université
PARIS-SACLAY

Riemannian geometry for statistical estimation and learning: application to remote sensing

Géométrie riemannienne pour l'estimation et l'apprentissage statistiques : application à la télédétection

Thèse de doctorat de l'Université Paris-Saclay

École doctorale n° 580 : Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat : Traitement du signal et des images
Graduate School : Informatique et sciences du numérique
Référent : CentraleSupélec

Thèse préparée dans l'unité de recherche SONDRA (Université Paris-Saclay, CentraleSupélec, ONERA), sous la direction de **Jean-Philippe OVARLEZ**, Directeur de recherche ONERA, la co-direction de **Guillaume GINOLHAC**, Professeur des universités, le co-encadrement de **Chengfang REN**, Maître de conférences, et le co-encadrement de **Arnaud BRELOY**, Maître de conférences.

Thèse soutenue à Paris-Saclay, le 25 novembre 2022, par

Antoine COLLAS

Composition du jury
Membres du jury avec voix délibérative

Audrey GIREMUS Professeur des universités, Université de Bordeaux (IMS)	Présidente
Nicolas LE BIHAN Directeur de recherche, CNRS (GIPSA-Lab)	Rapporteur & Examinateur
Cédric RICHARD Professeur des universités, Université Côte d'Azur (Lagrange)	Rapporteur & Examinateur
Nicolas BOUMAL Assistant professor, EPFL (Institute of Mathematics)	Examinateur
Alexandre GRAMFORT Senior research scientist, Meta (Reality Labs)	Examinateur

THESE DE DOCTORAT

NNT : 2022UPASG075

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Géométrie riemannienne pour l'estimation et l'apprentissage statistiques : application à la télédétection

Mots-clés traitement du signal, apprentissage automatique, geométrie riemannienne, optimisation, statistiques robustes, observation de la Terre

Résumé:

Les systèmes de télédétection offrent une opportunité accrue d'enregistrer des séries temporelles d'images multivariées de la surface de la Terre. Ainsi, l'intérêt pour les outils automatiques de traitement de ces données augmente considérablement. Dans cette thèse, nous proposons un *pipeline de partitionnement et de classification* pour segmenter des séries temporelles d'images multivariées. Pour ce faire, des paramètres de lois de probabilité sont estimés de manière robuste puis partitionnés ou classifiés. Une grande partie de la thèse est consacrée à la théorie de la *géométrie riemannienne* et à son sous-domaine, la *géométrie de l'information*, qui étudie les variétés riemanniennes dont les points sont des distributions de probabilité. Elle permet d'estimer des paramètres de lois de probabilité très rapidement, même sur des problèmes à grande échelle, mais aussi de calculer des centres de masse riemanniens. En effet, des divergences sont développées pour mesurer les proximités entre les paramètres estimés. Ensuite, des groupes de paramètres sont moyennés en calculant leurs centres de masse riemanniens associés à ces divergences. Ainsi, nous adaptons des algorithmes classiques d'apprentissage automatique tels que le *K-means++* ou le *classifieur du centroïde le plus proche* à des variétés riemanniennes. Ces algorithmes ont été mis en œuvre pour de nombreuses combinaisons de paramètres, divergences et centres de masse riemanniens et testés sur des jeux de données réels tels que l'image *Indian pines* et le grand jeu de données de cartographie des types de cultures *Breizhcrops*.

Title: Riemannian geometry for statistical estimation and learning: application to remote sensing

Keywords: signal processing, machine learning, Riemannian geometry, optimization, robust statistics, earth observation

Abstract:

Remote sensing systems offer an increased opportunity to record multi-temporal and multi-dimensional images of the earth's surface. This opportunity greatly increases the interest in data processing tools based on multivariate image time series. In this thesis, we propose a *clustering-classification pipeline* to segment these data. To do so, *robust statistics* are estimated and then clustered or classified to obtain a segmentation of the original multivariate image time series. A large part of the thesis is devoted to the theory of *Riemannian geometry* and its subfield, the *information geometry*, which studies Riemannian manifolds whose points are probability distributions. It allows to estimate robust statistics very quickly, even on large scale problems, but also to compute Riemannian centers of mass. Indeed, divergences are developed to measure the proximities between the estimated statistics. Then, groups of statistics are averaged by computing their Riemannian centers of mass associated to these divergences. Thus, we adapt classical machine learning algorithms such as the *K-means++* or the *Nearest centroid classifier* to Riemannian manifolds. These algorithms have been implemented for many different combinations of statistics, divergences and Riemannian centers of mass and tested on real datasets such as the *Indian pines* image and the large crop type mapping dataset *Breizhcrops*.

Acknowledgements

Before thanking all the people who helped me in the achievement of this thesis, I mention here the people who supervised me. They do not all appear on the front page of this manuscript although they each contributed to the realization of this thesis:

- Jean-Philippe OVARLEZ, Director, Directeur de recherche, ONERA Palaiseau (DEMR) and CentraleSupélec, Université Paris-Saclay (SONDRA),
- Guillaume GINOLHAC, co-Director, Professeur des universités, Université Savoie Mont Blanc (LISTIC),
- Chengfang REN, Maître de conférences, CentraleSupélec, Université Paris-Saclay (SONDRA),
- Arnaud BRELOY, Maître de conférences, Université Paris Nanterre (LEME),
- Florent BOUCHARD, Chargé de recherche, CNRS, CentraleSupélec, Université Paris-Saclay (L2S).

I now switch to French for the rest of the acknowledgements. Cette thèse n'aurait pu se dérouler ainsi sans l'aide des personnes qui m'ont entouré durant ces trois années.

Tout d'abord, je remercie mes encadrants de thèse dont le dévouement m'a permis de réaliser ce travail. Jean-Philippe, tu as cru en moi dès le départ, m'as dirigé et poussé à approfondir mes connaissances tout au long de ma thèse. Je ne pourrai jamais assez te remercier de m'avoir orienté vers ce sujet de thèse alliant statistiques, optimisation et géométrie riemannienne. Guillaume, tu a été présent tout au long de ces trois années. Tu m'as dirigé à travers les difficultées rencontrées et ton dévouement a été largement au-delà du devoir. De même, je ne pourrai jamais assez te remercier de m'avoir introduit ce sujet de thèse. Aussi, je te remercie, ainsi que toute ta famille, pour toutes les fois où vous m'avez hébergé à Annecy. Arnaud, je te remercie d'avoir été présent tout au long de ma thèse. Tu m'as apporté des idées intéressantes et a su m'aider dans les moments difficiles qu'impliquent la recherche. Chengfang, tu as toujours été là pour répondre à mes questions. Je te remercie pour tes nombreuses remarques sur mes travaux ainsi que pour tes conseils tout au long de ma thèse. Enfin, Florent, tu m'as introduit les principaux outils en optimisation sur variétés riemanniennes et m'as aidé à déchiffrer les livres références dans le domaine. Je t'en remercie beaucoup

et n'oublierai jamais ce fort investissement alors que tu étais encore postdoc à Annecy !

Ensuite, je remercie mon jury de thèse d'avoir accepté d'évaluer mes travaux de recherche et d'être venus jusqu'à Saclay. En particulier, je remercie Nicolas Le Bihan et Cédric Richard pour les lectures approfondies de ce manuscrit. Recevoir des commentaires détaillés de ses pairs est toujours très appréciable et précieux.

Je remercie toutes les personnes du laboratoire SONDRA. Vous m'avez donné un cadre de travail très agréable durant ces trois années de thèse. Je remercie tous les doctorants et postdocs qui m'ont accueilli et avec qui j'ai eu le plaisir de commencer ma thèse: Agustin, Ammar, Bruno, Cyprien, Dihia, Giovanni, Nathan, Thibault, et Vlad. Je remercie également tous les doctorants et stagiaires arrivés au cours de ma thèse: Alexandre, Axel, Ba-Huy, Florent, Hugo, Louis, Max, Pierre, Thomas, Tyler, et Yanisse. Vous avez su redonner beaucoup d'animation au laboratoire après les nombreux confinements. Je remercie Virginie et Isabelle pour l'aide durant mes nombreuses tâches administratives. Aussi, je remercie Israel et Mohammed pour tous les déjeuners et moments passés ensemble. Je remercie également le directeur de SONDRA, Stéphane, de m'avoir permis de travailler avec autant de libertés et sur des recherches aussi ambitieuses.

Enfin, je remercie ma famille et mes amis de l'encouragement tout au long de ces trois années. Votre compagnie en période de Covid a été fondamentale. Je remercie mes parents de m'avoir donné cette passion pour les sciences. Cette envie d'approfondir mes connaissances m'a conduit à réaliser cette thèse. Je remercie également ma compagne, Tiphaine, qui m'a épaulé durant les moments difficiles de cette thèse et m'a tiré vers le haut. Cette thèse vous est dédiée.

Mes derniers mots vont à Thierry et Orian avec qui j'ai partagé de bons moments à SONDRA que je n'oublierai pas.

Introduction

Remote sensing systems offer an increased opportunity to record multi-temporal and multi-dimensional images of the earth's surface by improving temporal and spatial resolution. Indeed, in the recent years, many countries and companies have deployed satellites or run UAV (Unmanned Aerial Vehicle) for earth observation. Examples of these remote sensing instruments are the Sentinel, Landsat and TerraSAR-X satellites or the UAVSAR. This big increase in the number, performance and diversity of these systems enables the development of many applications such as the monitoring of the environment (*e.g.* glaciers, forests, urbanism), major events (*e.g.* earthquakes, floods), human activity (*e.g.* maritime and borders surveillance) as well as weather forecasting. These opportunities greatly increase the interest of data processing tools based on multivariate image time series.

A recent trend in machine learning, mostly coming from the EEG/MEG (Electroencephalography/Magnetoencephalography) community, proposes to estimate covariance matrices from data and then to classify them using *Riemannian geometry*. Indeed, the theory of *Riemannian geometry* and its subfield, the *information geometry*, suits well to covariances matrices which are then seen as parameters of centered multivariate Gaussian distributions. In this case, the classical straight line is replaced by geodesics, the Euclidean distance by Riemannian distances and the arithmetic mean by Riemannian centers of mass. In practice, the use of Riemannian geometry gives much better performance than its Euclidean counterpart when dealing with covariance matrices. In this thesis, we propose to apply this *clustering-classification pipeline* to remote sensing data and to extend it in multiple ways. The contributions are fourfold.

First, statistical estimators are developed by leveraging the theory of optimization on Riemannian manifolds. In particular, gradient descent methods are developed to estimate jointly locations (centers of the distribution) and covariance matrices. This is of first importance for applications where the location is a discriminative feature contrary to EEG/MEG. Furthermore, in practice data can not always be assumed to be distributed as Gaussian distribution due to outliers or heavy tailed distributions. To remediate to this problem, we leverage the theory of robust statistics to construct new Riemannian based robust estimators. Finally, estimators are developed for structured covariance matrices when dealing with high dimensional data. All these Riemannian base estimators are fast and suit well for large scale datasets.

Second, intrinsic Cramér-Rao bounds (ICRB) are derived to analyze the performance of estimators of structured covariance matrices. These ICRBs lower bound the mean squared Riemannian distance between estimated pa-

rameters and the true one. This enables to take into account constraints of the parameter space.

Third, divergences between statistics and their associated centers of mass are proposed. These divergences, and the associated centers of mass, are chosen with respect to the statistical model to obtain better performance in practice. Also, gradient based Riemannian optimization algorithms are derived to compute efficiently these centers of mass.

A fourth contribution is the development of metric learning algorithms. Metric learning methods propose to cluster or classify raw data with a learned Mahalanobis distance. In this thesis, we demonstrate that some classical metric learning problems can be seen as covariance estimation problems. With this novel view, we derive two new Riemannian based metric learning algorithms.

All these contributions are tested on generated data as well as real datasets such as the *Indian pines* image and the large scale crop type mapping dataset *Breizhcrops* and show promising results.

List of publications :

Journals :

- A. Mian, **A. Collas**, A. Breloy, G. Ginolhac and J.-P. Ovarlez, "Robust Low-Rank Change Detection for Multivariate SAR Image Time Series," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 13, pp. 3545-3556, 2020.

- **A. Collas**, F. Bouchard, A. Breloy, G. Ginolhac, C. Ren and J.-P. Ovarlez, "Probabilistic PCA From Heteroscedastic Signals: Geometric Framework and Application to Clustering," in IEEE Transactions on Signal Processing, vol. 69, pp. 6546-6560, 2021.

- **A. Collas**, A. Breloy, C. Ren, G. Ginolhac and J.-P. Ovarlez, "Riemannian optimization for non-centered mixture of scaled Gaussian distributions," submitted to IEEE Transactions on Signal Processing.

Conferences proceedings :

- **A. Collas**, F. Bouchard, A. Breloy, C. Ren, G. Ginolhac and J.-P. Ovarlez, "A Tyler-Type Estimator of Location and Scatter Leveraging Riemannian Optimization," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, Canada, pp. 5160-5164.

- **A. Collas**, F. Bouchard, G. Ginolhac, A. Breloy, C. Ren and J.-P. Ovarlez, "On the Use of Geodesic Triangles between Gaussian Distributions for Classification Problems," ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, pp. 5697-5701.

- **A. Collas**, A. Breloy, G. Ginolhac, C. Ren and J.-P. Ovarlez, "Robust Geometric Metric Learning," 2022 30th European Signal Processing Conference (EUSIPCO), Belgrade, Serbia. **Best student paper award**

- **A. Collas**, A. Breloy, G. Ginolhac, C. Ren and J.-P. Ovarlez, "Apprentissage robuste de distance par géométrie riemannienne", GRETSI 2022 XXVIIIème colloque, Nancy, France.

