use the notation $\lambda_{ji} = \lambda_j(\Sigma_i)$ to indicate the j -th eigenvalue of the i -th matrix of a set $\{\Sigma_i\}_{i=1}^n$.

Given $\Sigma \in \mathcal{S}^+$ and $\Omega \in \mathbb{O}$ a squared orthogonal matrix of the same dimension, it is easy to observe that $\Omega\Sigma \notin \mathcal{S}^+$. A transformation that preserve the symmetry and positive definiteness of a matrix exists and can be defined as (TUAN et al. 2010):

$$\begin{aligned} T_{\Omega}: \quad \mathcal{S}^+ &\rightarrow \mathcal{S}^+ \\ \Sigma &\mapsto T_{\Omega}(\Sigma) = \Omega\Sigma\Omega^{\text{T}}, \end{aligned} \quad (3.3)$$

which, as $\Omega \in \mathbb{O}$, also preserves the determinant and the eigenvalues of the matrix and as such can be considered as the equivalent of the orthogonal transformations for SPD matrices. Notice that this transformation corresponds to the congruent transformation described in Remark 1.2 in Chapter 1, limited to orthogonal matrices.

The rest of the chapter is organized as follows. In Section 3.1 we introduce the proposed Riemannian registration method. Its practical performance is illustrated in Section 3.2 and Section 3.3, both with synthetic and real world datasets related to the illustrative application on the EEG data nominated before and a functional Magnetic Resonance Imaging (fMRI) dataset discussed in CANALE et al. 2018, respectively.

I RIEMANNIAN REGISTRATION ALGORITHM

Our goal is to introduce matrices Σ_i^* for $i = 1, \dots, n$, through orthogonal rotations $\Sigma_i^* = T_{\Omega_i}(\Sigma_i)$. Our solution is obtained in two steps motivated by two main desiderata:

1. the sum of the distances between the transformed Σ_i^* is minimum
2. the *rotational effort* to pass from Σ_i to Σ_i^* is as small as possible.

In Section 3.1.I we first introduce the notion of *reference matrix* $\tilde{M} \in \mathcal{S}^+$ for a sample of SPD Σ_i , $i = 1, \dots, n$ showing that its distance from each rotated matrix depends only on their eigenvalue structure. Moreover, we use a result in orthogonal Procrustes analysis to obtain a closed form expression for the orthogonal matrices Ω_i that depends only on the original sample. Section 3.1.II specifies the notion of *rotational effort* and how to exploit it to obtain a closed form expressions for reference matrix \tilde{M} .

In Remark 3.1 the relation between the proposed method and the classical orthogonal Procrustes analysis (GOWER and DIJKSTERHUIS 2004) is highlighted.

1.I Reference for a rotated set of SPD matrices

Given a sample of n SPD matrices we introduce the reference matrix $\tilde{M} \in \mathcal{S}^+$, minimizing the following sum of distances

$$\tilde{M} = \operatorname{argmin}_{M \in \mathcal{S}^+} \sum_{i=1}^n d_{AI}(M, \Sigma_i^*). \quad (3.4)$$

where $\Sigma_i^* = T_{\Omega_i}(\Sigma_i)$ is the orthogonal transformation of matrix Σ_i through the generic $\Omega_i \in \mathbb{O}$.

Notably, this problem is reminiscent of the Riemannian center of mass of a set of n matrices (BERGER 2003; MOAKHER 2005), with the difference being that we are interested in the center of mass of the *rotated matrices* Σ_i^* . This

difference looks subtle but will enable a closed form solution that solves the problem without the need of an explicit minimization.

We first search for suitable expressions for each Ω_i as a function of M , defined in Equation (3.4). Said $\underline{\Omega} = \{\Omega_i\}$ the collection of the n orthogonal matrices we are looking for, we define it as the solution of

$$\underline{\Omega} = \underset{W \in \mathbb{O}^n}{\operatorname{argmin}} \sum_{i=1}^n d_{\text{AI}}(M, T_{W_i}(\Sigma_i))^2 \quad (3.5)$$

for fixed M . Each term of this sum is independent from all the others and greater or equal than zero, thus it is sufficient to find n orthogonal matrices Ω_i that depends only on M and on the i -th Σ_i matrix.

Notably, each term of the sum resembles the orthogonal Procrustes problem (GOWER and DIJKSTERHUIS 2004) and indeed a similar solution can be found. This result is given in the following lemma.

LEMMA 3.1 (SYMMETRIC AND POSITIVE DEFINITE PROCRUSTES)

Let $\Theta_1, \Theta_2 \in \mathcal{S}^+$. Then

$$\underset{\Omega \in \mathbb{O}}{\operatorname{argmin}} d_{\text{AI}}(\Theta_1, T_{\Omega}(\Theta_2))^2 = \Gamma_1 \Gamma_2^{\top}. \quad (3.6)$$

where $\Gamma_1 \Lambda_1 \Gamma_1^{\top} = \Theta_1$ and $\Gamma_2 \Lambda_2 \Gamma_2^{\top} = \Theta_2$ are the two Eigenvalue Decomposition (ED) (that exist and are unique due to the spectral theorem).

Proof. This proof was firstly provided in BHATIA and CONGEDO 2019 and is here provided for ease of reference.

First we recall the substitution that, thanks to the cyclic property of the eigenvalues, the identity

$$\|\Sigma_1^{-1/2} \Sigma_2 \Sigma_1^{-1/2}\|_{\text{AI}}^2 = \|\Sigma_1^{-1} \Sigma_2\|_{\text{AI}}^2$$

holds, which allows us to write

$$d(\Sigma_1, \Sigma_2)^2 = \sum_h^p \log^2 \lambda_h(\Sigma_1^{-1} \Sigma_2)$$

Then we use a result on majorization for eigenvalues from Gel'fand, Naimark and Lidskii (theorem III.4.6 in BHATIA 1997, p. 73), which states that

$$\log \lambda^\downarrow(\Sigma_1) + \log \lambda^\uparrow(\Sigma_2) < \log \lambda(\Sigma_1 \Sigma_2) < \log \lambda^\downarrow(\Sigma_1) + \log \lambda^\downarrow(\Sigma_2)$$

or equivalently (if we replace Σ_2 with its inverse):

$$\log \lambda^\downarrow(\Sigma_1) - \log \lambda^\downarrow(\Sigma_2) < \log \lambda(\Sigma_1 \Sigma_2^{-1}) < \log \lambda^\downarrow(\Sigma_1) - \log \lambda^\uparrow(\Sigma_2). \quad (3.7)$$

In this context we use the standard notation for majorizations (see for instance MARSHALL et al. 2010) which, given $x = (x_1, \dots, x_n) \in \mathbb{R}^n$, defines x^\uparrow (resp. x^\downarrow) to be the vector obtained by arranging the components of x in increasing (resp. decreasing) order and defines $x < y$ (x is *majorised* by y) if

$$\sum_{i=1}^k x_i^\downarrow \leq \sum_{i=1}^k y_i^\downarrow \quad \text{for } k < n \quad \text{and} \quad \sum_{i=1}^n x_i = \sum_{i=1}^n y_i.$$

Majorizations (3.7) hold also for any gauge function of the sequence of the logarithms of the eigenvalues (such as the sum of squares of the elements, see BHATIA 1997, § IV on the general definition of *gauge function*) and thus they imply that

$$d(\Lambda_1^\downarrow, \Lambda_2^\downarrow)^2 \leq d(\Sigma_1, \Sigma_2)^2 \leq d(\Lambda_1^\downarrow, \Lambda_2^\uparrow)^2$$

Now it is sufficient to observe that the left and right hand sides of this inequality do not change if we replace Σ_2 with $\Omega \Sigma_2 \Omega^\top$, which means that to obtain the minimum of $d(\Sigma_1, \Omega \Sigma_2 \Omega^\top)$ we need that

$$\lambda(\Gamma_1^{-\top} \Lambda_1^{-1} \Gamma_1^{-1} \Omega \Gamma_2 \Lambda_2 \Gamma_2^\top \Omega^\top) = \lambda(\Lambda_1^{-1} \Lambda_2)$$

which implies (using the cyclic property of the eigenvalues)

$$\Omega = \Gamma_1 \Gamma_2^\top$$

as we wanted to show. \square

Using the result of Lemma 3.1 and defining $\Omega_i = \Gamma_M \Gamma_{\Sigma_i}^\top$, we go back to the problem in (3.4), noting that now we can express each of its terms as

$$\begin{aligned} d_{AI}(M, \Sigma_i^*)^2 &= \sum_{j=1}^P \log^2 \lambda_j (M^{-1/2} \Sigma_i^* M^{-1/2}) \\ &= \sum_i^K \left[\sum_h^P \log^2 \lambda_h (M^{-1} \Gamma_M \Gamma_i^\top \Sigma_i \Gamma_i \Gamma_M^\top) \right] \\ &= \sum_{j=1}^P \log^2 \frac{\lambda_{ji}}{\lambda_j(M)}. \end{aligned} \quad (3.8)$$

where $\lambda_{ji} = \lambda_j(\Sigma_i) = \lambda_j(\Sigma_i^*)$ due to the properties of the orthogonal-like transformation we defined in Equation (3.3). From Equation (3.8), it is clear that each distance in Equation (3.4) depends only on the eigenvalues of M and Σ_i . Consistent with this, if we consider the polar decomposition of a SPD matrix as

$$\mathcal{S}^+ \ni \Sigma = \Omega \text{diag}(e^r) \Omega^\top, \quad (3.9)$$

the problem in Equation (3.4) can be seen as pertaining only the *radial* component $r \in \mathbb{R}^P$ and not to the *angular* component $\Omega \in \mathbb{O}$.

Thus, we focus on the set of diagonal matrices with positive entries $\mathcal{D}^+ \subset \mathcal{S}^+$. Each point in \mathcal{D}^+ is associated to an infinite collection of matrices in \mathcal{S}^+ with the same set of eigenvalues. In our context this implies that

1. the distance after the registration procedure between any two elements of \mathcal{S}^+ is equal to the one between the corresponding representatives in \mathcal{D}^+ obtained after diagonalization, consistently with Equation (3.8);

2. for two representative matrices in \mathcal{D}^+ , the Riemannian distance reduces to the Euclidean distance between the radial components of the polar decomposition in Equation (3.9).