Contents

Notations

General symbols

x	Scalar (lowercase character)
$\boldsymbol{x}$	Vector (bold lowercase character)
$\boldsymbol{X}$	Matrix (bold uppercase character)
p	Dimension of data
n	Number of data per batch
M	Number of batches
K	Number of classes
acosh	Inverse hyperbolic cosine
Card	Cardinality operator, *i.e.* returns the number of elements of a given set
$\mathfrak{Re}(x)$	Real part of x
i	Imaginary unit ($i^2 = -1$)
sign	Sign function: $\operatorname{sign}(x)$ returns 1 if $x \geq 0$ and -1 otherwise
$\underset{\theta}{\operatorname{minimize}}\ h(\theta)$	Minimization problem of the real valued function h
$\underset{\theta}{\arg\min}\ h(\theta)$	Argument minimizing the real valued function h
$\mathrm{D}\, h(\theta)[\xi]$	Directional derivative of h at θ in the direction ξ
$\operatorname{grad} h(\theta)$	Gradient of h at θ

Sets

$[\![n_1, n_2]\!]$	Set of integers from n_1 and n_2
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^p$	Set of real valued vectors of size p
$\mathbb{R}_*^+$	Set of strictly positive real numbers
$(\mathbb{R}_*^+)^n$	Set of n dimensional vectors with strictly positive entries
$\mathbb{C}$	Set of complex numbers
$\mathbb{C}^p$	Set of complex valued vectors of size p
GL_p	Set of $p \times p$ invertible matrices
S^{p-1}	$p-1$ dimensional sphere in $\mathbb{R}^p$
$\mathcal{O}_p$	Set of $p \times p$ orthogonal matrices
$\mathcal{U}_p$	Set of $p \times p$ unitary matrices
$\mathsf{St}_{p,k}$	Set of orthogonal basis of k-dimensional subspaces in $\mathbb{R}^p$ (or $\mathbb{C}^p$) (Stiefel manifold)
$\mathsf{Gr}_{p,k}$	Set of k-dimensional subspaces of $\mathbb{R}^p$ (or $\mathbb{C}^p$) (Grassmann manifold)

$\mathcal{S}_p$	Set of $p \times p$ symmetric matrices
$\mathcal{S}_p^+$	Set of $p \times p$ symmetric positive semi definite matrices
$\mathcal{S}_p^{++}$	Set of $p \times p$ symmetric positive definite matrices
$\mathcal{SS}_p^{++}$	Set of $p \times p$ symmetric positive definite matrices with unit determinant
$\mathcal{A}_p$	Set of $p \times p$ skew-symmetric matrices
$\mathcal{H}_p$	Set of $p \times p$ Hermitian matrices
$\mathcal{H}_p^{++}$	Set of $p \times p$ Hermitian positive definite matrices

Linear algebra

$.^T, .^H$	Transpose, transpose conjugate
$\otimes$	Kronecker product
$\mathcal{E}$	A vector space
$\mathrm{span}(\boldsymbol{A})$	Span/image of $\boldsymbol{A}$, *i.e.* $\{\boldsymbol{A}\boldsymbol{x} : \text{ for all } \boldsymbol{x} \in \mathbb{R}^p\}$
$\lvert\boldsymbol{A}\rvert$	Determinant of $\boldsymbol{A}$
$\mathrm{Tr}(\boldsymbol{A})$	Trace of $\boldsymbol{A}$
$\mathrm{rank}(\boldsymbol{A})$	Rank of $\boldsymbol{A}$
$\mathrm{sym}(\boldsymbol{A})$	Symmetric part of $\boldsymbol{A}$, *i.e.* $\mathrm{sym}(\boldsymbol{A}) = \frac{\boldsymbol{A}+\boldsymbol{A}^T}{2}$
$\mathrm{herm}(\boldsymbol{A})$	Hermitian part of $\boldsymbol{A}$, *i.e.* $\mathrm{herm}(\boldsymbol{A}) = \frac{\boldsymbol{A}+\boldsymbol{A}^H}{2}$
$\mathrm{vec}(\theta)$	Vectorize θ, *i.e.* stacks the coordinates of θ into a vector
$\mathrm{diag}(\boldsymbol{x})$	Diagonal matrix whose diagonal contains the elements of the vector $\boldsymbol{x}$
$\boldsymbol{0}_{l\times m}$	Zero matrix of size $l \times m$
$\boldsymbol{I}_p$	Identity matrix of size $p \times p$
$\boldsymbol{1}_p$	Vector of size p whose elements are equal to 1

Statistics

$\overset{d}{=}$	Equality of distribution
$f(.;\theta)$	Probability density function parametrized by θ
$\mathcal{N}$	Gaussian distribution
$\mathbb{C}\mathcal{N}$	Complex Gaussian distribution
$\mathcal{X}$	Sample space (a linear space)
X	Random variable
$\boldsymbol{x}_i$	Vector sample
$\mathcal{M}$	Feature space (a Riemannian manifold)
θ	Parameter
$\mathbb{E}[X]$	Expectation of the random variable X
$\mathcal{L}$	(Negative) log-likelihood
$\boldsymbol{\mu}$	Location

$\hat{\boldsymbol{\mu}}_{\mathsf{SM}}$	Sample mean, *i.e.* $\hat{\boldsymbol{\mu}}_{\mathsf{SM}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i$
$\boldsymbol{\Sigma}$	Covariance/Scatter matrix
$\hat{\boldsymbol{\Sigma}}_{\mathsf{SCM}}$	Sample covariance matrix, *i.e.* $\hat{\boldsymbol{\Sigma}}_{\mathsf{SCM}} = \frac{1}{n}\sum_{i=1}^{n} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{\mathsf{SM}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{\mathsf{SM}})^T$
$\boldsymbol{\tau}$	Texture parameters
$\boldsymbol{U}$	Orthogonal basis (element of the Stiefel manifold $\mathsf{St}_{p,k}$)
$\boldsymbol{F}_\theta$	Fisher information matrix

Riemannian geometry

$\mathcal{E}$	Ambient space
$\mathcal{M}$	Riemannian manifold
$T_\theta\mathcal{M}$	Tangent space at $\theta \in \mathcal{M}$
η, ξ	Tangent vectors
$P_\theta^{\mathcal{M}}$	Orthogonal projection from $\mathcal{E}$ onto $T_\theta\mathcal{M}$
$\mathfrak{F}(\mathcal{M})$	Set of scalar fields
$\mathfrak{X}(\mathcal{M})$	Set of vector fields
γ	Geodesic
inj	Injectivity radius
$\nabla^{\mathcal{M}}$	Levi-Civita connection
$\exp_\theta^{\mathcal{M}}$	Riemannian exponential mapping
$\mathcal{T}_{\theta_1,\theta_2}^{\mathcal{M}}$	Parallel transport from $T_{\theta_1}\mathcal{M}$ onto $T_{\theta_2}\mathcal{M}$
$\log_\theta^{\mathcal{M}}$	Riemannian logarithmic mapping
$d^{\mathcal{M}}$	Riemannian distance
$R_\theta^{\mathcal{M}}$	Retraction
r	Curve associated to a retraction, *i.e.* $r(t) = R_\theta^{\mathcal{M}}(t\xi)$
$\dot{r}$	Derivative of r, *i.e.* $\dot{r}(t) = \frac{d}{dt}r(t)$
$\ddot{r}$	Second derivative of r, *i.e.* $\ddot{r}(t) = \frac{d^2}{dt^2}r(t)$
$\mathrm{grad}_{\mathcal{M}}\, h(\theta)$	Riemannian gradient of h at $\theta \in \mathcal{M}$

Riemannian quotient manifolds

$\overline{\mathcal{M}}$	Riemannian manifold
$T_{\bar{\theta}}\overline{\mathcal{M}}$	Tangent space at $\bar{\theta} \in \overline{\mathcal{M}}$
$\bar{\eta}, \bar{\xi}$	Tangent vectors of $T_{\bar{\theta}}\overline{\mathcal{M}}$
$\sim$	Equivalence relation
$[\bar{\theta}]$	Equivalence class
$\pi(\bar{\theta})$	Natural/Canonical projection, *i.e.* $\pi(\bar{\theta}) = [\bar{\theta}]$

$\mathcal{V}_{\bar{\theta}}, \mathcal{H}_{\bar{\theta}}$	Vertical and Horizontal spaces
$\mathcal{M}$	Riemannian quotient manifold, *i.e.* $\mathcal{M} = \overline{\mathcal{M}}/\sim$
$T_\theta\mathcal{M}$	Tangent space at $\theta = [\bar{\theta}] \in \mathcal{M}$
η, ξ	Tangent vectors of $T_\theta\mathcal{M}$
$\mathrm{lift}_{\bar{\theta}}(\xi)$	Horizontal lift of ξ at $\bar{\theta}$

Acronyms

AA	Average Accuracy
BCD	Block Coordinate Descent
CRB	Cramér-Rao bound
EEG	Electroencephalography
ICRB	Intrinsic Cramér-Rao bound
KL	Kullback-Leibler divergence
MEG	Magnetoencephalography
MLE	Maximum Likelihood Estimator
MSE	Mean Squared Error
MSG	Mixture of Scaled Gaussian distributions
NC-MSG	Non-Centered Mixture of Scaled Gaussian distributions
NLL	Negative Log Likelihood
OA	Overall Accuracy
PCA	Principal Component Analysis
PDF	Probability Density Function
PPCA	Probabilistic Principal Component Analysis
SAR	Synthetic Aperture Radar
SCM	Sample Covariance Matrix
SNR	Signal-to-Noise Ratio
SVD	Singular Value Decomposition
UAV	Unmanned Aerial Vehicle
WCSS	Within-Cluster Sum of Squares
WGN	White Gaussian Noise

1 - Statistical learning for time series

This chapter introduces the framework of this manuscript. It presents the basics of the concepts discussed in the next chapters starting from the *Earth observation* and its growing number of applications, then addressing some notions of statistics and machine learning and ending with the incorporation of the theory of Riemannian geometry in all these problematics. It should be noted that the Riemannian geometry is only briefly discussed at the end of this chapter and that the next chapter (Chapter 2) is entirely dedicated to it.

First of all, we present *multivariate image time series* for Earth observation in Section 1.1. The use of these data is motivated and *multispectral imagery* is presented. Furthermore, two datasets of classification are introduced: *Indian Pines* [9] and *Breizhcrops* [118]. Then, a *clustering/classification pipeline* is detailed in Section 1.2. It aims at the clustering/classification of *multivariate image time series* and is composed of 3 steps: vectors extraction, features estimation and features clustering/classification. Section 1.3 introduces the basics of the feature estimation step with some central definitions to statistics such as the *maximum likelihood estimators*. Section 1.4 presents two standard machine learning algorithms [64]: the *K-means++* [7] for clustering and the *Nearest centroïd classifier* for classification. This chapter finishes with the motivation of the usage of *Riemannian geometry* for the presented clustering/classification pipeline. Indeed, some statistical features lie on non-Euclidean spaces called *Riemannian manifolds* and their curvatures can be taken into account in the clustering/classification pipeline.

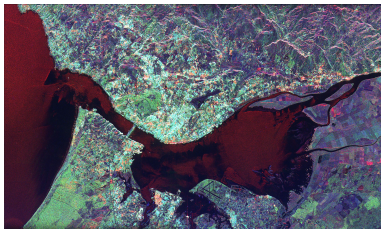
1.1 . Earth observation and datasets

1.1.1 . Earth observation and multispectral imagery

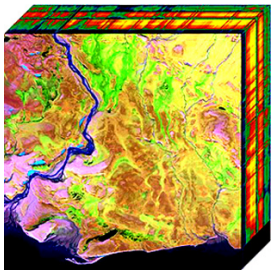
Earth observation provides a unique way of gathering informations about our planet. For this purpose, many remote sensing instruments have been developed and deployed in recent years. They are the cornerstone of many applications such as monitoring the evolution of our environment (*e.g.* glaciers, forests, urbanism), major events (*e.g.* earthquakes, floods), human activity (*e.g.* maritime and borders surveillance) as well as weather forecasting. Examples of these remote sensing instruments are

- *Synthetic Aperture Radar (SAR) satellites*: TerraSAR-X [1], Sentinel

[1] https://earth.esa.int/eogateway/missions/terrasar-x-and-tandem-x



(a) Example of a SAR image (from nasa.gov).



(b) Example of a multispectral image (from nasa.gov).

Figure 1.1: Different types of earth observation data: SAR and multispectral images.

1 [2], Capella [3], and ICEYE [4],

- *airborne radars*: UAVSAR [5],
- *spectrometer satellites*: Sentinel 5P,
- *altimeter satellites*: Sentinel 6A,
- and *multispectral satellites*: Sentinel 2 and 3, and landsat [6].

Figure 1.1 illustrates these sensors with two images: a SAR image and a multispectral image.

In the rest of the manuscript, we focus on multispectral imagery due to the availability of *annotated data*, *i.e.* data which come with ground truths. However the different developed methods along the chapters also apply to other types of data such as radar imagery.

Classical digital cameras measure the solar radiation reflected on given surfaces for three different wavelengths of the electromagnetic spectrum. These wavelengths correspond to three colors: Red, Green and Blue (RGB). Each pixel of such an image contains the three values of radiances associated with these colors. Hence, an image is stored as a datacube, also called tensor, $\boldsymbol{X} \in \mathbb{R}^{w \times h \times 3}$ where w and h are the number of pixels of width and height respectively. Multispectral imagery [59] proposes to extend this process by measuring radiances across many more different wavelengths. Thus, each pixel contains as much values as the number p of considered wavelengths and thus a multispectral image is stored as a datacube $\boldsymbol{X} \in$

[2] https://sentinels.copernicus.eu/web/sentinel/home
[3] https://www.capellaspace.com
[4] https://www.iceye.com
[5] https://uavsar.jpl.nasa.gov
[6] https://landsat.gsfc.nasa.gov

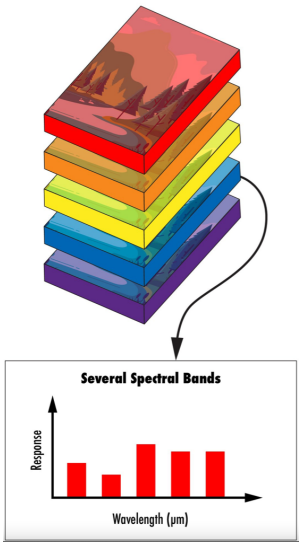


Figure 1.2: Multispectral imagery measures the radiance across different wavelengths of the electromagnetic spectrum. A multispectral image is the concatenation of these measurements. (Figure from www.edmundoptics.jp)

$\mathbb{R}^{w\times h\times p}$. The interest of considering many wavelengths is to enable a fine analysis of what is on the ground. A classic example of the use of these bands is the *normalized difference vegetation index* (NDVI) which is $\frac{\rho_{\text{NIR}}-\rho_{\text{VIS}}}{\rho_{\text{NIR}}+\rho_{\text{VIS}}}$ with ρ_{NIR} being the reflectance measured in the near infrared ($\sim 800nm$) and ρ_{VIS} the reflectance measured in the visible ($\sim 600nm$). This index is correlated with the physical properties of the vegetation canopy such as the biomass or the fractional vegetation recover [35]. Thus, it provides valuable informations on the vegetation canopy. An illustration of a multispectral image is presented in Figure 1.2. In this figure, five wavelengths are measured and thus each pixel contains five values.