Consistent with implication 1, we recast the minimization of Equation (3.4) as

$$\tilde{\Lambda} = \operatorname{argmin}_{\Lambda \in \mathcal{D}^+} \sum_{i=1}^n d_{\text{AI}}(\Lambda, \Lambda_{\Sigma_i}).$$

Then, exploiting the decomposition in Equation (3.9) and consistently with implication 2, the solution to the previous equation is a diagonal matrix with entries that satisfy

$$\operatorname{argmin}_{r \in \mathbb{R}^p} \sum_{i=1}^n (r_i - r)^2, \quad (3.10)$$

where $r_i \in \mathbb{R}^p$, $r_i = (r_{i1}, \dots, r_{ip})^\top$, and $r_{ij} = \log \lambda_j(\Sigma_i)$. Clearly, the minimization in (3.10) has a simple solution and it leads, for $j = 1, \dots, p$ to

$$\lambda_j = \left[\prod_{i=1}^n \lambda_j(\Sigma_i) \right]^{\frac{1}{n}} = \exp \left[\frac{1}{n} \sum_i r_{ij} \right]. \quad (3.11)$$

Now, let $\tilde{\Lambda}$ be the diagonal matrix containing the eigenvalues obtained with Equation (3.11). Since $\mathcal{D}^+ \subset \mathcal{S}^+$, $\tilde{\Lambda}$ represents just one, yet elegant and computationally undemanding, of the infinite solutions to the problem in Equation (3.4). Indeed, there are infinite matrices in \mathcal{S}^+ obtained as $\Gamma \tilde{\Lambda} \Gamma^\top$ for $\Gamma \in \mathbb{O}$ leading to the same sum of distances.

In the following section we introduce a notion of *rotational effort* and use it to obtain a single element \tilde{M} of the form $\Gamma \tilde{\Lambda} \Gamma^\top$ lying in the full manifold \mathcal{S}^+ and, as we have seen, solving Equation (3.4).

1.11 Constraining the eigenvector matrix

We introduce the following definition of *rotational effort* R :

DEFINITION 3.1: Given a set of rotations $\underline{\Omega} = \{\Omega_i\}_{i=1}^n$, the rotational effort $R_{\underline{\Omega}}$ is the sum of the effective rotations, measured as the Frobenius distance from the identity element of the orthogonal group

$$R_{\underline{\Omega}} = \sum_{i=1}^n d_F(\Omega_i, \mathbb{1})^2,$$

where $\mathbb{1}$ is the diagonal unitary matrix.

A similar definition could, in principle, be expressed relative to any matrix $O \in \mathbb{O}$. The results obtained in the following would not change much in this more general case, but for the sake of exposition we maintain the definition with respect to the diagonal unitary matrix.

Consistently with Definition 3.1, it is reasonable to look for a matrix $\Gamma \in \mathbb{O}$ minimizing the rotational effort induced by each $\Omega_i(\Gamma) = \Gamma \Gamma_{\Sigma_i}^\top$ arising from the application of Lemma 3.1 with Equation (3.11), i.e.

$$\tilde{\Gamma} = \underset{G \in \mathbb{O}}{\operatorname{argmin}} R_{\underline{\Omega}(G)} \quad (3.12)$$

The intuition behind this choice is appealing: we are imposing that the rotation performed on each data point Σ_i is as small as possible, provided that we still obtain the appropriate alignment. This is particularly interesting when the graphical representation of the matrix data is still interesting: indeed, by imposing this constrain we are asking to be as close as possible to the original space, allowing us to preserve some notion of spatial distribution of the variables, for instance when dealing with EEG data as we will show in Section 3.III.

Notably, Equation (3.12) is a particular version of a generalized Procrustes problem, similar to the ones discussed in Section 9 of GOWER and DIJKSTER-

HUIS 2004. A few algebraic manipulations lead to

$$\begin{aligned}
\tilde{\Gamma} &= \operatorname{argmin}_{G \in \mathbb{O}} \sum_i^K d_F(G\Gamma_i^\top - \mathbb{I})^2 = \operatorname{argmin}_{G \in \mathbb{O}} \sum_i^K \langle G\Gamma_i^\top - \mathbb{I}, G\Gamma_i^\top - \mathbb{I} \rangle \\
&= \operatorname{argmin}_{G \in \mathbb{O}} \sum_i^K \left[\|G\Gamma_i^\top\|_F^2 + \|\mathbb{I}\|_F^2 - 2 \langle G\Gamma_i^\top, \mathbb{I} \rangle \right] \\
&= \operatorname{argmax}_{G \in \mathbb{O}} \sum_i^K \langle G\Gamma_i^\top, \mathbb{I} \rangle = \operatorname{argmax}_{G \in \mathbb{O}} \sum_i^K \langle G, \Gamma_i \rangle \\
&= \operatorname{argmax}_{G \in \mathbb{O}} \sum_i^K \operatorname{Tr}[G^\top \Gamma_i] = \operatorname{argmax}_{G \in \mathbb{O}} \operatorname{Tr} \left[G^\top \left(\sum_i^K \Gamma_i \right) \right] \\
&= \operatorname{argmax}_{G \in \mathbb{O}} \langle G, \sum_i^K \Gamma_i \rangle = UV^\top
\end{aligned} \tag{3.13}$$

where U and V are obtained from the Singular Value Decomposition (SVD) $\sum_{i=1}^n \Gamma_i = UDV^\top$.

The reference matrix \tilde{M} and thus the optimal registration providing the minimal rotational effort and minimal sum of distances between the registered matrices are then identified by

$$\tilde{M} = \tilde{\Gamma} \tilde{\Lambda} \tilde{\Gamma}^\top, \quad \tilde{\lambda}_j = \exp \left[\frac{1}{n} \sum_i^n \log \lambda_{ji} \right], \quad \tilde{\Gamma} = UV^\top \tag{3.14}$$

where λ_{ji} is the j -th eigenvalue for matrix Σ_i and U and V are obtained from the SVD of $\sum_{i=1}^n \Gamma_{\Sigma_i}$. The rotated matrices then are simply

$$\Sigma_i^* = T_{\tilde{\Gamma}\Gamma_{\Sigma_i}^\top}(\Sigma_i) = \tilde{\Gamma} \Lambda_i \tilde{\Gamma}^\top, \quad i = 1, \dots, n. \tag{3.15}$$

REMARK 3.1: *It might be interesting to observe the similarity of the problem in Equation (3.4) to one of the form of the generalized Procrustes problem see identity 9.2 in GOWER and DIJKSTERHUIS 2004. Indeed, in that context the sum of the pairwise distances is equal to the sum of the distances from the barycenter, while in the context of SPD matrices this equivalence does not hold anymore due to the different*

properties of the metric. In fact, to prove that identity one has to rely on the distribution property of the trace and of the linear operators that appear in the Frobenius norm, but the metric of Equation (3.2) features a non linear combinations of the two matrices (namely, $\text{Log}(A^{-1/2}BA^{-1/2})$) and the distribution property does not hold anymore.

II SIMULATION STUDY

In the context of the SPD matrices space, we consider the unbiased centered sample variance as a measure of dispersion of a sample $\{\Sigma_i\}_{i=1}^n, \Sigma_i \in \mathcal{S}^+$ defined as

$$s^2 = \frac{1}{n-1} \sum_i^n d^2(\hat{\Sigma}, \Sigma_i) \quad (3.16)$$

where $\hat{\Sigma}$ is an estimator for the center of the sample, such as the Karcher barycenter (which is indeed defined as the $\hat{\Sigma}$ that minimize s^2).

It is easy to observe that, by virtue of how we set up Equations (3.4) and (3.5), the sample variance of the set of registered matrices centered with respect to the optimal reference \tilde{M} defined by Equation (3.14) will always be smaller or equal than the sample variance of the original distribution, as it holds that

$$\sum_i d^2(\tilde{M}, \Sigma_i^*) \leq \sum_i d^2(\hat{\Sigma}, \Sigma_i),$$

with equality holding only if all the original matrices are already diagonal. Moreover, as the expression to compute \tilde{M} are in closed form, we expect the computational time required to perform the operations on the left hand side to be orders of magnitude shorter than the one required to obtain the karcher estimate of the barycenter. We call s_*^2 the equivalent of Equation (3.16) when using the registered set of matrices Σ_i^* and the reference matrix \tilde{M} as the center.

Indeed, we show this behavior through a simulation, which results are shown in Figures 3.1 and 3.2. First, we select a matrix dimension p and a sample size n , while we vary both, we will show the results with a fixed n as the sample size has a predictable effect on the results: the time required increase with the sample size, while the deviation s^2 remains approximately constant. Notice that the tested values of n ranges between 20 and 500, with no appreciable differences. We explore the performance of our algorithm for matrix dimensions ranging from 5 to 50, as for larger matrix the computational time required for the computation of the Karcher barycenter becomes too large.

To show the properties of our method we then randomly generate a true *center* for the distribution by sampling uniformly $U_C \in \mathbb{O}_p$ and $r_C \in [-5, 5]^p$ and then computing

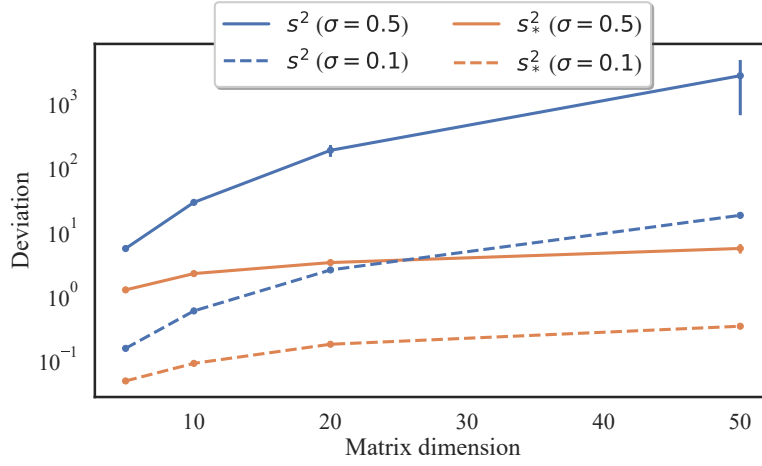
$$C = U_C \text{diag}(e^{r_C}) U_C^T.$$

From this, we generate a sample of n SPD matrices centered around C according to two distributions, namely the Wishart distribution (WISHART 1928) and the Riemannian Gaussian distribution (SAID et al. 2017):

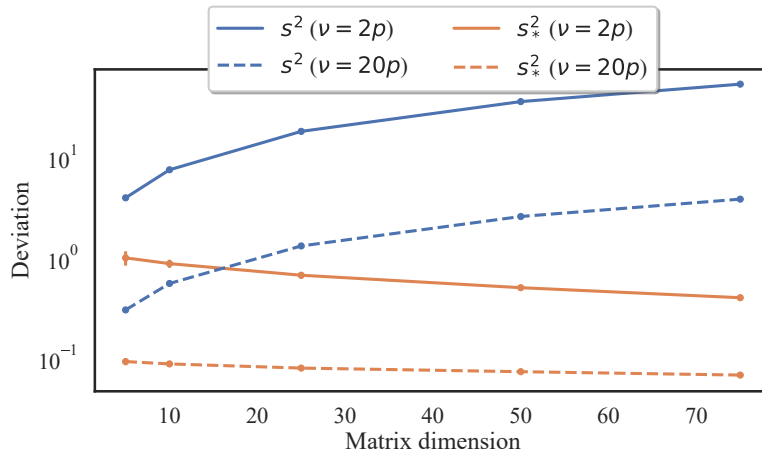
- For the Wishart distribution we fix the degree of freedom as $\nu = 2p$ (resp. $\nu = 20p$) to obtain a large (resp. small) variance in the simulated sample. Then we generate the sample from $\mathcal{W}_p(C/\nu, \nu)$.
- For the Riemannian Gaussian distribution we fix $\sigma = 0.5$ (resp. $\sigma = 0.1$) to obtain a large (resp. small) variance in the simulated sample. Then we generate the sample from $\mathcal{RG}(C, \sigma)$.

For each combination of parameters we repeat the experiment $N_{\text{rep}} = 100$ times, and for each simulation we compute the deviation s^2 from equation (3.16) (which requires finding the Karcher center of mass $\hat{\Sigma}$) and then $s_*^2 = \frac{1}{n-1} \sum_i d^2(\tilde{M}, \Sigma_i^*)$

The results for s^2 and s_*^2 are shown in Figure 3.1. It is interesting to observe that, as expected, the measured deviation after our rotation is always



(A) Riemannian Gaussian distributed



(B) Wishart distributed

FIGURE 3.1: The results of the simulation described in Section 3.11 for $n = 500$. On the upper panel, results for matrices sampled from a Riemannian Gaussian distribution are shown, while on the lower panel are shown results for Wishart-distributed matrices. For each plot the blue lines show the dispersion as a function of the matrix dimension for the original data, while the orange lines show the dispersion after the Riemannian registration algorithm discussed in this work. Moreover, the solid and dashed lines distinguish between larger and smaller variance in the initial distribution of the data.

smaller than the original one but also, perhaps more surprisingly, that in the Wishart case, it seems to not increase with the matrix dimension (even shrinking slightly). As we have seen previously, since we can express the distance of the registered matrices with respect to the optimal reference M as the Euclidean distance in the space of the eigenvalues, the deviation in the rotated space can be written as (using the expression in Equation (3.11) for the eigenvalues of \tilde{M})

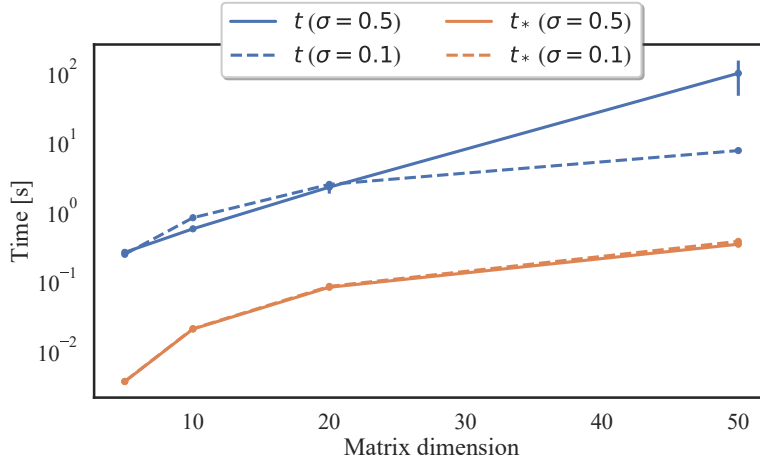
$$\frac{1}{n-1} \sum_i^n d^2(\tilde{M}, \Sigma_i^*) = \frac{1}{n-1} \sum_i^n \sum_j^p \left(r_{ji} - \frac{1}{n} \sum_k^n r_{jk} \right)^2 = \sum_j^p s_j^2$$

where s_j^2 is the unbiased sample variance of the logarithm of the eigenvalues. The fact that this term is decreasing, then, is a known consequence of the distribution of the eigenvalues of a Wishart distributed matrix (see ZANELLA et al. 2009).

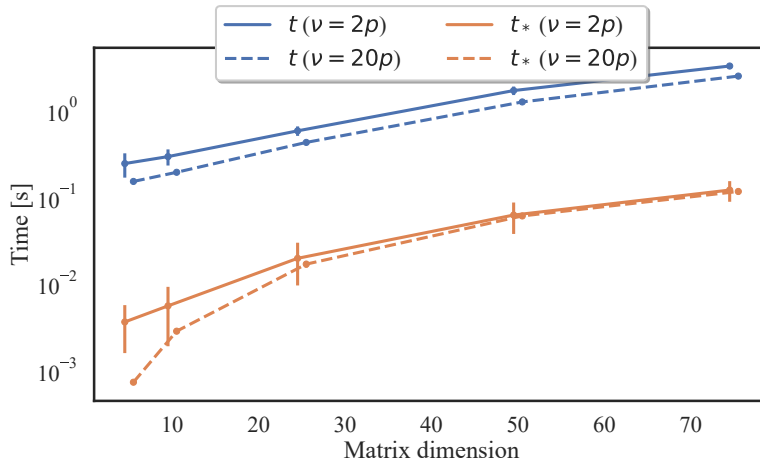
In Figure 3.2, instead, the time required for a single run of the simulation, *i.e.* the time, given the sample, to find the Karcher center of mass (resp. the reference matrix) and compute the dispersion s^2 (resp. s_r^2) is shown. A difference in the time required of around 2 order of magnitude is always present. Moreover, notice that the time required for our algorithm does not depend on the dispersion of the original sample, which for the Riemannian Gaussian distributed data is indeed the case when the matrix dimension is large.

III APPLICATIONS TO MEDICAL IMAGING DATA

We now show two applications to real data of our method. In both cases the applications stems from neuroscience and the study of the brain and are to be considered illustrative of the potential of the method. In the first case we use resting state EEG data to show that the proposed RR method can lead to an improvement in the interpretation of the connectivity network. In the



(A) Riemannian Gaussian distributed



(B) Wishart distributed

FIGURE 3.2: The times (in seconds) required for a single run of the simulation described in Section 3.11 for $n = 500$. As in Figure 3.1 the blue lines relate to the original data, while the orange ones to the data after the Riemannian registration discussed in this work, while solid and dashed lines distinguish between larger and smaller variance in the initial distribution of the data.

second dataset we use *fMRI* data to show how this method enhances – and simplifies – the inferential approach and prediction.

III.1 EEG data

First, we analyse the *EEG* data provided by CATTAN et al. 2018¹, a dataset collected at the GIPSA-lab in 2017 and containing electroencephalographic recordings of 20 subjects in a resting-state with eyes open and closed.

After a standard pre-processing of the data, following the authors' guidelines, in which we remove the noisiest frequency components with a band-pass filter between 3 and 40 Hz, we center the data and compute the covariance matrix of the signals for each subject. Then, we perform a qualitative exploration of the data, which shows that our *RR* procedure allows the underlying processes of the data to better emerge, removing much of the single subject variability due, in the example of this application, to the specific positioning of the electrodes on the scalp. As can be seen directly from drawing the correlation matrices (see for instance, three different subjects in Figure 3.3), an underlying structure is clearly emerging, even if the subjects still maintains their individuality.

This effect is even more apparent in Figure 3.4, where the Karcher barycenter and the reference matrix are shown. Indeed in the reference matrix, shown in the right hand panel, the same pattern that was emerging in the single subject matrices is clear, while the Karcher barycenter of the original data provides no indication of an underlying latent structure: all the entries in the matrix are pretty similar to each other and close to zero.

III.2 fMRI data

For a second application we used the data from the enhanced Nathan Kline Institute-Rockland Sample (*NKI1*) described in NOONER et al. 2012². In this

¹Available at <https://doi.org/10.5281/zenodo.2348892>

²Available at http://fcon_1000.projects.nitrc.org/indi/CoRR/html/nki_1.html.