Multispectral images of simple scenes can be modeled with mixing models [77]. Indeed, in multispectral imagery, it is assumed that $s \leq p$ materials, also called *endmembers*, (*e.g.* water, grass, wood) constitute the observed scene and that there is no interactions between endmembers. The last assumption means that any given package of incident radiation interacts only with one endmember (*e.g.* a light beam reflects on a piece of wood and then hits the multispectral sensor). Thus a pixel $\boldsymbol{x}_i \in \mathbb{R}^p$ is the linear combination, also called linear mixing, of the endmembers spectra with coefficients equal to the proportions of the areas covered by the endmembers. These coefficients are called *fractional abundances*. Mathematically, this linear mixing model writes

$$\boldsymbol{x}_i = \boldsymbol{A}\boldsymbol{w}_i + \boldsymbol{n}_i \tag{1.1}$$

where $\boldsymbol{A} \in \mathbb{R}^{p \times s}$ is the full rank mixing matrix whose columns are the spectra of the endmembers, $\boldsymbol{w}_i \in \Delta^s = \left\{ (t_1, \cdots, t_s)^T \in \mathbb{R}^s : t_i \geq 0, \sum_{i=1}^s t_i = 1 \right\}$ contains the fractional abundances, and $\boldsymbol{n}_i$ is an observation additive noise. The vectors $\boldsymbol{w}_i$ are important for classification problems since they contain abundances of the endmembers which are closely related to the classes. However, in practice, $\boldsymbol{A}$, $\boldsymbol{w}_i$ and $\boldsymbol{n}_i$ are unknown. Thus, an approach is to estimate $\boldsymbol{A}$ and $\boldsymbol{w}_i$ such that $\boldsymbol{x}_i \approx \boldsymbol{A}\boldsymbol{w}_i$, *e.g.* see [66, 12]. Another approach is to develop machine learning algorithms that are invariant to mixing models. Indeed, divergence-based algorithms, such as the *K-means++* algorithm, can be equipped with affine invariant divergences. These divergences measure the proximity between hidden vectors $\boldsymbol{w}_i$ by measuring the proximity between observed vectors $\boldsymbol{x}_i$. This point of view is developed in Section 1.5.

1.1.2 . Multivariate image time series

So far, we have presented Earth observation images. In practice, the same area of the Earth is revisited regularly by the same satellite. Indeed, the travels of a satellite are cyclic. The time elapsed between two visits of the same place is called the revisit time and corresponds to the time elapsed for the satellite to complete one cycle. For example, the Sentinel 2A satellite has a revisit time of ten days: it takes ten days to complete a full cycle. This revisit time can be shortened by using multiple satellites. For example, Sentinel 2A and 2B together have a revisit time of five days. Thus, several images of the same area are taken over time with a given frequency and each image has several measurements (*e.g.* different wavelengths in multispectral imagery). This is called a *multivariate image time series* and is stored as a tensor $\boldsymbol{X}^{T \times w \times h \times p}$ where T is the number of dates/images. These time series are rich since they contain three diversities: the temporal, the spatial, and the sensor diversity (measured wavelengths for multispectral imagery, polar for SAR). A scheme of a multivariate image time series is represented in Figure 1.3. In the following, we take advantage of these data to propose solutions to some of the applications mentioned earlier.

Earlier, we presented many applications that leverage Earth observation. Here, we focus on applications that can be casted as K-class *segmentation problems*. In its general form, the problem we consider is the following: a tensor $\boldsymbol{X}$ of pixels is available and we must predict a label in $[\![1, K]\!]$ for each pixel. In Figure 1.3, the tensor $\boldsymbol{X}$ is the whole time series. Classes are any discrete and non-ordered valuable informations for a given application. For crop type mapping, examples of labels are corn, wheat and meadow. We refer to a classification problem when a part of $\boldsymbol{X}$ has already been labeled, called a *training set*, and only the remaining part, called the *test set*, must be segmented (*supervised learning*). A clustering problem is when $\boldsymbol{X}$ is not at all labeled and thus there is only a test set (*unsupervised learning*). Thus pixels in $\boldsymbol{X}$ must be partitioned into K sets. In practice, the test

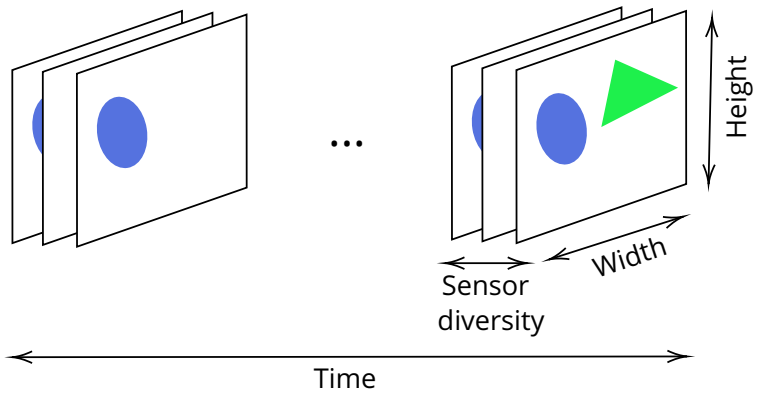


Figure 1.3: Scheme of a multivariate image time series. Several images of a same area on the Earth are taken over time. Each image has several measurements ("Sensor diversity" in the scheme).

set is also labeled but these labels are kept hidden until the evaluation of a proposed solution to the clustering/classification problem. The association of $\boldsymbol{X}$ and labels is called a *dataset*. In the following, we present two datasets of multispectral imagery.

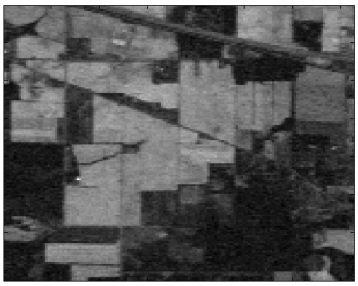
1.1.3 . Datasets: *Indian Pines* and *Breizhcrops*

The first dataset is called *Indian Pines* [9] and is a $w \times h = 145 \times 145$ pixels hyperspectral image. This image consists of $p = 200$ spectral bands in the wavelength range 0.4-$2.5\mu m$. The task is to segment it into $K = 16$ classes without training data: it is a clustering problem. Figure 1.4 shows the image as well as the ground truth. Table 1.1 in Appendix 1.A.1 gives the classes names as well as the number of samples per class. In practice, we apply a sliding window of size $w_s \times w_s$ before doing the clustering. Thus, mathematically, the task is to cluster a datacube $\boldsymbol{X} \in \mathbb{R}^{M \times p \times n}$, where $M = w \times h$ and $n = w_s \times w_s$, into K clusters. The clustering is represented by a vector in $[\![1, K]\!]^M$. By reshaping this vector into a $w \times h$ matrix, we get a segmentation map as in Figure 1.4b.

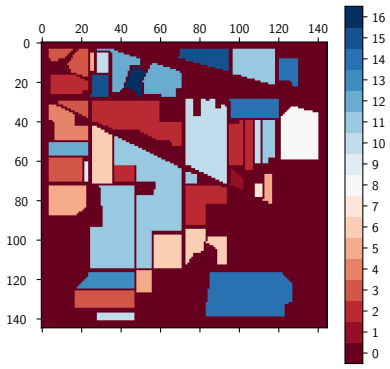
The second dataset is called *Breizhcrops* [118] [7]. This crop type mapping dataset gathers more than $600,000$ time series to classify[8]. These data have been measured with the Sentinel-2 satellite from January 1, 2017 to December 31, 2017 across the whole region of *Brittany*, *France* (see the maps of Figure 1.5). Each time series has $p = 13$ spectral bands, a length of $n = 45$ and belongs to one of the $K = 9$ classes presented in Table 1.2 in Appendix 1.A.2. The dataset is geographically split into a training set

[7] https://breizhcrops.org/

[8] The *Breizhcrops* dataset is composed with two processing levels. Here, we use the raw reflectances at the top-of-atmosphere (level 1C).

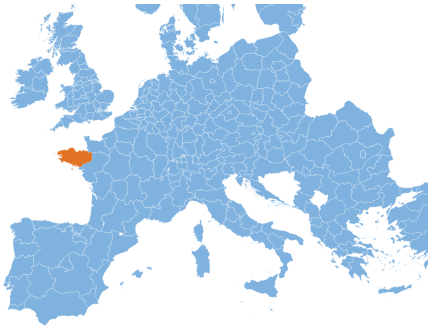


(a) Magnitude of the *Indian Pines* image.

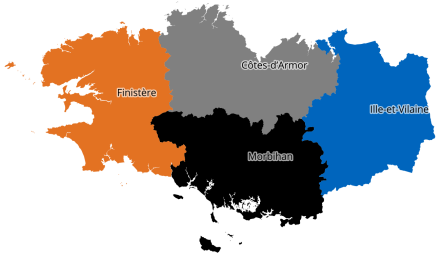


(b) Ground truth. The background (no class available) is represented by the class 0.

Figure 1.4: *Indian Pines*: a multispectral dataset [9].

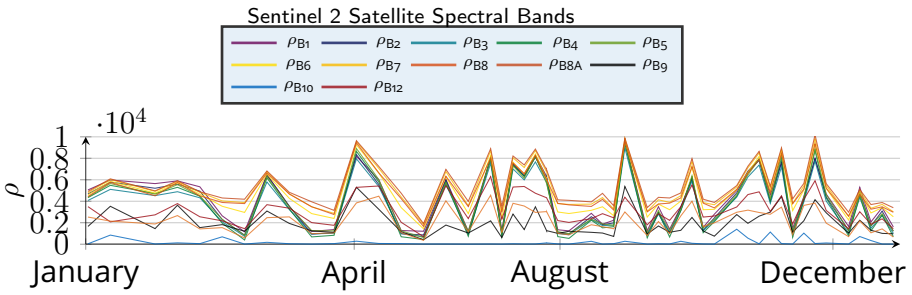


(a) Map of the Europe with the Brittany region in orange.

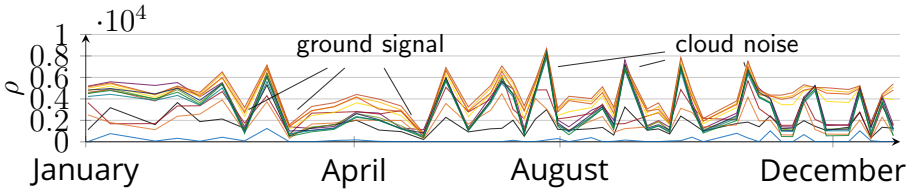


(b) Map of the Brittany region with its four departments.
Training set: Finistère, Côtes-d'Armor, and Ille-et-Vilaine.
Test set: Morbihan.

Figure 1.5: The *Breizhcrops* dataset [118] is a time series dataset that have been measured across the whole region of Brittany, France. Three departments of this region are used to construct the training set and the remaining one constitutes the test set. Figure courtesy [118].



(a) A time series of meadows.



(b) A time series of corn.

Figure 1.6: Reflectances ρ of two different time series. Figure courtesy [118].

of M_{train} time series from three departments of Brittany and a test set of M_{test} time series from the remaining department. Two time series from two different classes are presented in Figure 1.6. These illustrate the importance of the temporal dimension for classification. Indeed, we observe that the two time series are very close from January to March even though they belong to two different classes. If measurements are made only at the beginning of the year, it is difficult to classify them, whereas measurements from April onwards allow us to differentiate them. Thus, mathematically, the task is to train a classifier on the training datacube $\boldsymbol{X}_{\text{train}} \in \mathbb{R}^{M_{\text{train}} \times p \times n}$ with the vector of labels in $[\![1, K]\!]^{M_{\text{train}}}$ and then to predict the labels of the test datacube $\boldsymbol{X}_{\text{test}} \in \mathbb{R}^{M_{\text{test}} \times p \times n}$. The prediction takes the form of a vector in $[\![1, K]\!]^{M_{\text{test}}}$.

1.2 . Clustering and classification pipeline

In this section, the objective is to address segmentation problems that arise as depicted in the previous section. To do so, we present a clustering/classification pipeline, illustrated in Figure 1.7. We emphasis that it applies to both clustering and classification problems. This pipeline decomposes into three steps:

1. *vectors extraction*,
2. *features estimation*,

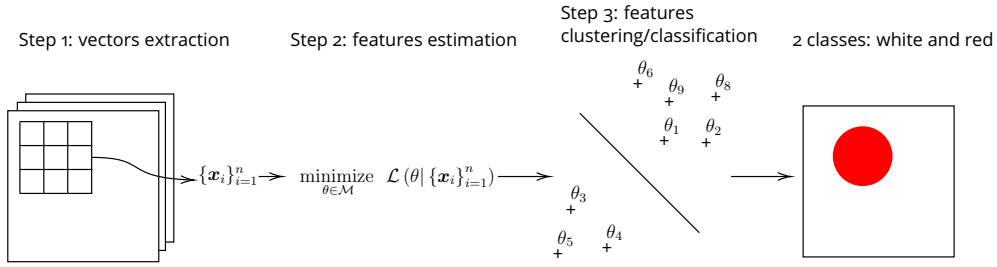


Figure 1.7: Clustering-classification pipeline.

3. *features clustering/classification*.