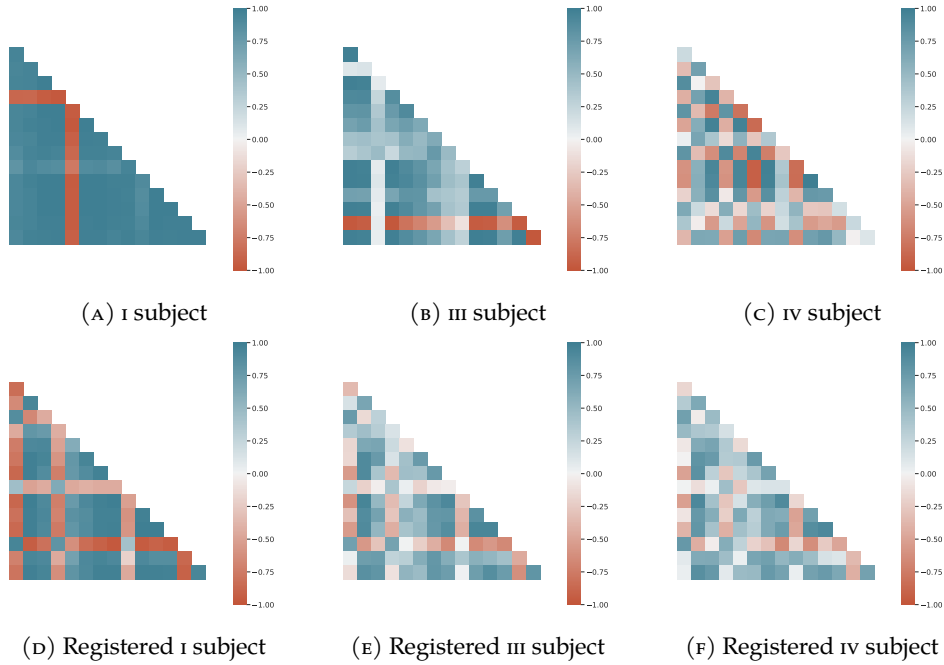


FIGURE 3.3: The empirical correlation matrices of three subjects from the EEG dataset CATTAN et al. 2018 before and after the Riemannian registration process described in this work. A darker color means a value closer to ± 1 (resp. +1 blue and -1 red).

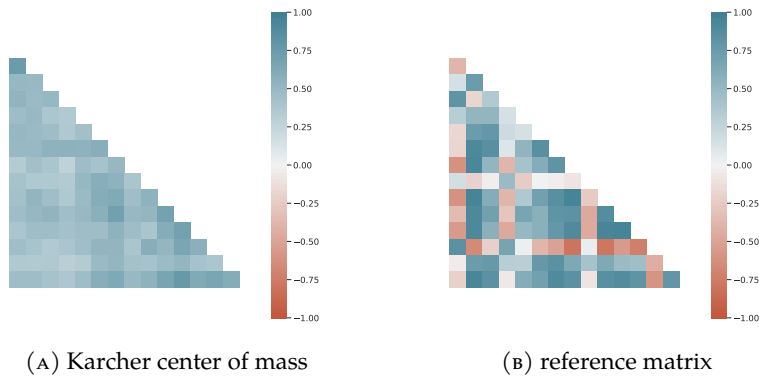


FIGURE 3.4: The Karcher center of mass and the reference matrix \tilde{M} for the subjects from CATTAN et al. 2018. Darker shade of blue (resp. red) means a value closer to the +1 (resp. -1).

study, 18 subjects are divided in two group by the presence of a clinical diagnosis, and data on their functional network are provided by means of Resting state fMRI (R-fMRI). The available diagnosis ranges from alcoholism to severe schizophrenic disorder, but given the limited amount of data we focused on the binary outcome of having a diagnosis against the control group of healthy subjects.

The two groups are compared on the basis of their connectivity measured as the empirical variance-covariance matrix, built from the 70 brain parcellation based on the Desikan Atlas (DESIKAN et al. 2006). Notice that 8 out of the 18 subjects have multiple scans available, thus allowing both inference and validation on the same subject, but as we focus on an illustrative example of inference we used only a single scan per subject (see the introduction to CANALE et al. 2018, for details on the dataset structure).

First we perform a permutation test PESARIN 2001 to check the group differences for the original data and for the matrices after the proposed Riemannian registration algorithm. The permutations (obtained by $k = 1000$ random permutations of the group membership for the subject and then computing the distance between the center of the groups) show that the groups are significantly apart in both scenarios, with observed p -values of 0.032 before the Riemannian registration and 0.004 after it.

As previously discussed, the proposed Riemannian Registration method highlights the importance of the latent eigenvalues structure of the data. Consistently with this, we focus the analysis on the eigenvalues set of each matrix. An example of such analysis is the test for the statistical significance of the eigenvalues in predicting the membership of the subject to any of the two groups. Such tests can then be performed within standard frameworks for multiple t-testing, as the one discussed in PESARIN 2001 (*i.e.* min-p multiple testing correction, described in WESTFALL and YOUNG 1993). In Figures 3.5 and 3.6 the results of such analysis are shown, where the significance t-tests implemented for the R software (R CORE TEAM 2021) in the package `flip` (FINOS 2018) have been used.

A classical task for this kind of data is of course to be able to perform

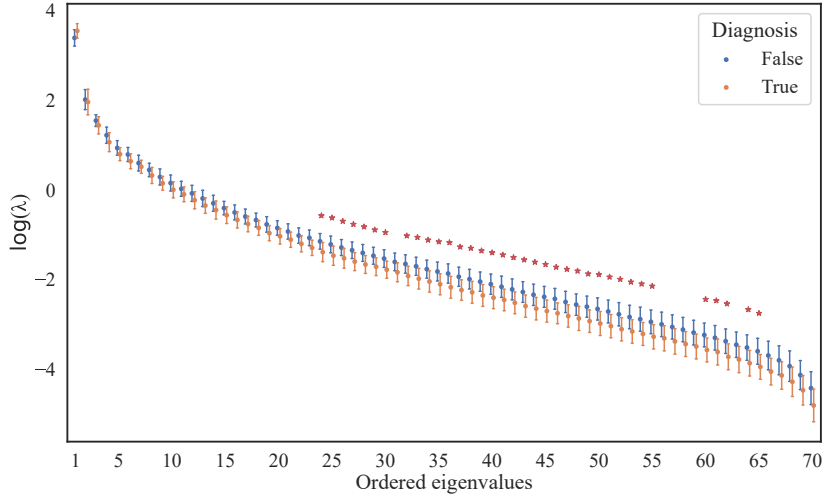


FIGURE 3.5: Distribution of the eigenvalues for the two groups, shown are the mean eigenvalue for the group and the bars represents one standard deviation. A small red star indicates the significant eigenvalues, as described in Figure 3.6.

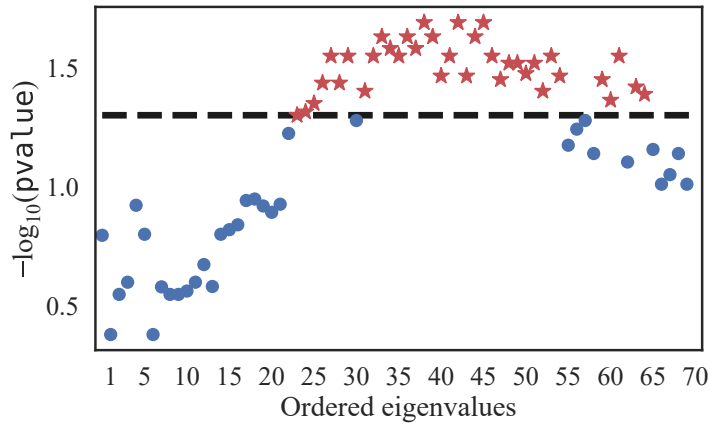


FIGURE 3.6: Significance (represented via the usual $-\log_{10}(p)$ transformation) of each eigenvalue in predicting the diagnosis label obtained via a permutation test. The red stars shows the significant eigenvalues, *i.e.* the eigenvalues for which the p-value of the permutation test is lower or equal then 0.05.

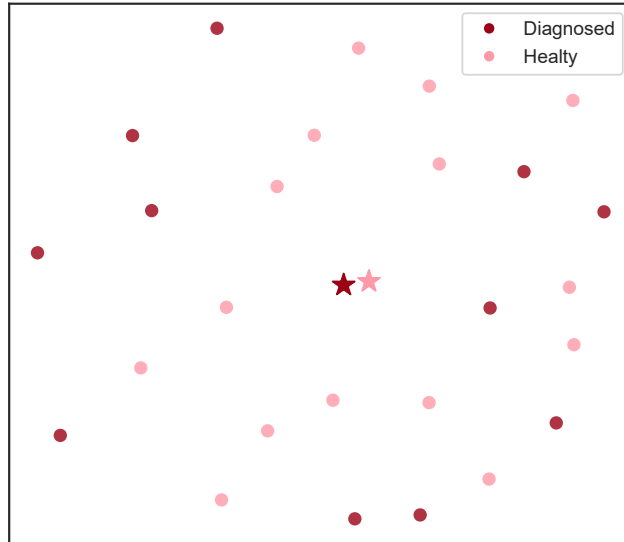
classification, and standard methods in the literature are available for such a scenario that exploits the Riemannian structure, such as the Minimum Distance to Riemannian Mean (MDRM) by BARACHANT et al. 2012, which classifies each subject according to its Riemannian distance from the Karcher center of mass of the group.

Our RR method, then, impacts this classification algorithm by performing a registration of each cluster with respect to its reference matrix first and then classifying a new subject by registering it with respect to each reference matrix and comparing the two distances of the registered subject to its reference.

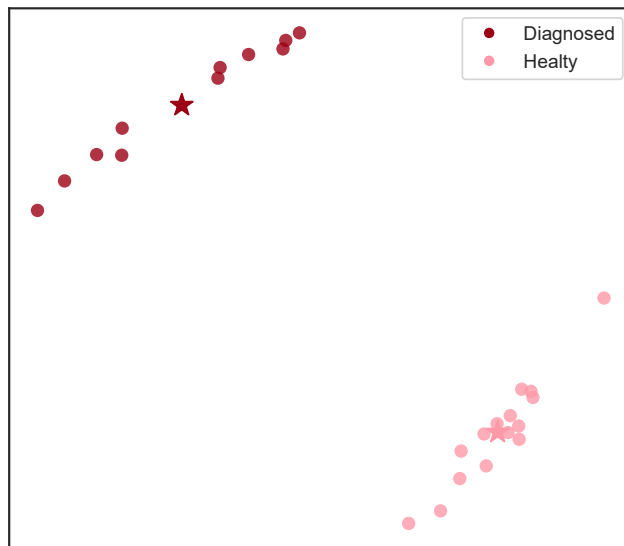
A fast comparison of the two methods on the NKI₁ dataset does not show significant differences in performances when applied to such a dataset due to its dimensions, with both methods obtaining the same 89% accuracy on train and 62.5% accuracy on test (where, as discussed before, we used the second scan for the subjects that have it available to test the classification). By pooling the two sets together and performing a leave-one-out classification procedure, we obtain an estimate on the accuracy of 79% for our algorithm and 74% for the standard MDRM.

Moreover, we tested the two algorithms also for the EEG dataset of Section 3.III.I, again with a leave-one-out procedure, obtaining an accuracy of 71% with our method and 57% (slightly above random chance) for the MDRM algorithm.

Figure 3.7 shows the two groups before and after the RR procedure. To obtain a 2 dimensional projection, we used the classical metric Multi Dimensional Scaling (MDS) algorithm (BORG and GROENEN 2005) in the implementation of Scikit-Learn (PEDREGOSA et al. 2011). This low-dimensional representation of the data is built in order to have the distances respect *well* the distances in the original high-dimensional space, and the metric variant does so by setting the distances between output two points to be as close as possible to the similarity or dissimilarity in the original data. As it can be seen, even for such a small dataset, the RR procedure provides a substantial increase in the shrinking and separation of the two groups.



(A) Original matrices



(B) Registered matrices

FIGURE 3.7: A 2 dimensional MDS projection of the NKI₁ dataset, where the colors distinguish between healthy and diagnosed subjects. On the left panel, the projected data are the original covariance matrices while on the right one are shown the projection of the registered matrices after the RR algorithm.

4 | DISCUSSION

The aim of this work has been to introduce different techniques for dealing with non Euclidean data in statistics, specifically with data lying in the Symmetric and Positive-Definite manifold. In particular, we dealt with two examples of the kind of situations that may arise in this scenarios: first, in an inference scenario, the presence of a variance-covariance parameter is common and, in those cases, it is often required to perform the maximization of a function (generally the log-likelihood or the log-posterior of the model) over that parameter. As a second example, we have considered the case in which the Symmetric and Positive-Definite (*SPD*) manifold is actually the sample space: this situation appears frequently in many modern domains, such as neuro-imaging or computer vision. As with most scenarios of this kind, a pre-processing step is often required to ensure that the inference or classification algorithms are allowed to work in the optimal way. We will now proceed to summarize briefly the main contributions of this work, along with several line of future development that we deem interesting and potentially open.

First, in Chapter 1 we provided a compendium of the main ideas and results needed to use differential geometry within the context of statistical applications, starting with a general description of the concepts of Riemannian manifold, metric and geodesics and then detailing those concepts in the case of *SPD* matrices. It is interesting to observe that already in this ere is room for extending this work, as the choice of the Affine-Invariant (*AI*) metric is entirely arbitrary (although the *AI* metric has some properties that makes

it a very good choice, see the discussion in MOAKHER 2005; PENNEC et al. 2006). This choice can be changed, though, and indeed recently there have been some works trying to define new metrics that might show different behaviors in different situations (see for instance LIN 2019). At this moment this new proposals are still in an initial phase, but they sure seem worth exploring in the future. Changing the choice of the metric clearly modifies all the subsequent quantities, such as the expression for the geodesic and the parallel transport, then requiring to rethink some of the details of the other parts of this work as they are build upon this properties.

Chapter 2 is devoted to the development of a Riemannian optimization method tuned to the specific properties of the SPD manifold. In Section 2.1 we have explained how to construct a quasi-Newton optimization algorithm (we have used the Limited memory BFGS (L-BFGS) class of algorithms, as discussed in Appendix A), discussing the use of approximate geodesic and parallel transport expressions. We have shown via simulations that this approach provides a substantial improvement over the more traditional technique of transforming the SPD matrix through Cholesky decomposition in order to perform a standard Euclidean optimization in terms of the number of iterations required to reach convergence. It is interesting to observe in this context the similarity of this traditional technique with the metric by LIN 2019 cited earlier, and this difference in performance is one of the reasons why we think that a more thorough discussion of that idea might be interesting.

In Chapter 3 a Riemannian Registration (RR) algorithm has been developed and tested, showing its similarities in principle with the Orthogonal Procrustes analysis. As we have shown in Section 3.iii, this method allows to maintain some of the spatial relations thanks to the variational principle used in defining the *rotational effort*, and at the same time provides a framework to perform eigenvalues-based analysis of the matrices. This has the double advantage of allowing the usual Euclidean suite of statistical methods and techniques to perform inference while maintaining a high interpretability of the data. Notice again that also in this context the use of

a different metric would substantially change the results obtained in Section 3.1.1, namely the analytical expressions of Equations (3.14) and (3.15) that depends heavily on the metric expression. Interestingly, though, one of the more appealing property of the Log-Cholesky metric (LIN 2019) is the possibility of obtaining a closed form expression for the Riemannian center of mass, and as such this metric could probably still provide some of the advantage of the AI one. Nevertheless, the interesting property of allowing to focus on the eigenvalues for the statistical analysis would be definitely lost, providing a worse experience in general. Moreover, integrating our procedure within other similar procedures (such as, for instance, the one of ZANINI et al. 2018) is something worth investigating for the increase in the classification performances shown by our method.

A

QUASI-NEWTON OPTIMIZATION METHODS: IMPLEMENTATION DETAILS

In this appendix we will discuss some technical details on the quasi-Newton methods discussed in Section 2.1.1. The code has been implemented in the Python within the JAX framework (BRADBURY et al. 2018) to allow for automatic differentiation for the computations of gradients. The code is available (still in alpha version) as the Python package `optispd`¹. Provided within the package is also the code to reproduce some of the simulations of Chapter 2.

I STRONG WOLFE LINE-SEARCH PROCEDURE

We start discussing the line-search procedure from algorithm 3.5 and 3.6 of NOCEDAL and WRIGHT 2006. This algorithm is used to ensure stability during the phase of line search, all the while increasing the convergence speed of the algorithm.

To find a step length η that satisfies the strong Wolfe conditions of Equation (2.2), this algorithm (shown in Algorithm A.1) starts from a trial estimate η_1 , which is then increased repeatedly until an acceptable step length or an interval containing it is found. In the latter case a sub-procedure of *zoom*, described in Algorithm A.2, is called to repeatedly reduce the interval that contain the step length satisfying the strong Wolfe conditions. An

¹At the moment available only on Github at github.com/jschiavon/optispd.

in-depth discussion of the details of this algorithm can be found by the interested reader in § 3.5 NOCEDAL and WRIGHT 2006.

ALGORITHM A.1: Line search algorithm based on algorithm 3.5 of NOCEDAL and WRIGHT 2006

```

1: function LINESEARCH( $\phi, \phi'$ )
2:    $\eta_0 = 0; \eta_{\max} = 1; m = 2$ 
3:    $\eta_1 \in (\eta_0, \eta_{\max})$ 
4:   for  $i = 1, \dots$  do
5:     Evaluate  $\phi(\eta_i)$ 
6:     if  $\phi(\eta_i) > \phi(0) + c_1 \eta_i \phi'(0)$  then
7:        $\eta_* \leftarrow \text{ZOOM}(\eta_{i-1}, \eta_i);$  Break
8:     Evaluate  $\phi'(\eta_i)$ 
9:     if  $|\phi'(\eta_i)| \leq -c_2 \phi'(0)$  then
10:       $\eta_* \leftarrow \eta_i;$  Break
11:     if  $\phi'(\eta_i) \geq 0$  then
12:       $\eta_* \leftarrow \text{ZOOM}(\eta_i, \eta_{i-1});$  Break
13:      $\eta_{i+1} \leftarrow m\eta_i$ 
14:   return  $\eta_*$ 

```

Here we will just note the definition of ϕ and its derivative ϕ' . Indeed, in the original (euclidean) version of the algorithm $\phi(\eta) = f(x_k + \eta d_k)$ is the value assumed by the function f computed in the evolved point, as a function of the step length used for the evolution. This definition remains the same when dealing with a Riemannian manifold, but in this case the evolution is the retraction that, for the Symmetric and Positive-Definite (SPD) manifold, is described by the exponential map of Equation (2.3).

Moreover, as can be easily seen with a little bit of calculus, the value of $\phi'(\eta)$ is the projection of the descent direction along the direction of the gradient, which is simply the inner product of the gradient with the descent direction, and which is adapted within the context of the SPD manifold as Equation (2.4).

ALGORITHM A.2: Zoom phase of the line search algorithm, based on algorithm 3.6 of NOCEDAL and WRIGHT 2006

```

1: function ZOOM( $\eta_{\text{low}}, \eta_{\text{high}}$ )
2:   for  $i = 1, \dots$  do
3:      $\eta_j \leftarrow \text{INTERPOLATE}(\eta_{\text{low}}, \eta_{\text{high}})$ 
4:     Evaluate  $\phi(\eta_j)$ 
5:     if  $\phi(\eta_j) > \phi(0) + c_1 \eta_j \phi'(0)$  or  $\phi(\eta_j) \geq \phi(\eta_{\text{low}})$  then
6:        $\eta_{\text{high}} \leftarrow \eta_j$ 
7:     else
8:       Evaluate  $\phi'(\eta_j)$ 
9:       if  $|\phi'(\eta_j)| \leq -c_2 \phi'(0)$  then
10:         $\eta_* \leftarrow \eta_j$ ; Break
11:       if  $\phi'(\eta_j)(\eta_{\text{high}} - \eta_{\text{low}}) \geq 0$  then
12:         $\eta_{\text{high}} \leftarrow \eta_{\text{low}}$ 
13:         $\eta_{\text{low}} \leftarrow \eta_j$ 
14:   return  $\eta_*$ 

```

II LIMITED MEMORY BFGS ALGORITHM

In this section we will discuss the implementation of the limited memory version of the Broyden Fletcher Goldfarb Shanno (BFGS) algorithm when dealing with a Riemannian optimization setting. A comprehensive discussion of the original Limited memory BFGS (L-BFGS) algorithm can be found in § 7.2 NOCEDAL and WRIGHT 2006, but in synthesis this example of quasi-Newton algorithm approximates the inverse hessian matrix using only the information on the curvature of the last m iterations, as the information from iterations far in the past is more likely to be less impacting on the current value of the descent direction. This is a significant difference with the BFGS method, as in that case a full, albeit approximated, hessian matrix has to be computed, stored and updated for every iteration.