This pipeline is meant to be general and applies to many data. Here, we focus on data that presents as the datasets of Subsection 1.1.3.

1.2.1 . Vectors extraction

The first step, called vectors extraction, consists in extracting data batches $\{\boldsymbol{x}_i\}_{i=1}^n \subset \mathbb{R}^p$ of the same cardinality n. Sometimes, this step is direct as for the *Breizhcrops* dataset since it has already been pre-processed. Indeed, the latter is presented as $\boldsymbol{X}_{\text{train}} \in \mathbb{R}^{M_{\text{train}}\times p\times n}$ and $\boldsymbol{X}_{\text{test}} \in \mathbb{R}^{M_{\text{test}}\times p\times n}$. Thus, M_{train} and M_{test} data batches are easily extracted for the training and test sets respectively. The batches correspond to time series; it is a *temporal extraction*. In other cases, an extraction must be explicitly achieved as in the *Indian Pines* dataset. Indeed, and as explained in Subsection 1.1.3, a $w_s \times w_s$ sliding window is applied to get $\boldsymbol{X} \in \mathbb{R}^{M\times p\times n}$ with $n = w_s \times w_s$. This way $M = h \times w$, with h and w being the height and the width of the image respectively, data batches are extracted. In this case, we performed a *spatial extraction*.

1.2.2 . Features estimation

The second step of the pipeline is the features estimation. Once we have data batches from the previous step, we estimate features from them. Indeed, the use of statistical descriptors is classical in machine learning since they are often more discriminant than raw data; *e.g.* see [8, 134, 133]. Thus, each batch of raw data $\{\boldsymbol{x}_i\}_{i=1}^n$ is transformed into a feature θ. This estimation is written as a minimization problem of a loss function $\mathcal{L} : \mathcal{M} \to \mathbb{R}$ over a given set $\mathcal{M}$, possibly constrained,

$$\underset{\theta\in\mathcal{M}}{\text{minimize}}\ \mathcal{L}(\theta|\{\boldsymbol{x}_i\}_{i=1}^n). \tag{1.2}$$

It should be noted that θ can take many different forms: it can be a vector, a vector and a covariance matrix, a subspace, and so on ... Thus, this step is general and encompasses many classical algorithms such as the Principal Component Analysis (PCA), the Sample Covariance Matrix (SCM), etc ... As

done for the previous step, we present the implications the feature estimation step has on the two datasets *Indian Pines* and *Breizhcrops*. The previous step extracts batches $\{\boldsymbol{x}_i\}_{i=1}^n$ spatially for the *Indian Pines*. This means that this features estimation step transforms local patches of the image into features. Thus, each pixel is characterized by one feature θ. For the *Breizhcrops* dataset, each batch corresponds to one time series. Hence, the feature estimation step transforms each time series into a feature θ. Hence, in both datasets, the datacube $\boldsymbol{X} \in \mathbb{R}^{M\times p\times n}$ becomes a set of features $\{\theta_i\}_{i=1}^M$. The formulation (1.2) is discussed with more details in Section 1.3, which is dedicated to statistical estimation. Finally, we emphasis that this features estimation step could be applied to many other datasets. The only requirement is that to have batches of data $\{\boldsymbol{x}_i\}_{i=1}^n$.

1.2.3 . Features clustering/classification

Once we have a set of features $\{\theta_i\}_{i=1}^M$, it remains to cluster/classify it. This is the third step: the features clustering/classification step. We distinguish two cases: clustering and classification.

In the first case, the objective is to partition $\{\theta_i\}_{i=1}^M$ into K sets in an unsupervised manner, *i.e.* without informations about the desired classes. In practice this is achieved by grouping features θ_i that are close to each other using, for example, the *K-means++* algorithm [7] presented in Section 1.4. The *Indian Pines* dataset can be used as a clustering dataset. Indeed, clustering the features $\{\theta_i\}_{i=1}^M$ into K sets gives a clustering of the pixels since each θ_i is associated to one pixel $\boldsymbol{x}_i$.

In the second case, the set of features $\{\theta_i\}_{i=1}^M$ is divided into two non-overlapping sets: a training set $\{\theta_i\}_{i=1}^{M_{\mathsf{train}}}$ with labels $\{y_i\}_{i=1}^{M_{\mathsf{train}}}$ and a test set $\{\theta_i\}_{i=1}^{M_{\mathsf{test}}}$. The goal is to classify the test set by leveraging the training that provides informations on the classes. An example of a classifier is the *Nearest centroïd classifier* presented in Section 1.4. Briefly, it computes the center of mass, sometimes called the mean, of each class in the training set. Then, it classifies the test set by searching the closest the center of mass of a given point. On the *Breizhcrops* dataset, we recall that each feature θ_i is associated with a time series. Thus, a classifier learns to classify time series with the training set. Then, it infers the labels of the features from the test set. By doing so, we get a classification of the time series of original test set.

1.3 . Statistical models and their features

In Section 1.2, we presented the clustering/classification pipeline from Figure 1.7. Its second step performs statistical estimation. The objective of this section is to go more into details on the estimation theory. We mention

the essential concepts to understand the following sections and chapters of the manuscript. For a complete presentation of the topic, the reader is referred to the book [76].

1.3.1 . Some reminders on the estimation theory

In this section, we recall some classical definitions and results from the estimation theory. Given a *measurement* $\{\boldsymbol{x}_i\}_{i=1}^n$ in the *sample space* $\mathcal{X}$, we seek a *parameter* θ in the *parameter space* $\mathcal{E}$, a linear space (*e.g.* $\mathbb{R}^q$, the set of symmetric matrices, ...). Indeed, we assume that samples follow a statistical distribution for which a *probability density function* (PDF) exists. The latter depends on the parameter θ which is assumed to be a discriminant feature for a given application. Therefore, θ must be estimated. To motivate the introduction of the estimation theory, we point out that many problems can be written as estimation problems. They typically arise in many signal processing and machine learning topics such as change detection [91, 90], dimensionality reduction [131], graphical model estimation [57] and clustering [67]. Before going any further, it is worth noting that the parameter space $\mathcal{E}$ is a q-dimensional linear space. We endow it with the Euclidean inner product $\langle \theta_1, \theta_2 \rangle^{\mathcal{E}} = \mathrm{vec}(\theta_1)^T \mathrm{vec}(\theta_2)$ and the Euclidean distance $d_{\mathcal{E}}(\theta_1, \theta_2) = \|\mathrm{vec}(\theta_1) - \mathrm{vec}(\theta_2)\|_2$ where $\mathrm{vec} : \mathcal{E} \to \mathbb{R}^q$ vectorizes the input by stacking its coordinates into a vector. It should be noted that the definitions and results stated in this subsection are extended to possibly non-linear spaces $\mathcal{M}$ in Chapter 2, Section 2.5.

In practice, an *estimate* $\hat{\theta}$ of θ is produced from the measurement $\{\boldsymbol{x}_i\}_{i=1}^n$. The corresponding mapping from $\mathcal{X}$ to $\mathcal{E}$ is called an *estimator*.

Definition 1. *An estimator $\hat{\theta} : \mathcal{X} \to \mathcal{E}$ maps every measurement $\{\boldsymbol{x}_i\}_{i=1}^n$ to an estimate $\hat{\theta}(\{\boldsymbol{x}_i\}_{i=1}^n)$.*

It should be noted that, as mentioned earlier, the estimators presented in this section are associated with statistical distributions. However, this is not mandatory. Indeed, estimators can be defined without assuming that data follow a given statistical distribution, *e.g.* see the M-estimators [86, 71]. Then, a central tool to the estimation theory is the *negative log-likelihood* (NLL) function. Let a measurement $\{\boldsymbol{x}_i\}_{i=1}^n \in \mathcal{X}$ be a realization of a random variable X following a PDF f parametrized by $\theta \in \mathcal{E}$, *i.e.*

$$X \sim f(.; \theta), \tag{1.3}$$

then, the NLL $\mathcal{L}$ is defined as minus the logarithm of f.

Definition 2. *Given $\{\boldsymbol{x}_i\}_{i=1}^n \in \mathcal{X}$ and a PDF such that $\forall \theta \in \mathcal{E}$ $f(\{\boldsymbol{x}_i\}_{i=1}^n; \theta) > 0$, the negative log-likelihood (NLL) function $\mathcal{L} : \mathcal{E} \to \mathbb{R}$ is defined by*

$$\mathcal{L}(\theta | \{\boldsymbol{x}_i\}_{i=1}^n) = -\log f(\{\boldsymbol{x}_i\}_{i=1}^n; \theta).$$

In the rest of the subsection, we assume that the NLL is at least twice differentiable, which is verified for usual distributions. The definition of the NLL on $\mathcal{E}$ enables the derivation of an estimator of the true parameter θ called the *maximum likelihood estimator* (MLE). The intuition is to find the PDF $\theta \mapsto f(\{\boldsymbol{x}_i\}_{i=1}^n; \theta)$ that is the most likely to produce the measurement $\{\boldsymbol{x}_i\}_{i=1}^n \in \mathcal{X}$. The MLE realizes this maximization. By recalling that maximizing $\theta \mapsto f(\{\boldsymbol{x}_i\}_{i=1}^n; \theta)$ is equivalent to minimizing $\theta \mapsto -\log f(\{\boldsymbol{x}_i\}_{i=1}^n; \theta)$, we get the following definition.

Definition 3. *Given $\{\boldsymbol{x}_i\}_{i=1}^n \in \mathcal{X}$, the MLE $\hat{\theta} \in \mathcal{E}$ is the minimizer of the NLL function*

$$\hat{\theta} = \arg\min_{\theta \in \mathcal{E}} \ \mathcal{L}(\theta | \{\boldsymbol{x}_i\}_{i=1}^n).$$

Once an estimator is defined, the question of its performance arises. To answer this question, the theory of *Cramér-Rao bounds* (CRB) has been developed [47, 116]. Indeed, the latter lower bounds, in the Loewner sense, the covariance matrices of estimators for a given statistical problem. Thus, inequalities are derived and estimators are compared to these lower bounds. In the following, the estimator $\hat{\theta}$ is seen as a random variable. Indeed, each new measurement $\{\boldsymbol{x}_i\}_{i=1}^n$ is associated with a new value for the estimator. Thus statistics of $\hat{\theta}$ can be computed. We begin by defining the *bias* of an estimator.

Definition 4. *The bias of an estimator $\hat{\theta} \in \mathcal{E}$ for a given parameter $\theta \in \mathcal{E}$ is the mean error vector*

$$\boldsymbol{b}_\theta = \mathbb{E}\left[\mathrm{vec}(\hat{\theta}) - \mathrm{vec}(\theta)\right].$$

An estimator is unbiased if its bias is zero everywhere

$$\boldsymbol{b}_\theta = \boldsymbol{0} \text{ for all } \theta \in \mathcal{E}.$$

For simplicity, in the following, we focus on unbiased estimators. However, the following definitions can be extended to biased estimators. Then, the covariance matrix of an unbiased estimator is presented. To do so, we recall the definitions of two sets. The sets of $q \times q$ symmetric matrices and $q \times q$ symmetric positive semidefinite matrices are defined as

$$\mathcal{S}_q = \left\{\boldsymbol{\Sigma} \in \mathbb{R}^{q \times q} : \boldsymbol{\Sigma}^T = \boldsymbol{\Sigma}\right\}, \tag{1.4}$$

and

$$\mathcal{S}_q^+ = \left\{\boldsymbol{\Sigma} \in \mathcal{S}_q : \forall \boldsymbol{x} \in \mathbb{R}^q, \, \boldsymbol{x}^T \boldsymbol{\Sigma} \boldsymbol{x} \geq 0\right\} \tag{1.5}$$

respectively.

Definition 5. *For an unbiased estimator $\hat{\theta}$, the covariance matrix $\boldsymbol{C}_\theta \in \mathbb{R}^{q\times q}$ is a symmetric, positive semidefinite matrix given by*

$$\boldsymbol{C}_\theta = \mathbb{E}\left[(\mathrm{vec}(\hat{\theta}) - \mathrm{vec}(\theta))(\mathrm{vec}(\hat{\theta}) - \mathrm{vec}(\theta))^T\right].$$

From Definition 5, the trace of $\boldsymbol{C}_\theta$ is the variance of the estimator $\hat{\theta}$

$$\mathrm{Tr}(\boldsymbol{C}_\theta) = \mathbb{E}\left[d^2_{\mathcal{E}}(\hat{\theta},\theta)\right] = \mathbb{E}\left[||\,\mathrm{vec}(\hat{\theta}) - \mathrm{vec}(\theta)||_2^2\right]. \tag{1.6}$$

This variance is also sometimes referred as the *Mean Squared Error* (MSE) since it measures the quadratic error between the estimator $\hat{\theta}$ and the true parameter θ in average. A last definition in this subsection is the *Fisher information matrix*. This matrix is leveraged to derive CRBs.

Definition 6. *The Fisher information matrix $\boldsymbol{F}_\theta$ is the $q \times q$ symmetric, positive semidefinite matrix whose entries are given by*

$$(\boldsymbol{F}_\theta)_{ij} = \mathbb{E}\left[\frac{\partial\mathcal{L}(\theta|\{\boldsymbol{x}_i\}_{i=1}^n)}{\partial\theta^i}\frac{\partial\mathcal{L}(\theta|\{\boldsymbol{x}_i\}_{i=1}^n)}{\partial\theta^j}\right] = \mathbb{E}\left[\frac{\partial^2\mathcal{L}(\theta|\{\boldsymbol{x}_i\}_{i=1}^n)}{\partial\theta^i\partial\theta^j}\right]$$

where $\mathcal{L}$ is the NLL from Definition 2, $\frac{\partial_\cdot}{\partial\theta^i}$ the partial derivative with respect to the i^{th} coordinate of θ and $\frac{\partial^2_\cdot}{\partial\theta^i\theta^j}$ the second partial derivative with respect to the i^{th} and j^{th} coordinates of θ.

With the tools defined previously, we are now able to present the main theorem of CRBs for unbiased estimators.

Theorem 1. *Let $\theta \in \mathcal{E}$ and consider an estimation problem on $\mathcal{E}$ such that the Fisher information matrix $\boldsymbol{F}_\theta$ is invertible. Then, for any unbiased estimator, the covariance matrix $\boldsymbol{C}_\theta$ obeys the following matrix inequality*

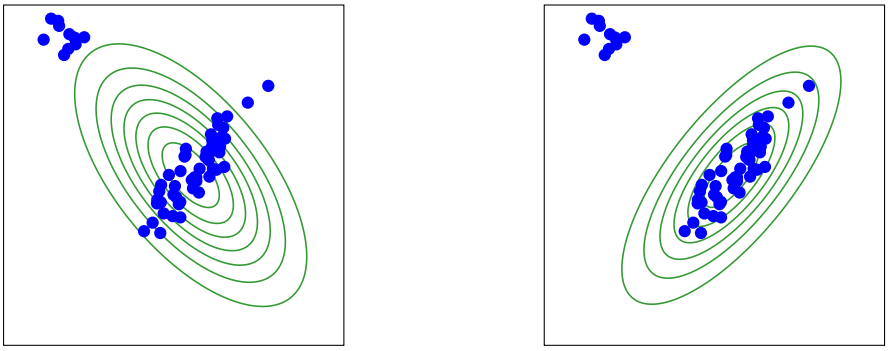
$$\boldsymbol{C}_\theta \succeq \boldsymbol{F}_\theta^{-1}$$

where $\succeq$ is the Loewner inequality, i.e. $\boldsymbol{C}_\theta - \boldsymbol{F}_\theta^{-1} \in \mathcal{S}_q^+$.

From Theorem 1, we get the following CRB

$$\mathbb{E}\left[d^2_{\mathcal{E}}(\hat{\theta},\theta)\right] = \mathbb{E}\left[||\,\mathrm{vec}(\hat{\theta}) - \mathrm{vec}(\theta)||_2^2\right] \geq \mathrm{Tr}\left(\boldsymbol{F}_\theta^{-1}\right). \tag{1.7}$$

In general, the MLEs are consistent, *i.e.* they tend to the true parameter when the number of measurements tends to the infinity. Also they are asymptotically unbiased and efficient, *i.e.* they reach the CRB of the estimation problem when the number of measurements tends to the infinity. This means that these estimators are asymptotically optimal which justifies their use. In practice, the convergence of the MLE towards their CRBs is fast and thus the optimal variance is reached for reasonable numbers of samples. The following theorem is restricted to independent and identically distributed samples however it can be extended to other cases; see [76, Chapter 7].



(a) SCM (b) Tyler's M-estimator

Figure 1.8: Set of data points $\{\boldsymbol{x}_i\}_{i=1}^n \subset \mathbb{R}^2$ with outliers. The sample covariance matrix (SCM), from Proposition 1, is biased towards the outliers whereas the Tyler's M-estimator, from Proposition 3, is robust to them.

Theorem 2. *Assuming that the samples from the measurement $\{\boldsymbol{x}_i\}_{i=1}^n$ are independent realizations of an identical random variable of PDF f satisfying some mild regularity conditions described in [76, Chapter 7], then the MLE $\hat{\theta}$ of the unknown parameter θ is asymptotically distributed according to*

$$\sqrt{n}\left(\mathrm{vec}(\hat{\theta}) - \mathrm{vec}(\theta)\right) \xrightarrow[n\to\infty]{} \mathcal{N}(\mathbf{0}, \boldsymbol{F}_\theta^{-1})$$

where $\boldsymbol{F}_\theta$ is the Fisher information matrix of a single sample $\boldsymbol{x}_i$ evaluated at the true value of the unknown parameter, i.e. $(\boldsymbol{F}_\theta)_{ij} = \mathbb{E}\left[\frac{\partial \mathcal{L}(\theta|\boldsymbol{x}_i)}{\partial \theta^i}\frac{\partial \mathcal{L}(\theta|\boldsymbol{x}_i)}{\partial \theta^j}\right] = \mathbb{E}\left[\frac{\partial^2 \mathcal{L}(\theta|\boldsymbol{x}_i)}{\partial \theta^i \partial \theta^j}\right]$.

1.3.2 . Gaussian distribution and Tyler's M-estimator

We presented some of the basics of the estimation theory in the previous subsection. Two standard estimators are presented in this subsection. In signal processing and machine learning, a classical statistical model is the multivariate Gaussian distribution. Indeed, signal and noise are often modeled with this distribution thanks to the Central Limit Theorem. The latter states that the sum of n independent and identically distributed random variables with finite mean and variance converge to a Gaussian distribution as $n \to \infty$. Furthermore, classical classification algorithms, such as the linear discriminant analysis, assume that data are Gaussian. This assumption gives simple closed form formula for the classification rules and good performance in practice. The Gaussian distribution is parametrized by the location $\boldsymbol{\mu} \in \mathbb{R}^p$ and the covariance matrix $\boldsymbol{\Sigma} \in \mathcal{S}_p^{++}$ where $\mathcal{S}_p^{++}$ is the set of $p \times p$ symmetric positive definite matrices

$$\mathcal{S}_p^{++} = \left\{\boldsymbol{\Sigma} \in \mathcal{S}_p : \forall \boldsymbol{x} \neq \mathbf{0} \in \mathbb{R}^p,\, \boldsymbol{x}^T \boldsymbol{\Sigma} \boldsymbol{x} > 0\right\}. \tag{1.8}$$

To make the transition with the previous sections, here the feature is $\theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $\mathcal{M}$ is the non-linear space $\mathbb{R}^p \times \mathcal{S}_p^{++}$ and $\mathcal{E}$ is the Euclidean space $\mathbb{R}^p \times \mathbb{R}^{\frac{p(p+1)}{2}}$ that contains the location and the upper triangular part of the covariance matrix. Then, the formal definition of the Gaussian distribution is given.

Definition 7. *A random vector $\boldsymbol{x} \in \mathbb{R}^p$ follows a multivariate Gaussian distribution if its PDF writes*

$$f_G(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{p}{2}} \ |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

where $\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{x}] \in \mathbb{R}^p$ is the location and $\boldsymbol{\Sigma} = \mathbb{E}\left[(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^T\right] \in \mathbb{R}^{p\times p}$ is the covariance matrix. In this case, we write $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

It should be noted that the PDF of the Gaussian distribution is entirely determined by the location and covariance matrix. The latter are easily estimated using the MLEs which solve the following problem

$$\underset{(\boldsymbol{\mu},\boldsymbol{\Sigma})\in\mathbb{R}^p\times\mathcal{S}_p^{++}}{\text{minimize}} \left\{\mathcal{L}_{\mathsf{G}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\{\boldsymbol{x}_i\}_{i=1}^n) = \log|\boldsymbol{\Sigma}| + \frac{1}{n}\sum_{i=1}^n (\boldsymbol{x}_i-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu})\right\}. \tag{1.9}$$

Proposition 1. *Given a measurement $\{\boldsymbol{x}_i\}_{i=1}^n$, the MLEs of the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ of the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ are the sample mean and the sample covariance matrix[9] (SCM).*

$$\begin{cases} \hat{\boldsymbol{\mu}}_{SM} = \dfrac{1}{n}\displaystyle\sum_{i=1}^n \boldsymbol{x}_i \\ \hat{\boldsymbol{\Sigma}}_{SCM} = \dfrac{1}{n}\displaystyle\sum_{i=1}^n (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{SM})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{SM})^T. \end{cases}$$

From Proposition 1, the minimization problem (1.9) admits a solution if and only if the centered data matrix is full rank (or row full rank), *i.e.* $\operatorname{rank}([\boldsymbol{x}_1 - \hat{\boldsymbol{\mu}}_{\mathsf{SM}}, \cdots, \boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_{\mathsf{SM}}]) = p$. Indeed, if its not full rank, $\hat{\boldsymbol{\Sigma}}_{\mathsf{SCM}}$ only belongs to $\mathcal{S}_p^+$ and not to $\mathcal{S}_p^{++}$.

However, the Gaussian distribution is not always well suited. Indeed, the noise can be impulsive or data can include outliers such as mislabeled data in classification. In these cases, the Gaussian MLEs are biased towards these outliers and thus performance are deteriorated. Many statistical tools exist

[9] Sometimes, the SCM has $\frac{1}{n-1}$ factor instead of $\frac{1}{n}$ in order to be an unbiased estimator of the covariance matrix. In the rest of the manuscript, the SCM refers to the MLE of the covariance matrix of the Gaussian distribution, *i.e.* has a $\frac{1}{n}$ factor.

to remediate to this problem such as the elliptically contoured distributions (complex elliptically symmetric distribution for complex data) [33, 104] or the M-estimators [86, 71]. Here, we present non-centered mixtures of scaled Gaussian distributions (NC-MSG), *i.e.*

$$\boldsymbol{x}_i \overset{d}{=} \boldsymbol{\mu} + \sqrt{\tau_i}\boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{n}_i \tag{1.10}$$

with $\overset{d}{=}$ means "equal in distribution" (same cumulative distribution functions), $\tau_i > 0$ is the sample dependent scale (sometimes called deterministic texture), and $\boldsymbol{n}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_p)$ are independent. For example, this model has been successfully applied to radar imaging [91, 106] and radar detection [46, 107] to model the clutter.

Definition 8. *A set of independent random vectors $\{\boldsymbol{x}_i\}_{i=1}^n \subset \mathbb{R}^p$ follows a NC-MSG (also called compound Gaussian distribution with deterministic textures) if its PDF writes*

$$f_{\text{NC-MSG}}(\{\boldsymbol{x}_i\}_{i=1}^n; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\tau}) = \prod_{i=1}^n f_G(\boldsymbol{x}_i; \boldsymbol{\mu}, \tau_i\boldsymbol{\Sigma})$$

where $\boldsymbol{\mu}$ is the location, $\boldsymbol{\Sigma}$ is the scatter matrix, $\boldsymbol{\tau}$ contains the textures $\{\tau_i\}_{i=1}^n$ and f_G is the Gaussian PDF from Definition 7. In this case, we write $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \tau_i\boldsymbol{\Sigma})$.

Then, the NLL of the MSG is minimized to estimate its parameters,

$$\underset{(\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\tau})\in\mathbb{R}^p\times\mathcal{S}_p^{++}\times(\mathbb{R}_*^+)^n}{\text{minimize}} \sum_{i=1}^n \mathcal{L}_{\mathsf{G}}(\boldsymbol{\mu}, \tau_i\boldsymbol{\Sigma}|\{\boldsymbol{x}_i\}_{i=1}^n) \tag{1.11}$$

where $\mathcal{L}_{\mathsf{G}}$ is defined in (1.9). The solution of this problem satisfies a system stated in the next proposition.

Proposition 2. *Given a measurement $\{\boldsymbol{x}_i\}_{i=1}^n$, the MLEs $\hat{\boldsymbol{\mu}}$, $\hat{\boldsymbol{\Sigma}}$ and $\hat{\boldsymbol{\tau}}$ of the parameters of a NC-MSG $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \tau_i\boldsymbol{\Sigma})$ satisfy*

$$\begin{cases} \hat{\boldsymbol{\mu}} = \left(\displaystyle\sum_{i=1}^n \frac{1}{\hat{\tau}_i}\right)^{-1} \sum_{i=1}^n \frac{\boldsymbol{x}_i}{\hat{\tau}_i} \\ \hat{\boldsymbol{\Sigma}} = \dfrac{1}{n}\displaystyle\sum_{i=1}^n \frac{(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^T}{\hat{\tau}_i} \\ \hat{\tau}_i = \dfrac{1}{p}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^T\hat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}). \end{cases}$$

If the location parameter $\boldsymbol{\mu}$ is known and all $\boldsymbol{x}_i \neq \boldsymbol{\mu}$ then solving the system from Proposition 2 reduces to solving the following equation

$$\hat{\boldsymbol{\Sigma}} = \frac{p}{n} \sum_{i=1}^{n} \frac{(\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T}{(\boldsymbol{x}_i - \boldsymbol{\mu})^T \hat{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu})} \triangleq \mathcal{H}_{\mathsf{Ty}} \left(\hat{\boldsymbol{\Sigma}} \right) \tag{1.12}$$

with respect to $\hat{\boldsymbol{\Sigma}}$ and then computing the textures with

$$\hat{\tau}_i = \frac{1}{p} (\boldsymbol{x}_i - \boldsymbol{\mu})^T \hat{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}). \tag{1.13}$$

Fixed point Equation (1.12) has been extensively studied in [136, 108, 110, 104, 56] and the following proposition gives its solution.

Proposition 3. *If $n > p$ and for any p two by two distinct indices $i(1) < \cdots < i(p)$ chosen in $[\![1, n]\!]$ the centered data $\{\boldsymbol{x}_{i(j)} - \boldsymbol{\mu}\}_{j=1}^{p}$ are linearly independent then Equation* (1.12) *has a unique solution (up to a strictly positive scale factor). In this case, it is solved iteratively with the following iterates*

$$\hat{\boldsymbol{\Sigma}}^{(l+1)} = \mathcal{H}_{\mathit{Ty}} \left(\hat{\boldsymbol{\Sigma}}^{(l)} \right) \tag{1.14}$$

which converge in $\mathcal{S}_p^{++}$ for all initializations $\hat{\boldsymbol{\Sigma}}^{(1)} \in \mathcal{S}_p^{++}$. This estimator is called the Tyler's M-estimator.

We illustrate the robustness to outliers of the Tyler's M-estimator compared to the SCM in Figure 1.8. Unfortunately, when the location $\boldsymbol{\mu}$ is unknown, the system from Proposition 2 does not necessarily admit a solution. If it admits one, there is no guarantee that fixed point iterations converge to it. In practice, $\boldsymbol{\mu}$ is estimated with the sample mean $\hat{\boldsymbol{\mu}}_{\mathsf{SM}} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i$. Then, it is subtracted to the samples $\boldsymbol{x}_i$ before applying (1.14) to estimate the scatter matrix and (1.13) to estimate the textures.