Specifically, the L-BFGS algorithm stores a modified version of the hessian implicitly by storing a certain number m (the so called memory parameter of the algorithm) of vector pairs $\{s_k, y_k\}$ and a scalar quantity γ_k , defined

for the euclidean setting as

$$s_k = x_{k+1} - x_k \quad y_k = \nabla f_{k+1} - \nabla f_k \quad \rho_k = \frac{1}{\langle s_k, y_k \rangle}.$$

Algorithm A.3 shows the use of this quantities to update the descent direction starting from the gradient ∇f_k . Notice Line 9, where a multiplication by a factor γ_k appears: this is a *per-step* approximation of the initial hessian, as explained in NOCEDAL and WRIGHT 2006, and is computed as

$$\gamma_k = \langle s_k, y_k \rangle / \|y_k\|^2.$$

ALGORITHM A.3: Two-loop recursion to approximate the inverse hessian correction to the gradient in the L-BFGS optimization scheme, based on algorithm 7.4 of NOCEDAL and WRIGHT 2006

```

1: function HESSIANGRADIENT( $\rho, s, y, \gamma_k$ )
2:    $q \leftarrow \nabla f_k$ 
3:   if  $k = 1$  then return  $-q$ 
4:   else
5:      $l \leftarrow \min(m, k)$ 
6:     for  $i = k - 1, k - 2, \dots, k - l$  do
7:        $\alpha_i \leftarrow \rho_i \langle s_i, q \rangle_{x_k}$ 
8:        $q \leftarrow q - \alpha_i y_i$ 
9:      $r \leftarrow \gamma_k q$ 
10:    for  $i = k - l, k - l + 1, \dots, k - 1$  do
11:       $\beta \leftarrow \rho_i \langle y_i, r \rangle_{x_k}$ 
12:       $r \leftarrow r + (\alpha_i - \beta) s_i$ 
13:    return  $-r$ 

```

To compute these quantities in the Riemannian setting, though, we need to apply some careful consideration: indeed, as discussed in Chapter 1 and specifically in Table 1.1, the expression for the difference of two elements on the manifold should be expressed through the logarithm map, and thus $s_k = \log_{x_k}(x_{k+1})$. Moreover, we cannot compare vectors belonging to different vector space, as already said, thus the we need to parallel transport to

ALGORITHM A.4: Update history for Riemannian L-BFGS. Differently from algorithm 7.5 of NOCEDAL and WRIGHT 2006 all the vectors needs to be moved along with the parallel transport.

```

1: function UPDATEHISTORY
2:   if  $k > m$  then remove  $s_{k-m}$ ,  $y_{k-m}$  and  $\rho_{k-m}$ 
3:    $s_k \leftarrow \Pi_{x_k, d_k}(\eta_k)(\log_{x_k}(x_{k+1}))$ 
4:    $y_k \leftarrow \nabla f_{k+1} - \Pi_{x_k, d_k}(\eta_k)(\nabla f_k)$ 
5:    $\rho_k \leftarrow 1 / \langle s_k, y_k \rangle_{x_{k+1}}$ 
6:   for  $i = k - 1, k - 2, \dots, k - l$  do  $\triangleright$  Transport parallelly
7:   |  $s_i \leftarrow \Pi_{x_k, d_k}(\eta_k)(s_i)$ ;  $y_i \leftarrow \Pi_{x_k, d_k}(\eta_k)(y_i)$   $\triangleright$  older history elements
8:   return  $\rho, s, y$ 
    
```

place all the vectors on $T_{x_{k+1}}S^+$. The operations required to build the history are collected in the function UPDATEHISTORY described in Algorithm A.4, which is the last piece needed to build Algorithm A.5, that describe the whole Riemannian L-BFGS procedure.

ALGORITHM A.5: Riemannian version of L-BFGS algorithm, based on algorithm 7.5 of NOCEDAL and WRIGHT 2006

```

Input:  $x_0, f$   $\triangleright$  Starting point and cost function
Input:  $m$   $\triangleright$  Memory size
Input:  $\epsilon, \delta$   $\triangleright$  Tolerances
1: for  $k = 1, 2, \dots, N_{\max}$  do
2:    $d_k \leftarrow \text{HESSIANGRADIENT}(\rho, s, y, \gamma_k)$   $\triangleright$  Descent direction
3:    $\phi(\eta) \leftarrow f(\gamma_{x_k, d}(\eta)); \phi'(\eta) \leftarrow \langle \nabla f(\gamma_{x_k, d}(\eta)), d_k \rangle_{x_k}$ 
4:    $\eta_k \leftarrow \text{LINESEARCH}(\phi, \phi')$   $\triangleright$  Step length
5:    $x_{k+1} \leftarrow \exp_{x_k}(\eta_k d_k)$ 
6:   if  $\|\nabla f_{k+1}\|_{x_{k+1}} < \epsilon$  or  $|f_{k+1} - f_k| < \delta$  then Break  $\triangleright$  Convergence
7:    $s, y, \rho \leftarrow \text{UPDATEHISTORY}$   $\triangleright$  Update history vectors
8:    $\gamma_k \leftarrow \langle s_k, y_k \rangle_{x_{k+1}} / \|y_k\|_{x_{k+1}}^2$   $\triangleright$  and prepare for next iteration
Output:  $x_k$ 
    
```

B | TECHNICAL DETAILS ON SKEW-NORMAL VARIATIONAL APPROXIMATION

In this provide some technical tool required for the Skew-Normal Variational Approximation (SNVA) described in Section 2.iii. We start by introducing the skew-normal distribution and some of its properties and then we provide the computations to obtain the exact expression for f_{sn} of Equation (2.14).

I SOME PROPERTIES OF THE SKEW-NORMAL DISTRIBUTION

Most of the properties and details of the skew-normal distribution discussed here comes from the seminal papers by AZZALINI and DALLA VALLE 1996; AZZALINI and CAPITANIO 1999 and later collected in the fundamental textbook AZZALINI 2013.

Given a random vector $\theta \in \mathbb{R}^p$, it is distributed according to a multivariate skew-normal distribution with parameters μ , Σ and ξ if its density is:

$$\begin{aligned} \theta &\sim \text{SN}(\mu, \Sigma, \xi) \quad \text{if} \\ q(\theta \mid \mu, \Sigma, \xi) &= 2\phi_{\Sigma}(\theta - \mu)\Phi(\xi^{\top}(\theta - \mu)) \end{aligned} \tag{B.1}$$

where $\phi_{\Sigma}(\theta - \mu)$ is the multivariate normal with mean μ and variance-covariance Σ , while Φ is the cumulative distribution of the univariate standard normal distribution. A useful quantity, which we will frequently use

to reduce the cumbersome notation in some of the more common expressions that will be needed for our work, is

$$\delta = \frac{\Sigma \zeta}{\sqrt{1 + \zeta^\top \Sigma \zeta}},$$

which is actually one of the main re-parametrization of the skew-normal.

The moment generating function for the skew-normal distribution can be written as

$$\mathbb{E} [e^{t^\top \theta}] = 2 \exp \left[t^\top \mu + \frac{t^\top \Sigma t}{2} \right] \Phi (t^\top \delta)$$

and from this we can immediately compute the mean and variance of a skew-normal distributed random variable

$$\mathbb{E} [\theta] = \mu + \sqrt{\frac{2}{\pi}} \delta \quad (\text{B.2})$$

$$\text{Var} [\theta] = \Sigma - \frac{2}{\pi} \delta \delta^\top. \quad (\text{B.3})$$

Moreover, it can be shown that the distribution is invariant under affine transformation, in the sense that

$$z = A\theta + b \sim \text{SN}(\mu_z, \Sigma_z, \zeta_z)$$

$$\text{with} \begin{cases} \mu_z = A\mu + b & \Sigma_z = A\Sigma A^\top \\ \zeta_z = A^{-\top} \zeta (1 + \zeta^\top \Sigma \zeta)^{-1/2} & \delta_z = A\delta \end{cases},$$

corresponding to Proposition 3 of [AZZALINI and CAPITANIO 1999](#).

Another useful observation is to recognize that

$$e^{t^\top \theta} q(\theta | \mu, \Sigma, \zeta) = 2 \exp \left[t^\top \mu + \frac{t^\top \Sigma t}{2} \right] \Phi (t^\top \delta) \tilde{q}(\theta | \mu, \Sigma, \zeta, t) \quad (\text{B.4})$$

where \tilde{q} is the density defined in ARNOLD et al. 2002 as

$$\tilde{q}(\text{vec}\theta \mid \mu, \Sigma, \zeta, t) = \frac{\Phi(\zeta^\top(\theta - \mu))}{\Phi(t^\top \delta)} \phi_\Sigma(\theta - \mu - \Sigma t).$$

Said $\tilde{\theta}$ a random variable distributed according to \tilde{q} , then its moment generating function and the first two moments can be written (with $\zeta_k(x) = d^k \log \Phi(x) / dx^k$) as

$$\begin{aligned} \mathbb{E} \left[e^{s^\top \tilde{\theta}} \right] &= 2 \exp \left[s^\top (\mu + \Sigma t) + \frac{s^\top \Sigma s}{2} \right] \frac{\Phi((t^\top + s^\top) \delta)}{\Phi(t^\top \delta)}, \\ \mathbb{E} [\tilde{\theta}] &= \mu + \Sigma t + \zeta_1(t^\top \delta) \delta, \\ \text{Var} [\tilde{\theta}] &= \Sigma + \zeta_2(t^\top \delta) \delta \delta^\top. \end{aligned}$$

II COMPUTATIONS FOR THE NORMAL SAMPLE

Considering the model of Equation (2.15), with the parametrization $\sigma^2 = e^\gamma$ to obtain the posterior written in Equation (2.16) and reported here for convenience

$$\begin{aligned} \log p(\theta \mid y) &= - \left(a + \frac{n}{2} \right) \langle t, \theta \rangle - e^{-\langle t, \theta \rangle} \left(b + \frac{\sum_i^n (y_i - \langle r, \theta \rangle)^2}{2} \right) \\ &\quad - \frac{\langle r, \theta \rangle^2}{2\sigma_v^2} + \kappa \end{aligned}$$

for the vector parameter $\theta = (v, \gamma)$, with $t = (0, 1)^\top$ and $r = (1, 0)^\top$ to select each component of the parameter vector.