1.3.3 . Regularized and low rank structure estimators

In the previous subsection, we presented the MLEs of the Gaussian distribution in Proposition 1. We said that the SCM is the MLE if only if the centered data matrix is full rank, *i.e.* $\operatorname{rank}([\boldsymbol{x}_1 - \hat{\boldsymbol{\mu}}_{\mathsf{SM}}, \cdots, \boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_{\mathsf{SM}}]) = p$. Otherwise, the SCM does not belong to $\mathcal{S}_p^{++}$ since its rank is strictly inferior to p. This problem arises when $n < p$, *i.e.* when the number of data is inferior to their dimension. This can easily happen in multi-spectral imagery since p can be several hundred whereas n is the number of data in a neighborhood of pixel and thus cannot exceed several dozens. Another problem is when the conditioning of the SCM is large; *i.e.* when the ratio of its largest eigenvalue over its lowest one is large. This is problematic since a small

perturbation in the eigenvalues heavily affects the output of algorithms using this estimator. Several approaches have been proposed in the literature to remediate to these problems. We present two of them: the *regularized estimators* and estimators with a *low rank structure*. It should be noted that all the presented reasoning can be applied to the Tyler's M-estimator. Some references on regularized and structured Tyler's M-estimators are [103, 126, 100] and [127, 17] respectively.

We begin with the regularized estimators which are estimators shrunk towards a target. This *shrinkage* is defined with the help of a *penalty*, also sometimes referred as a *regularization*. A classical example is the SCM shrunk towards the identity. Given $\beta \in [0, 1]$, we define the following optimization problem which is the minimization of the Gaussian NLL (1.9) with an additional penalty,

$$\underset{(\boldsymbol{\mu},\boldsymbol{\Sigma})\in\mathbb{R}^p\times\mathcal{S}_p^{++}}{\text{minimize}} \mathcal{L}_{\mathsf{G}}(\boldsymbol{\mu},\boldsymbol{\Sigma}|\{\boldsymbol{x}_i\}_{i=1}^n) + \beta \underbrace{\operatorname{Tr}\left(\boldsymbol{\Sigma}^{-1}\left[\frac{\operatorname{Tr}(\hat{\boldsymbol{\Sigma}}_{\mathsf{SCM}})}{p}\boldsymbol{I}_p - \hat{\boldsymbol{\Sigma}}_{\mathsf{SCM}}\right]\right)}_{\text{penalty}}. \tag{1.15}$$

The minimizer of this optimization problem is presented in the next proposition.

Proposition 4. *The minimizer of* (1.15) *is the sample mean and the SCM shrunk towards the identity*

$$\begin{cases} \hat{\boldsymbol{\mu}}_{SM} = \dfrac{1}{n}\displaystyle\sum_{i=1}^n \boldsymbol{x}_i, \\ \hat{\boldsymbol{\Sigma}} = (1-\beta)\hat{\boldsymbol{\Sigma}}_{SCM} + \beta\dfrac{\operatorname{Tr}(\hat{\boldsymbol{\Sigma}}_{SCM})}{p}\boldsymbol{I}_p. \end{cases}$$

From Proposition 4, the penalty from Equation (1.15) simply shrinks the eigenvalues of the SCM towards their mean. If $\beta = 0$, we recover the MLE of the Gaussian distribution. Otherwise, if $\beta \in]0, 1]$, Equation (1.15) admits a minimizer if and only if at least one centered data $\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{\mathsf{SM}}$ is not zero, *i.e.* $\operatorname{rank}([\boldsymbol{x}_1 - \hat{\boldsymbol{\mu}}_{\mathsf{SM}}, \cdots, \boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_{\mathsf{SM}}]) \geq 1$. Hence, this estimator does not require the centered data matrix to be of full rank as for the SCM. Furthermore, the conditioning of the estimator $\hat{\boldsymbol{\Sigma}}$ is improved since $\frac{\lambda_{\mathsf{max}}}{\lambda_{\mathsf{min}}} \to 1$ as $\beta \to 1$; where λ_{max} and λ_{min} are the maximum and minimum eigenvalues of $\hat{\boldsymbol{\Sigma}}$ respectively. We mention that various strategies to choose automatically the hyperparameter β have been proposed; *e.g.* see [78, 37, 101].

Then, we present an estimator with a low-rank structure derived in [131]. This estimator is the MLE of a statistical model that assumes that a Gaussian

signal is embedded in a white Gaussian noise (WGN). For all rank $k < \min\{p, n\}$, this model writes

$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{1.16}$$

where $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_{\text{LR}} + \sigma^2 \boldsymbol{I}_p$ with $\boldsymbol{\Sigma}_{\text{LR}} \in \mathcal{S}_p^+$, $\text{rank}(\boldsymbol{\Sigma}_{\text{LR}}) = k$ and $\sigma^2 > 0$. $\boldsymbol{\Sigma}_{\text{LR}}$ is the covariance of the signal whereas $\sigma^2 \boldsymbol{I}_p$ is the covariance of the noise. Then, the optimization problem to estimate the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ is the minimization of the Gaussian NLL while respecting the structure of the covariance matrix,

$$\begin{aligned} \underset{(\boldsymbol{\mu}, \boldsymbol{\Sigma}_{\text{LR}}, \sigma^2) \in \mathbb{R}^p \times \mathcal{S}_p^+ \times \mathbb{R}_*^+}{\text{minimize}} \quad & \mathcal{L}_{\text{G}}(\boldsymbol{\mu}, \boldsymbol{\Sigma} | \{\boldsymbol{x}_i\}_{i=1}^n) \\ \text{subject to} \quad & \boldsymbol{\Sigma} = \boldsymbol{\Sigma}_{\text{LR}} + \sigma^2 \boldsymbol{I}_p, \\ & \text{rank}(\boldsymbol{\Sigma}_{\text{LR}}) = k. \end{aligned} \tag{1.17}$$

The solution of (1.17) is given in the following proposition.

Proposition 5. *The minimizer of* (1.17) *is the sample mean and the SCM whose $p - k$ lowest eigenvalues have been averaged*

$$\begin{cases} \hat{\boldsymbol{\mu}}_{SM} = \dfrac{1}{n} \displaystyle\sum_{i=1}^n \boldsymbol{x}_i \\ \hat{\boldsymbol{\Sigma}} = \hat{\boldsymbol{\Sigma}}_{LR} + \hat{\sigma}^2 \boldsymbol{I}_p \end{cases}$$

where $\hat{\boldsymbol{\Sigma}}_{LR} = \boldsymbol{U}_k (\boldsymbol{\Lambda}_k - \hat{\sigma}^2 \boldsymbol{I}_k) \boldsymbol{U}_k^T$, $\hat{\sigma}^2 = \dfrac{1}{p-k} \displaystyle\sum_{i=1}^{p-k} (\boldsymbol{\Lambda}_{p-k})_{ii}$ *with the singular value decomposition (SVD) of the SCM, with eigenvalues in the descending order, denoted*

$$\hat{\boldsymbol{\Sigma}}_{SCM} \overset{SVD}{=} \left[\boldsymbol{U}_k, \boldsymbol{U}_{p-k}\right] \begin{pmatrix} \boldsymbol{\Lambda}_k & (0) \\ (0) & \boldsymbol{\Lambda}_{p-k} \end{pmatrix} \left[\boldsymbol{U}_k, \boldsymbol{U}_{p-k}\right]^T.$$

If k is chosen such as $k < \text{rank}(\hat{\boldsymbol{\Sigma}}_{\text{SCM}})$ then (1.17) admits the solution of Proposition 5. Thus, n can be arbitrary small and $\hat{\boldsymbol{\Sigma}}$ from Proposition 5 still belongs to $\mathcal{S}_p^{++}$. Thus, Equation (1.17) admits a minimizer if and only if at least one centered data $\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{\text{SM}}$ is not zero, *i.e.* $\text{rank}([\boldsymbol{x}_1 - \hat{\boldsymbol{\mu}}_{\text{SM}}, \cdots, \boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_{\text{SM}})] \geq 1$, as for the regularized estimation from Equation (1.15). A second remark is that the conditioning of $\hat{\boldsymbol{\Sigma}}$ is greater than the one of the SCM since $\hat{\sigma}^2 \geq \lambda_{\text{min}}$ where λ_{min} is the lowest eigenvalue of the SCM. Finally, it should be noted that there exists methods to choose automatically the rank k; see *e.g.* [92].

1.4 . *K-means++* and *Nearest centroïd classifier*

Once the statistical features are estimated, it remains to cluster/classify them. It is the third step of the pipeline from Figure 1.7. To do so, we leverage two simple machine learning algorithms: *K-means++* [7] and *Nearest centroid classifier*. The first one is a clustering algorithm, *i.e.* it does not make use of labels. The second one uses labels and thus is a classification algorithm. In the following, a feature is denoted θ and belongs to the set $\mathcal{M}$. For example, a feature can be the SCM which is a symmetric positive definite matrix, *i.e.* $\theta = \hat{\boldsymbol{\Sigma}}_{\text{SCM}} \in \mathcal{M} = \mathcal{S}_p^{++}$. We insist on the fact that the described algorithms are general. Indeed, they also apply to couples of parameters, *e.g.* the MLE of the Gaussian distribution $\theta = (\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}_{\text{SCM}})$, to parameters with constraints, *e.g.* belonging to a sphere, and etc ...

1.4.1 . Divergence, distance, and center of mass

Before going further, we give several definitions that are central for these algorithms. The first is one is that of divergence. This divergence measures the proximity between pairs of features θ_i and is leveraged in the definition of the *center of mass*.

Definition 9. *Given a set $\mathcal{M}$, the function $\delta : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ is a divergence if it satisfies the following conditions for all $\theta_1, \theta_2 \in \mathcal{M}$*

1. $\delta(\theta_1, \theta_2) \geq 0$ *(positivity),*
2. $\delta(\theta_1, \theta_2) = 0$ *if and only if* $\theta_1 = \theta_2$ *(separability).*

Given a subset of indices $\mathcal{I} \subset \mathbb{N}^*$, a definition of the center of mass c, associated with δ, of a set of features $\{\theta_i\}_{i \in \mathcal{I}}$ is a minimizer of the variance V [75],

$$c = \underset{\theta \in \mathcal{M}}{\arg\min} \left\{ V(\theta) = \frac{1}{\text{Card}\,(\mathcal{I})} \sum_{i \in \mathcal{I}} \delta(\theta, \theta_i) \right\} \tag{1.18}$$

where Card is the operator that returns the cardinality of a given set. We add two remarks to this definition. First, the minimum (1.18) is not necessarily unique and thus the center of mass of $\{\theta_i\}_{i \in \mathcal{I}}$ is also not necessarily unique. Second, in practice δ is differentiable with regards to its two arguments and we will simply look for a stationary point, *i.e.* $\text{grad}\, V(\theta) = 0$ with $\text{grad}\, V$ is a gradient of V. A subset of the divergences which is to be distinguished is that of the distances, *i.e.* symmetrical divergences that respect the triangle inequality.

Definition 10. *Given a set $\mathcal{M}$, the function $d : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ is a distance if it is a divergence (Definition 9) and if it satisfies the following conditions for all $\theta_1, \theta_2, \theta_3 \in \mathcal{M}$*

1. $d(\theta_1, \theta_2) = d(\theta_2, \theta_1)$ *(symmetry),*

2. $d(\theta_1, \theta_2) \leq d(\theta_1, \theta_3) + d(\theta_3, \theta_2)$ *(triangle inequality).*

Often divergences (that are not distances) are homogeneous to squared distances. Thus, (1.18) becomes

$$c = \underset{\theta \in \mathcal{M}}{\arg\min} \left\{ V(\theta) = \frac{1}{\mathrm{Card}\,(\mathcal{I})} \sum_{i \in \mathcal{I}} d^2(\theta, \theta_i) \right\}. \tag{1.19}$$

Then, we give two examples. The first one presents the Gaussian *Kullback Leibler* (KL) divergence on $\mathcal{S}_p^{++}$ and its associated center of mass which is the harmonic mean. The second one presents the Euclidean distance between matrices and its associated center of mass which is the arithmetic mean. Both examples give practical divergences between covariance matrices $\boldsymbol{\Sigma}_i \in \mathcal{S}_p^{++}$ as well as practical centers of mass.

Example 1. *Let $\{\boldsymbol{\Sigma}_i\}_{i \in \mathcal{I}} \subset \mathcal{S}_p^{++}$ with $\mathcal{I} \subset \mathbb{N}^*$, the Gaussian KL divergence on $\mathcal{S}_p^{++}$ is*

$$\delta_{KL}(\boldsymbol{\Sigma}_i, \boldsymbol{\Sigma}_j) = \frac{1}{2} \left(\mathrm{Tr}\left(\boldsymbol{\Sigma}_j^{-1} \boldsymbol{\Sigma}_i \right) + \log \left| \boldsymbol{\Sigma}_j \boldsymbol{\Sigma}_i^{-1} \right| - p \right). \tag{1.20}$$

The center of mass $\boldsymbol{C} \in \mathbb{R}^{p \times n}$ is defined as

$$\boldsymbol{C} = \underset{\boldsymbol{\Sigma} \in \mathcal{S}_p^{++}}{\arg\min} \left\{ V(\boldsymbol{\Sigma}) = \frac{1}{\mathrm{Card}\,(\mathcal{I})} \sum_{i \in \mathcal{I}} \delta_{KL}(\boldsymbol{\Sigma}, \boldsymbol{\Sigma}_i) \right\}. \tag{1.21}$$

By cancelling the gradient of V, we get that the center of mass is the harmonic mean

$$\boldsymbol{C} = \left(\frac{1}{\mathrm{Card}\,(\mathcal{I})} \sum_{i \in \mathcal{I}} \boldsymbol{\Sigma}_i^{-1} \right)^{-1}. \tag{1.22}$$

Example 2. *Let $\{\boldsymbol{A}_i\}_{i \in \mathcal{I}} \subset \mathbb{R}^{p \times n}$ with $\mathcal{I} \subset \mathbb{N}^*$, a distance on $\mathbb{R}^{p \times n}$ is*

$$d_{\mathbb{R}^{p \times n}}(\boldsymbol{A}_i, \boldsymbol{A}_j) = \|\boldsymbol{A}_i - \boldsymbol{A}_j\|_2 = \sqrt{\mathrm{Tr}\left((\boldsymbol{A}_i - \boldsymbol{A}_j)^T (\boldsymbol{A}_i - \boldsymbol{A}_j) \right)}. \tag{1.23}$$

The center of mass $\boldsymbol{Y} \in \mathbb{R}^{p \times n}$ is defined as

$$\boldsymbol{C} = \underset{\boldsymbol{Y} \in \mathbb{R}^{p \times n}}{\arg\min} \left\{ V(\boldsymbol{Y}) = \frac{1}{\mathrm{Card}\,(\mathcal{I})} \sum_{i \in \mathcal{I}} d^2_{\mathbb{R}^{p \times n}}(\boldsymbol{Y}, \boldsymbol{A}_i) \right\}. \tag{1.24}$$

By cancelling the gradient of V, we get that the center of mass is the classical elementwise arithmetic mean

$$\boldsymbol{C} = \frac{1}{\mathrm{Card}\,(\mathcal{I})} \sum_{i \in \mathcal{I}} \boldsymbol{A}_i. \tag{1.25}$$

Algorithm 1: *K-means++* on $\mathcal{M}$ with the divergence δ

Input : A set $\{\theta_i\}_{i=1}^M \subset \mathcal{M}$ to partition, a number of clusters K and a number of initializations n_{init}.
Output: Best partition $S^\star$.

$\phi^\star \leftarrow +\infty$
for 1 *to* n_{init} **do**
  # Initialization
  Take one center c_1, drawn uniformly from $\{\theta_i\}_{i=1}^M$.
  while $\mathrm{Card}(\{c_i\}) < K$ **do**
    Draw a new center $c_j \in \{\theta_i\}_{i=1}^M$ with probability $P(c_j = \theta_i) = \frac{D^2(\theta_i)^2}{\sum_{m=1}^M D^2(\theta_m)^2}$
  end
  # K-means
  while *no convergence* **do**
    Assignment step: $\forall i \in [\![1, M]\!]$ assign θ_i to the cluster S_j with the nearest c_j, $j \in [\![1, K]\!]$, using the divergence δ.
    Update step: Compute new centers c_j of clusters S_j, $\forall j \in [\![1, K]\!]$, using (1.18).
  end
  Compute $\phi(S)$ with (1.26).
  if $\phi(S) < \phi^\star$ **then**
    $S^\star \leftarrow S$
    $\phi^\star \leftarrow \phi(S)$
  end
end

1.4.2 . *K-means++*

We now have defined the necessary tools to implement *K-means++* and *Nearest centroïd classifier* associated to any divergence δ. We begin by describing *K-means++*. In the following, we assume having a set of features $\{\theta_i\}_{i=1}^M$ to partition into K subsets. In the following, the partition is denoted $S = \{S_1, \cdots, S_K\}$. We recall that such a partition is a set of K non-empty subsets such that every element of $\{\theta_i\}_{i=1}^M$ belongs to exactly one of these subsets. Since every θ_i is associated with one data point $\boldsymbol{X}_i$, a partition of $\{\theta_i\}_{i=1}^M$ gives a partition of the original data $\{\boldsymbol{X}_i\}_{i=1}^M$. First of all, *K-means++* initializes cluster centers $\{c_j\}_{j=1}^K$ by recursively choosing points θ_i with probability $\frac{D(\theta_i)}{\sum_{m=1}^M D(\theta_m)}$ [7]. Here, $D(\theta_i)$ denotes the divergence δ from θ_i to the closest center among those already chosen. Intuitively, this initialization is performed such that cluster centers are far away from each other at the initialization. We will see later that this initialization

gives a theoretical guarantee to *K-means++*. Once these cluster centers are initialized, *K-means++* iteratively applies two steps [7]:

1. **Assignment step**: each θ_i is assigned to the cluster S_j whose center c_j is the closest using the divergence δ,

2. **Update step**: each new cluster center c_j is computed as (1.18).

Once terminated, *K-means++* outputs the partition S. Intuitively, *K-means++* finds clusters S_j whose points $\theta_i \in S_j$ are close to each other using the divergence δ.

To analyze the performance of *K-means++* algorithm, we begin by defining the within-cluster sum of squares (WCSS),

$$\phi(S) = \sum_{j=1}^{K} \sum_{\theta_i \in S_j} \delta(c_j, \theta_i). \tag{1.26}$$

Unfortunately, finding the optimal partition that reaches ϕ_{OPT}, the minimum value of (1.26), is a NP-hard problem [81]. However, we can prove that *K-means++* algorithm decreases (1.26) and converges. Indeed, both steps "**Assignment step**" and "**Update step**" decrease (1.26) and, since δ is a divergence, $\phi(S) \geq 0$ for all the partitions S. Remarkably, [7] goes much further by proving that if δ is a squared distance, *i.e.* $\delta \equiv d^2$, then in expectation the WCSS of a partition produced by *K-means++* algorithm is upper bounded with respect to ϕ_{OPT}

$$\mathbb{E}[\phi] \leq 8(\ln K + 2)\phi_{\mathsf{OPT}} \tag{1.27}$$

where the expectation is taken with respect to the seeding procedure of the initialization. [10] This property is central to *K-means++* algorithm since it is proven that a plain *K-means* [80] cannot admit such a bound. Moreover, this bound is true from the initialization of *K-means++* algorithm. However, the clustering returned by *K-means++* is still not necessarily a global minimum of (1.26). Hence, a standard practice is to run the algorithm several times with different initializations and then to keep the clustering with the lowest WCSS (1.26). *K-means++* on $\mathcal{M}$ associated with the divergence δ and with the strategy of several initializations is presented in Algorithm 1.

1.4.3 . *Nearest centroïd classifier*

Let a K-class classification problem on a set $\mathcal{M}$ endowed with a divergence δ and a center of mass computation (1.18). Thus, a training set $\mathcal{T}_{\mathsf{train}} = \{(\theta_i, y_i)\}_{i=1}^{M_{\mathsf{train}}} \subset \mathcal{M} \times [\![1, K]\!]$ as well as a test set $\mathcal{T}_{\mathsf{test}} = \{\upsilon_i\}_{i=1}^{M_{\mathsf{test}}} \subset \mathcal{M}$ are

[10] The proof in [7] relies on the Euclidean distance between vectors however it can be easily extended to any distance as stated in [98].

Algorithm 2: *Nearest centroïd classifier* on $\mathcal{M}$ with the divergence δ

Input : A training set $\mathcal{T}_{\text{train}} = \{(\theta_i, y_i)\}_{i=1}^{M_{\text{train}}} \subset \mathcal{M} \times [\![1, K]\!]$ and a test set $\mathcal{T}_{\text{test}} = \{\upsilon_i\}_{i=1}^{M_{\text{test}}} \subset \mathcal{M}$.
Output: Predictions of the test set $\{y_i\}_{i=1}^{M_{\text{test}}} \subset [\![1, K]\!]$.
# Training
for $j = 1$ *to* K **do**
  Compute the center of mass c_j of $\{\theta_i \in \mathcal{T}_{\text{train}} | y_i = j\}$ using (1.18).
end
# Testing
for $i = 1$ *to* M_{test} **do**
  Assign υ_i to the class with the nearest center of mass c_j using the divergence δ.
end

available. The objective is to present *Nearest centroïd classifier* to predict the labels of the test set. This algorithm is simple and consists of two steps. First, it computes the center of mass of each class, also called *class center*, *i.e.* it computes the center of mass of $\{\theta_i \in \mathcal{T}_{\text{train}} | y_i = j\}$ for all $j \in [\![1, K]\!]$. Then, it assigns to each $\upsilon_i \in \mathcal{T}_{\text{test}}$ the label of the closest class center using the divergence δ. *Nearest centroïd classifier* is detailed in Algorithm 2.

1.5 . Riemannian perspectives of the clustering-classification pipeline

So far, we defined $\mathcal{M}$ as being a set containing the estimates of a given statistical estimator. In Section 1.3, we gave some examples of these estimators and we mentioned that they belong to many different sets $\mathcal{M}$ such as $\mathcal{S}_p^{++}$, $\mathbb{R}^p \times \mathcal{S}_p^{++}$, $\mathbb{R}^p \times \mathcal{S}_p^{+}$, $\mathcal{S}_p^{+}$ with $\operatorname{rank} = k$, ... All these sets can be formalized as Riemannian manifolds. The interests of this formalization are numerous such as transforming non-convex estimation problems to geodesically convex ones, handling constraints of the parameter space, developing fast estimators, computing Fisher-Rao distances (for machine learning applications), deriving Intrinsic Cramér-Rao bounds (ICRBs), ... Therefore, in the following, $\mathcal{M}$ is a Riemannian manifold. The latter generalizes the classical Euclidean sets. Its formalism as well as the motivation of its usage are detailed latter. We begin this section with an implementation of the clustering-classification pipeline from Section 1.2 on the Riemannian manifold $\mathcal{S}_p^{++}$. Then, we highlight the different contributions of this manuscript on this pipeline.

1.5.1 . Riemannian geometry in the clustering-classification pipeline

Shortly, a Riemannian manifold is a set that can be curved but is locally Euclidean. An example is the $p-1$-dimensional sphere S^{p-1} in $\mathbb{R}^p$ with the Euclidean inner product. Other examples are the different parameter spaces of the cost functions from Section 1.3. The theory of the *Riemannian geometry* [1, 19] is introduced in Chapter 2 and no prior knowledge in this field of the mathematics is required to read this manuscript.

To motivate the use of the Riemannian geometry, we detail a basic implementation of the clustering-classification pipeline, presented in Section 1.2, on the Riemannian manifold of the symmetric positive definite matrices $\mathcal{S}_p^{++}$. This Riemannian manifold is introduced in Chapter 2. Here, we only use its Riemannian distance. The pipeline we present in this section has been applied with great successes in the last decade in the EEG/MEG (Electroencephalography/Magnetoencephalography) community [8, 45] as well as in the SAR community [54]. First of all, [8] assumes that data $\{\boldsymbol{x}_i\}_{i=1}^n$ are independent realizations of a random variable $\boldsymbol{x}$ following a centered Gaussian distribution, *i.e.* $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$. Thus, in the feature estimation step (second step), the corresponding NLL is minimized by the SCM

$$\hat{\boldsymbol{\Sigma}}_{\mathrm{SCM}} = \underset{\boldsymbol{\Sigma} \in \mathcal{S}_p^{++}}{\arg\min} \ \mathcal{L}_{\mathrm{G}}(\boldsymbol{\Sigma} | \{\boldsymbol{x}_i\}_{i=1}^n). \tag{1.28}$$

Hence, each batch of data $\{\boldsymbol{x}_i\}_{i=1}^n$ is transformed into a covariance matrix that belongs to $\mathcal{S}_p^{++}$. It implies that the classification (third step) must be performed on $\mathcal{S}_p^{++}$. To do so, [8] uses the Riemannian distance on $\mathcal{S}_p^{++}$. Given $\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2 \in \mathcal{S}_p^{++}$, it writes

$$d_{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2) = \left\| \log\left(\boldsymbol{\Sigma}_1^{-\frac{1}{2}} \boldsymbol{\Sigma}_2 \boldsymbol{\Sigma}_1^{-\frac{1}{2}} \right) \right\|_2 \tag{1.29}$$

where $\log : \mathcal{S}_p^{++} \to \mathcal{S}_p$ is the matrix logarithm. One of the key properties of $d_{\mathcal{S}_p^{++}}$ is its affine invariance. Indeed, we have that, for all $\boldsymbol{A} \in \mathrm{GL}_p$, the set of $p \times p$ invertible matrices,

$$d_{\mathcal{S}_p^{++}}\left(\boldsymbol{A}\boldsymbol{\Sigma}_1\boldsymbol{A}^T, \boldsymbol{A}\boldsymbol{\Sigma}_2\boldsymbol{A}^T\right) = d_{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2). \tag{1.30}$$

This means that for data with a linear mixing model (1.1) $\boldsymbol{x}_i = \boldsymbol{A}\boldsymbol{w}_i$ (neglecting the noise) with $\boldsymbol{A} \in \mathrm{GL}_p$, we have[11]

$$d_{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2) = d_{\mathcal{S}_p^{++}}(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2) \tag{1.31}$$

where $\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2$ are SCMs computed on raw data $\boldsymbol{x}_i$ and $\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2$ are the corresponding SCMs computed on unmixed signals $\boldsymbol{w}_i$, *i.e.* $\boldsymbol{\Theta}_1 = \boldsymbol{A}^{-1}\boldsymbol{\Sigma}_1\boldsymbol{A}^{-T}$

[11] In the case where $\boldsymbol{A}$ has more rows than columns, then the data $\boldsymbol{x}_i$ must first be expressed with respect to a basis of $\mathrm{span}(\boldsymbol{A})$; *e.g.* doing a principal component analysis.

and $\boldsymbol{\Theta}_2 = \boldsymbol{A}^{-1}\boldsymbol{\Sigma}_2\boldsymbol{A}^{-T}$. Intuitively, this means that we can measure a distance between covariances of unmixed signals $\boldsymbol{w}_i$ by measuring a distance between covariances of raw data $\boldsymbol{x}_i$. This way, we remove the need of a preprocessing step to unmix the signal. Also, Equation (1.30) is true for all $\boldsymbol{A} \in \mathrm{GL}_p$, hence it is robust to $\boldsymbol{A}$ contrary to a preprocessing step that would unmix the signal with an estimated $\widehat{\boldsymbol{A}}$. Then, from Equation (1.19), the center of mass $\boldsymbol{\Sigma}$ of a set of covariance matrices $\{\boldsymbol{\Sigma}_i\}_{i\in\mathcal{I}}$ is

$$\boldsymbol{\Sigma} = \underset{\boldsymbol{\Sigma}\in\mathcal{S}_p^{++}}{\arg\min} \frac{1}{\mathrm{Card}(\mathcal{I})} \sum_{i\in\mathcal{I}} d^2_{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma}, \boldsymbol{\Sigma}_i). \tag{1.32}$$

This minimization problem can be achieved with a Riemannian gradient descent on $\mathcal{S}_p^{++}$. The algorithm of the Riemannian gradient descent generalizes the classical gradient descent to Riemannian manifolds and is detailed in Chapter 2. It should be noted that, using the affine invariance (1.30), if $\boldsymbol{\Sigma}$ is the center of mass of $\{\boldsymbol{\Sigma}_i\}_{i\in\mathcal{I}}$, then $\boldsymbol{\Theta} = \boldsymbol{A}^{-1}\boldsymbol{\Sigma}\boldsymbol{A}^{-T}$ is the center of mass of $\{\boldsymbol{\Theta}_i = \boldsymbol{A}^{-1}\boldsymbol{\Sigma}_i\boldsymbol{A}^{-T}\}_{i\in\mathcal{I}}$. Finally, [8] uses the distance (1.29) and the center of mass (1.32) in the *Nearest centroïd classifier* described in Algorithm 2 to classify the SCMs. In a clustering problem, the *Nearest centroïd classifier* can be replaced by *K-means++* described in Algorithm 1. This ends a first implementation of the clustering-classification pipeline described in Section 1.2. An important remark is that, thanks to the affine invariance (1.30), classifying (with a *Nearest centroïd classifier* or a *K-means++*) the raw data $\boldsymbol{x}_i$ or the unmixed signal $\boldsymbol{w}_i$ gives exactly the same labels prediction with this pipeline. From a practical point of view, we report that using this pipeline, on the *Breizhcrops* dataset, with the Euclidean distance between SCM (1.23) gives 23% of OA versus 56% with the affine invariant distance (1.30).