Then we can compute f_{sn} by taking the expected value of each piece of Equation (2.16) with respect to the density of the skew-normal. By using the expressions for the moment generating function and the results for the

mean and variance we are able to obtain quite easily:

$$\begin{aligned}\mathbb{E}\left[\left(a + \frac{n}{2}\right) t^\top \theta\right] &= \left(a + \frac{n}{2}\right) \left(t^\top \mu + \sqrt{\frac{2}{\pi}} t^\top \delta\right) \\ \mathbb{E}\left[\mathbf{b} e^{-t^\top \theta}\right] &= 2\mathbf{b} \exp\left[-t^\top \mu + \frac{t^\top \Sigma t}{2}\right] \Phi(-t^\top \delta) \\ \mathbb{E}\left[\frac{\langle r, \theta \rangle^2}{2\sigma_\mu^2}\right] &= \frac{1}{2\sigma_\mu^2} \left[r^\top \Sigma r + \langle r, \mu \rangle^2 + 2\sqrt{\frac{2}{\pi}} \langle r, \mu \rangle \langle r, \delta \rangle\right]\end{aligned}$$

To compute the missing piece of the log-likelihood we need to observe that we can rewrite the integral as

$$\begin{aligned}\mathbb{E}\left[e^{-t^\top \theta} (y_i - r^\top \theta)^2\right] &= \int e^{-t^\top \theta} (y_i - r^\top \theta)^2 q(\theta | \mu, \Lambda, \mathbf{d}) d\theta \\ &= \int (y_i - r^\top \theta)^2 \underbrace{e^{-t^\top \theta} q(\theta | \mu, \Lambda, \mathbf{d})}_{(B.4)} d\theta \\ &= 2 \exp\left[t^\top \mu + \frac{t^\top \Lambda t}{2}\right] \Phi(t^\top \delta) \mathbb{E}\left[(y_i - r^\top \tilde{\theta})^2\right]\end{aligned}$$

Which we can then use to compute the last expected values and, by unifying all the term, to obtain the complete expression for f_{SN} written in Equation (2.17).

ACRONYMS

SPD	Symmetric and Positive-Definite
AI	Affine-Invariant
RR	Riemannian Registration
SVD	Singular Value Decomposition
ED	Eigenvalue Decomposition
EEG	Electroencephalography
fMRI	functional Magnetic Resonance Imaging
R-fMRI	Resting state functional Magnetic Resonance Imaging (fMRI)
NKI ₁	Nathan Kline Institute-Rockland Sample
MDRM	Minimum Distance to Riemannian Mean
MDS	Multi Dimensional Scaling
VA	Variational Approximation
SNVA	Skew-Normal Variational Approximation
GVA	Gaussian Variational Approximation
BFGS	Broyden Fletcher Goldfarb Shanno
L-BFGS	Limited memory BFGS
R-L-BFGS	Riemannian Limited memory BFGS
MLE	Maximum Likelihood Estimator
GLMM	Generalized Linear Mixed Model
MCMC	Markov Chain Monte Carlo

BIBLIOGRAPHY

- ABSIL, P.-A., MAHONY, R., and SEPULCHRE, R. (2008). *Optimization Algorithms on Matrix Manifolds*, vol. 36, Princeton University Press, 241 pp.
- AMARI, S.-I. (2016). *Information geometry and its applications*, Springer Japan.
- AMARI, S.-I. and NAGAOKA, H. (2000). *Methods of Information Geometry*, American Mathematical Society.
- ANDRADE, J. M., GÓMEZ-CARRACEDO, M. P., KRZANOWSKI, W., and KUBISTA, M. (2004). Procrustes rotation in analytical chemistry, a tutorial, *Chemometrics and Intelligent Laboratory Systems* 72.2, Advances in Chromatography and Electrophoresis - Conferentia Chemometrica 2003, Budapest, pp. 123–132.
- ARNOLD, B. C. et al. (2002). Skewed multivariate models related to hidden truncation and/or selective reporting, *Test* 11.1, pp. 7–54.
- ATKINSON, C. and MITCHELL, A. F. (1981). Rao's distance measure, *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 345–365.
- AXLER, S. (2015). *Linear Algebra Done Right*, Undergraduate Texts in Mathematics, Cham: Springer International Publishing.
- AZZALINI, A. and DALLA VALLE, A. (1996). The multivariate skew-normal distribution, *Biometrika* 83.4, pp. 715–726.
- AZZALINI, A. (2021). *The R package sn: The skew-normal and related distributions such as the skew-t and the SUN (version 2.0.1)*. Università degli Studi di Padova, Italia.

- AZZALINI, A. (2013). *The Skew-Normal and Related Families*, Institute of Mathematical Statistics Monographs, Cambridge University Press.
- AZZALINI, A. and CAPITANIO, A. (1999). Statistical applications of the multivariate skew-normal distribution, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.3, pp. 579–602.
- BARACHANT, A., BONNET, S., CONGEDO, M., and JUTTEN, C. (2012). Multiclass brain-computer interface classification by Riemannian geometry, *IEEE Transactions on Biomedical Engineering* 59.4, pp. 920–928.
- BERGER, M. (2003). *A Panoramic View of Riemannian Geometry*, Berlin, Heidelberg: Springer-Verlag, 824 pp.
- BETANCOURT, M., BYRNE, S., LIVINGSTONE, S., and GIROLAMI, M. (2017). The geometric foundations of Hamiltonian Monte Carlo, *Bernoulli* 23.4A, pp. 2257–2298.
- BHATIA, R. (1997). *Matrix analysis*, Springer-Verlag New York.
- BHATIA, R. (2007). *Positive definite matrices*, Princeton University Press.
- BHATIA, R. and CONGEDO, M. (2019). Procrustes problems in Riemannian manifolds of positive definite matrices, *Linear Algebra and its Applications* 563, pp. 440–445.
- BHATTACHARYA, A. and DUNSON, D. B. (2010). Nonparametric Bayesian density estimation on manifolds with applications to planar shapes, *Biometrika* 97.4, pp. 851–865.
- BINI, D. A. and IANNAZZO, B. (2013). Computing the Karcher mean of symmetric positive definite matrices, *Linear Algebra and its Applications* 438.4, pp. 1700–1710.
- BLEI, D. M., KUCUKELBIR, A., and MCAULIFFE, J. D. (2017). Variational Inference: A Review for Statisticians, *Journal of the American Statistical Association* 112.518, pp. 859–877.

- BORG, I. and GROENEN, P. J. F. (2005). *Modern multidimensional scaling: theory and applications*, 2nd ed., Springer series in statistics, New York: Springer, 614 pp.
- BRADBURY, J. et al. (2018). *JAX: composable transformations of Python+NumPy programs*, version 0.2.5.
- BRONSTEIN, M. M., BRUNA, J., COHEN, T., and VELIČKOVIĆ, P. (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, arXiv: 2104.13478 [cs.DS].
- BURDEN, R., FAIRES, J., and REYNOLDS, A. (1985). *Numerical Analysis*, Prindle, Weber & Schmidt.
- CANALE, A., DURANTE, D., PACI, L., and SCARPA, B., eds. (2018). *Studies in Neural Data Science: StartUp Research 2017, Siena, Italy, June 25–27*, vol. 257, Springer.
- CATTAN, G., RODRIGUES, P. L. C., and CONGEDO, M. (2018). *EEG Alpha Waves dataset*, URL: <https://doi.org/10.5281/zenodo.2605110>.
- CONTRERAS-REYES, J. E. and ARELLANO-VALLE, R. B. (2012). Kullback–Leibler divergence measure for multivariate skew-normal distributions, *Entropy* 14.9, pp. 1606–1626.
- DEMPSTER, A. P., LAIRD, N. M., and RUBIN, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 39.1, pp. 1–22.
- DESIKAN, R. S. et al. (2006). An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest, *Neuroimage* 31.3, pp. 968–980.
- DRYDEN, I. L., FAGHIHI, M. R., and TAYLOR, C. C. (1997). Procrustes Shape Analysis of Planar Point Subsets, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 59.2, pp. 353–374.

- DRYDEN, I. L., KOLOYDENKO, A., and ZHOU, D. (2009). Non-Euclidean statistics for covariance matrices, with applications to diffusion tensor imaging, *The Annals of Applied Statistics* 3.3.
- FINOS, L. (2018). *flip: Multivariate Permutation Tests*, R package version 2.5.0.
- FÖRSTNER, W. and MOONEN, B. (2003). A metric for covariance matrices, in *Geodesic Challenges 3rd millennium*, ed. by E. W. GRAFAREND, F. W. KRUMM, and V. S. SCHWARZE, Berlin: Springer, pp. 299–309.
- GABAY, D. (1982). Minimizing a differentiable function over a differential manifold, *Journal of Optimization Theory and Applications* 37.2, pp. 177–219.
- GIROLAMI, M. and CALDERHEAD, B. (2011). Riemann manifold langevin and hamiltonian monte carlo methods, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73, pp. 123–214.
- GOODALL, C. (1991). Procrustes Methods in the Statistical Analysis of Shape, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 53.2, pp. 285–339.
- GOWER, J. C. and DIJKSTERHUIS, G. B. (2004). *Procrustes Problems*, Oxford: Oxford University Press.
- GREEN, B. F. (1952). The orthogonal approximation of an oblique structure in factor analysis, *Psychometrika* 17.4, pp. 429–440.
- HAMILTON, W. R. (1835). On a general method in dynamics, *Philosophical Transactions of the Royal Society of London* 125, pp. 95–144.
- HAXBY, J. V. et al. (2001). Distributed and overlapping representations of faces and objects in ventral temporal cortex, *Science* 293.5539, pp. 2425–2430.
- HOLBROOK, A., LAN, S., VANDENBERG-RODES, A., and SHAHBABA, B. (2018). Geodesic Lagrangian Monte Carlo over the space of positive definite

- matrices: with application to Bayesian spectral density estimation, *Journal of Statistical Computation and Simulation* 88.5, pp. 982–1002.
- HUANG, Z. and VAN GOOL, L. (2017). A riemannian network for SPD matrix learning, in *Thirty-First AAAI Conference on Artificial Intelligence*, pp. 2036–2042.
- KULLBACK, S. and LEIBLER, R. A. (1951). On information and sufficiency, *The annals of mathematical statistics* 22.1, pp. 79–86.
- LAN, S. et al. (2020). Flexible Bayesian Dynamic Modeling of Correlation and Covariance Matrices, *Bayesian Analysis* 15.4, pp. 1199–1228.
- LEE, J. M. (2009). *Manifolds and differential geometry*, Graduate studies in mathematics v. 107, Providence, Rhode Island: American Mathematical Society, 671 pp.
- LEE, J. M. (2003). *Introduction to Smooth Manifolds*, 2nd ed., New York: Springer, 628 pp.
- LI, D. and DUNSON, D. B. (2020). Classification via local manifold approximation, *Biometrika* 107.4, pp. 1013–1020.
- LIN, T. and ZHA, H. (2008). Riemannian manifold learning, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.5, pp. 796–809.
- LIN, Z. (2019). Riemannian Geometry of Symmetric Positive Definite Matrices via Cholesky Decomposition, *SIAM Journal on Matrix Analysis and Applications* 40.4, pp. 1353–1370.
- MALAGÒ, L. and PISTONE, G. (2015). Second-Order Optimization over the Multivariate Gaussian Distribution, in *International Conference on Geometric Science of Information*, Springer, pp. 349–358.
- MARSHALL, A. W., OLKIN, I., and ARNOLD, B. C. (2010). *Inequalities: Theory of Majorization and Its Applications*, 2nd ed., Springer Series in Statistics, Springer New York.

- MCCARTHY, D. J., CAMPBELL, K. R., LUN, A. T., and WILLS, Q. F. (2017). Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R, *Bioinformatics* 33.8, pp. 1179–1186.
- MCCRAE, R. et al. (1996). Evaluating Replicability of Factors in the Revised NEO Personality Inventory: Confirmatory Factor Analysis Versus Procrustes Rotation, English (US), *Journal of Personality and Social Psychology* 70.3, pp. 552–566.
- MOAKHER, M. (2005). A Differential Geometric Approach to the Geometric Mean of Symmetric Positive-Definite Matrices, *SIAM J. Matrix Anal. Appl.* 26.3, pp. 735–747.
- NIELSEN, F. (2020). An elementary introduction to information geometry, *Entropy* 22.10, p. 1100.
- NOCEDAL, J. and WRIGHT, S. J. (2006). *Numerical optimization*, 2nd ed., New York: Springer.
- NOONER, K. B. et al. (2012). The NKI-Rockland sample: a model for accelerating the pace of discovery science in psychiatry, *Frontiers in neuroscience* 6, p. 152.
- ORMEROD, J. T. (2011). Skew-normal variational approximations for Bayesian inference, URL: <http://talus.maths.usyd.edu.au/u/jormerod/JT0papers/snva.pdf>.
- ORMEROD, J. T. and WAND, M. P. (2010). Explaining variational approximations, *American Statistician* 64.2, pp. 140–153.
- ORMEROD, J. T. and WAND, M. P. (2012). Gaussian Variational Approximate Inference for Generalized Linear Mixed Models, *Journal of Computational and Graphical Statistics* 21.1, pp. 2–17.
- PARISI, G. (1988). *Statistical field theory*, Frontiers in Physics v. 66, Redwood City, CA: Addison-Wesley Pub. Co, 352 pp.

- PATRANGENARU, V. and ELLINGSON, L. (2016). *Nonparametric statistics on manifolds and their applications to object data analysis*, CRC Press., 541 pp.
- PEDREGOSA, F. et al. (2011). Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research* 12, pp. 2825–2830.
- PENNEC, X., FILLARD, P., and AYACHE, N. (2006). A riemannian framework for tensor computing, *International Journal of computer vision* 66.1, pp. 41–66.
- PESARIN, F. (2001). *Multivariate permutation tests: with applications in biostatistics*, vol. 240, Wiley Chichester.
- PETERSEN, P. (2016). *Riemannian geometry.*, 3rd ed., Graduate texts in mathematics, Springer.
- R CORE TEAM (2021). *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
- RAMSAY, J. O. and SILVERMAN, B. W. (2002). *Applied functional data analysis: methods and case studies*, vol. 77, Springer.
- RAO, C. R. (1945). Information and accuracy attainable in the estimation of statistical parameters, *Bulletin of the Calcutta Mathematical Society* 37 (3), pp. 81–91.
- RAPCSÁK, T. (1991). Geodesic convexity in nonlinear optimization, *Journal of Optimization Theory and Applications* 69.1, pp. 169–183.
- ROHLF, F. J. and SLICE, D. (1990). Extensions of the Procrustes method for the optimal superimposition of landmarks, *Systematic biology* 39.1, pp. 40–59.
- SAID, S., BOMBRUN, L., BERTHOUMIEU, Y., and MANTON, J. H. (2017). Riemannian Gaussian Distributions on the Space of Symmetric Positive Definite Matrices, *IEEE Transactions on Information Theory* 63.4, pp. 2153–2170.

- SAITO, V. S., FONSECA-GESSNER, A. A., and SIQUEIRA, T. (2015). How should ecologists define sampling effort? The potential of procrustes analysis for studying variation in community composition, *Biotropica* 47.4, pp. 399–402.
- SUSSMAN, G. J. and WISDOM, J. (2013). *Functional differential geometry*, vol. 1, Cambridge, Massachussets: MIT Press, 228 pp.
- TENENBAUM, J. B., DE SILVA, V., and LANGFORD, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction, *Science* 290.5500, pp. 2319–2323.
- TUAN, H. D., TRANG, T. T. N., and HIEU, D. T. (2010). Positive definite preserving linear transformations on symmetric matrix spaces, arXiv: 1008.1347 [cs.DS].
- VISHNOI, N. K. (2018). Geodesic convex optimization: Differentiation on manifolds, geodesics, and convexity, arXiv: 1806.06373 [cs.DS].
- WAND, M. P. (2014). Fully Simplified Multivariate Normal Updates in Non-Conjugate Variational Message Passing, *Journal of Machine Learning Research* 15.39, pp. 1351–1369.
- WESTFALL, P. H. and YOUNG, S. S. (1993). *Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment*, New York: Wiley.
- WISHART, J. (1928). The generalised product moment distribution in samples from a normal multivariate population, *Biometrika* 20.1/2, pp. 32–52.
- ZANELLA, A., CHIARI, M., and WIN, M. Z. (2009). On the marginal distribution of the eigenvalues of wishart matrices, *IEEE Transactions on Communications* 57.4, pp. 1050–1060.
- ZANINI, P. et al. (2018). Transfer Learning: A Riemannian Geometry Framework With Applications to Brain–Computer Interfaces, *IEEE Transactions on Biomedical Engineering* 65.5, p. 10.

Jacopo SCHIAVON

CURRICULUM VITAE

CONTACT INFORMATION

ADDRESS Department of Statistical Sciences
Università degli Studi di Padova
via Cesare Battisti, 241
35121 Padova, Italy

PHONE +39 334 200 6161

MAIL jacopo.schiavon.1@phd.unipd.it

WEBSITE jschiavon.github.io

CURRENT POSITION

APR 2022 PhD Student in Statistical Sciences

OCT 2018 Università degli Studi di Padova
Thesis Title: *Differential Geometry of symmetric and positive definite matrices for statistical applications*
Supervisor: Prof. Antonio Canale
Co-Supervisors: Prof. Mauro Bernardi, Prof. Livio Finos

EDUCATION

SEP 2017 MSc in PHYSICS, Università degli Studi di Padova 110/110
Dissertation title: *Detection of causality in complex systems*
Supervisor: Phd. Samir Suweis

DEC 2014 BSc in PHYSICS, Università degli Studi di Padova 94/110

OTHER EDUCATIONAL EXPERIENCES

- SEP 2019 Probabilistic Machine Learning (Bocconi University)
Instructor: David Dunson
- OCT 2019 School on Efficient Scientific Computing (INFN)
Efficiency and performance in modern computing (modern C++, GPUs, clusters)
- JUN 2021 DeepLearning.AI TensorFlow Developer
by DeepLearning.AI on Coursera, certificate ID: [BY2NPJRQ4ZDG](#)
- AUG 2021 Deep Learning
by DeepLearning.AI on Coursera, certificate ID: [E4P5LCPJJNY9](#)

OTHER WORK EXPERIENCES

- OCT 2018 *Data Scientist Consultant* at BIP, Milan (IT)
- OCT 2017 Various Data Science projects completed for clients of the Retail Industry, and active collaboration in project for clients in Telco and Finance industries.

COMPUTER SKILLS

Advanced programming skills in Python and C++ (for data analysis), good knowledge of the TensorFlow and Keras frameworks and basic knowledge of the R statistical software, Julia and MATLAB programming language. Advanced knowledge of the \LaTeX typesetting system for papers and reports, presentation and books. Personal interest in developing multiplatform UI interfaces with the Flutter and Qt ecosystems and basics of web programming with html / CSS and Javascript.

LANGUAGE SKILLS

Italian (mother tongue), English (comprehension C₁, production B₂), French (comprehension B₂, production B₁)

PUBLICATIONS

Schiavon, J., Bernardi, M. and Canale, A. (2021). Riemannian optimization on the space of covariance matrices. *Book of Short Papers SIS 2021* [ISBN 9788891927361]

CONFERENCE PRESENTATIONS

JUN 2021 Riemannian optimization on the space of covariance matrices
Contributed presentation, *SIS 2021*, virtual