1.5.2 . Contributions

Many contributions can be done on the presented pipeline. A first axis of contributions concerns the feature estimation step. Indeed, other estimators of the covariance matrix than the SCM can be used. For example, [54] uses the Tyler's M-estimator (1.12) instead of the SCM for SAR image classification. Many other possibilities exist: robust estimators, joint estimators of the location and the covariance matrix, subspaces estimators, and etc ... All these estimators are solutions of a cost function (1.2) over a Riemannian manifold $\mathcal{M}$. In the case where a closed form formula of the solution is not known, Riemannian optimization can be employed on $\mathcal{M}$. This is the first axis of contributions: the development of estimators that rely on Riemannian optimization. The advantages of the Riemannian optimization compared to other more classical methods such as fixed points estimators are numerous. We mention some of them:

- constrained estimators: easily handles constraints on the parameters,

e.g. estimation of an orthogonal basis of a subspace,

- diversity of optimizers: many different Riemannian optimization algorithms can be employed depending on the problem,

- guarantee of convergence: under reasonable conditions (which are counterparts of Euclidean gradient based optimization algorithms conditions) Riemannian gradient based optimization algorithms converge to a solution,

- large scale learning: fast estimators with the Riemannian stochastic gradient descent,

- geodesic convexity (convexity along geodesics, extension of straight lines to manifolds): changing geodesics can transform a non-convex problem to a convex one: uniqueness of the solution and fast optimization on strongly geodesically convex cost functions,

- fast estimators with statistical manifolds (Riemannian manifolds equipped with the Fisher information metric) for estimation problems, etc ...

Contributions on this axis are presented in Chapters 3 and 4.

A second axis of contributions concerns the second step with the computation of ICRB (CRBs on Riemannian manifolds). The latter illustrate the performance of a given estimator on a Riemannian manifold. They present several advantages compared to classical CRBs:

- constrained estimators: the Riemannian distance and the ICRB take into account the constraints of the estimation problem, *e.g.* constraints of orthogonality of a subspace basis, and thus are more interpretable and easier to derive than their Euclidean counterparts

- parameter-free bounds: when the distance associated with the Fisher information metric is known, the ICRB is the dimension of the parameter space and thus is parameter-free (this point of view is presented in Chapter 2, Section 2.5).

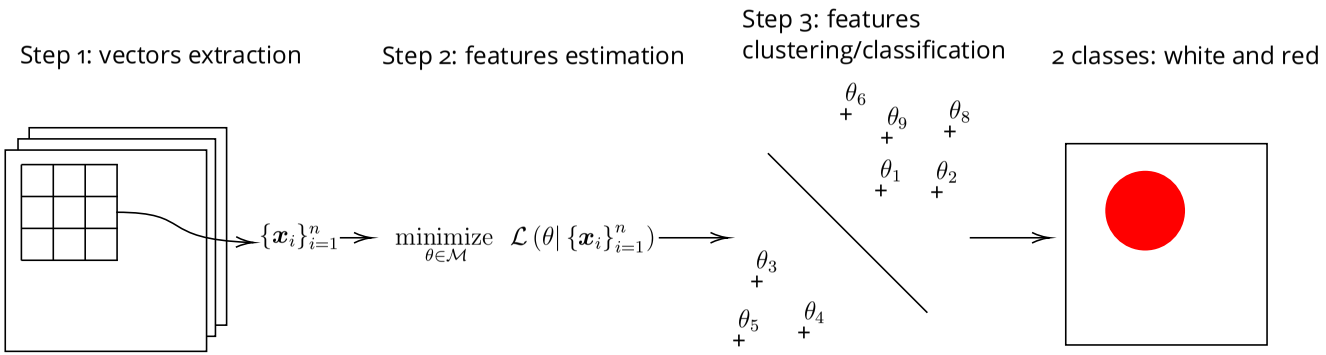
Contributions on this axis are presented in Chapter 4 with the ICRBs of a subspace estimation problem.

A third axis of contributions concerns the third step. Once features are estimated, a divergence and its corresponding center of mass must be defined. When a covariance matrix is estimated on $\mathcal{S}_p^{++}$ then the Riemannian distance (1.29) and its associated center of mass (1.32) can be used. Indeed, it is affine invariant (1.30) and gives very good performance in practice

compared to other more simple divergences such as the Euclidean distance. However, when $\mathcal{M} \neq \mathcal{S}_p^{++}$, *e.g.* when the location is added, $\mathcal{M} = \mathbb{R}^p \times \mathcal{S}_p^{++}$, then other divergences must be developed. Indeed, the Riemannian distance on a given given statistical manifold (different than $\mathcal{S}_p^{++}$) is often intractable. Chapter 3 presents contributions on the use of geodesic triangles on the statistical manifold of non-centered Gaussian distributions, *i.e.* $\mathcal{M} = \mathbb{R}^p \times \mathcal{S}_p^{++}$. Two affine invariant divergences are proposed and the associated centers of mass are estimated using Riemannian optimization. When geodesics on the statistical manifold are not known (which is often the case), other choices must be made. Chapter 3 describes contributions on the use of a KL divergence on the statistical manifold of NC-MSGs, *i.e.* $\mathcal{M} = \mathbb{R}^p \times \mathcal{S}_p^{++} \times (\mathbb{R}_*^+)^n$. The associated center of mass is estimated using Riemannian optimization. Finally, Chapter 4 presents a simplification of the Fisher information metric of a low-rank structured statistical model in order to get a closed form formula of the Riemannian distance. Then, the associated center of mass is derived using Riemannian optimization.

A fourth axis of contribution is the metric learning presented in Chapter 5. So far, we estimate parameters θ that belong Riemannian manifolds $\mathcal{M}$. These parameters are clustered-classified using divergences and centers of mass. The metric learning approach is different. Instead of using a predefined metric on the parameter space $\mathcal{M}$, we learn a metric directly on the sample space $\mathcal{X}$. Once this metric has been learned, data $\boldsymbol{x}_i$ are whitened by this metric and then classified directly on $\mathcal{X}$. Chapter 5 shows that this problem is closely related to covariance estimation problems. Two geodesically convex minimization problems are formulated and they are solved using fast Riemannian optimizers.

The different contributions on the pipeline as well as the statistical models are summarized in Figure 1.9.



Chapter 3:
$\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \tau_i\boldsymbol{\Sigma})$:
estimation using a product metric and the Fisher information metric.

Chapter 4:
$\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{0}, \tau_i\boldsymbol{U}\boldsymbol{U}^T + \boldsymbol{I}_p)$
(low-rank + identity model):
estimation using the Fisher information metric, ICRBs on the subspace $\mathrm{span}(\boldsymbol{U})$ and the textures τ_i.

Chapter 3:
$\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$:
Nearest centroïd classifier using geodesic triangles: divergences and centers of mass,
$\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \tau_i\boldsymbol{\Sigma})$:
Nearest centroïd classifier using a KL divergence: divergence and center of mass.

Chapter 4:
$\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{0}, \tau_i\boldsymbol{U}\boldsymbol{U}^T + \boldsymbol{I}_p)$
K-means++ using a product metric: distance and center of mass.

Chapter 5:
classification on raw data $\boldsymbol{x}_i$: learning metrics with geodesically convex estimation problems.

Figure 1.9: Summary of the contributions.

1.A . Appendix

1.A.1 . Classes of the *Indian Pines* dataset

#	Class	Number of samples
1	Alfalfa	46
2	Corn-notill	$1,428$
3	Corn-mintill	718
4	Corn	229
5	Grass-pasture	438
6	Grass-trees	730
7	Grass-pasture-mowed	28
8	Hay-windrowed	478
9	Oats	20
10	Soybean-notill	943
11	Soybean-mintill	$2,371$
12	Soybean-clean	577
13	Wheat	205
14	Woods	$1,265$
15	Buildings-Grass-Trees-Drives	290
16	Stone-Steel-Towers	93
	Total	$9,859$

Table 1.1: *Indian Pines* [9] classes.

1.A.2 . Classes of the *Breizhcrops* dataset

#	Class	Number of samples
1	Barley	$36,905$
2	Wheat	$89,555$
3	Rapeseed	$14,732$
4	Corn	$153,908$
5	Sunflower	19
6	Orchards	$3,070$
7	Nuts	49
8	Permanent meadows	$127,813$
9	Temporary meadows	$182,212$
	Total	$608,263$

Table 1.2: *Breizhcrops* [118] classes.

2 - Riemannian geometry, optimization and intrinsic Cramér-Rao bounds

Riemannian geometry [1, 19] has received an increasing interest over the years both for being theoretically appealing and for its multiple applications in signal processing and machine learning. This chapter is devoted to the introduction of the theory of Riemannian geometry as well as giving some Examples of *Riemannian manifolds*. This presentation is meant to be self-contained and requires only basic knowledge of linear algebra and calculus. Riemannian geometry being a very rich theory, we concentrate only on the exposition of the necessary tools for the following chapters. Several Riemannian manifolds needed for these chapters are presented: the manifold of $p \times p$ symmetric positive definite matrices with the affine invariance Riemannian metric denoted $\mathcal{S}_p^{++}$, the manifold of $p \times p$ symmetric positive definite matrices with unit determinant denoted $\mathcal{SS}_p^{++}$, the manifold of n-dimensional strictly positive vectors denoted $(\mathbb{R}_*^+)^n$, the compact Stiefel manifold denoted $\mathrm{St}_{p,k}$, and the Grassmann quotient manifold of k-dimensional subspaces in $\mathbb{R}^p$ denoted $\mathrm{Gr}_{p,k}$.

Two uses of Riemannian geometry will follow us throughout this manuscript: statistical estimation and classification on manifold. These applications typically make use of *smooth embedded submanifolds of linear spaces*. Examples are the sphere or the set of symmetric positive definite matrices (and its submanifolds). Thus, Section 2.1 begins by introducing what is a Riemannian manifold with the smooth embedded submanifolds of linear spaces. This introduction heavily relies on the excellent books [1, 19] and does not make use of more advanced tools such as *charts* or *atlases*. Furthermore, the presented concepts will be illustrated on the sphere in order to build an intuition. Then, in Section 2.3, concepts from Section 2.1 are extended to *Riemannian quotient manifolds*. These manifolds are of particular interest when dealing with functions with invariances such as functions of *linear subspaces*. Examples of important manifolds for the next chapters are presented in Sections 2.4. Finally, *Intrinsic Cramér-Rao bounds* are covered in Section 2.5.

2.1 . Elements of Riemannian geometry

As explained in the introduction of this section, most *smooth manifold* used in signal processing or machine learning are smooth embedded submanifold of linear spaces. A linear space (or *vector space*) over the reals, denoted $\mathcal{E}$ in the following, is a set whose elements may be added together

and multiplied by real numbers. Classical examples of linear spaces are $\mathbb{R}^d$, $\mathbb{R}^{n\times p}$, $\mathcal{S}_p$ (set of $p \times p$ real symmetric matrices). This definition can be extended to any *field* such as the *complex numbers*. In order to define smooth embedded submanifolds of linear spaces, *linear maps* and *differentials* are introduced. Given $\mathcal{E}, \mathcal{F}$ two linear spaces, $f : \mathcal{E} \to \mathcal{F}$ is a linear map if $f(ax + by) = af(x) + bf(y)$ for all $x, y \in \mathcal{E}$, $a, b \in \mathbb{R}$. Let U, V be open sets in two linear spaces $\mathcal{E}, \mathcal{F}$. A map $f : U \to V$ is *smooth* if it is infinitely differentiable on its domain. The differential of f at x is the linear map $\mathrm{D}\, f(x) : \mathcal{E} \to \mathcal{F}$ defined by

$$\mathrm{D}\, f(x)[\xi] = \lim_{t\to 0} \frac{f(x + t\xi) - f(x)}{t} \tag{2.1}$$

where $\xi \in \mathcal{E}$. $\mathrm{D}\, f(x)[\xi]$ is called the *directional derivative* of f at x in the direction ξ. Some classical rules of differentiation have their extensions for the directional derivatives. Given maps $f, g : U \to V$ the *sum rule* writes

$$\mathrm{D}(f + g)(x)[\xi] = \mathrm{D}\, f(x)[\xi] + \mathrm{D}\, g(x)[\xi]. \tag{2.2}$$

where $(f + g)(x) = f(x) + g(x)$. Then, the *product rule* writes

$$\mathrm{D}(f \times g)(x)[\xi] = \mathrm{D}\, f(x)[\xi]g(x) + f(x)\, \mathrm{D}\, g(x)[\xi] \tag{2.3}$$

where $(f \times g)(x) = f(x)g(x)$. Finally, the *chain rule* writes

$$\mathrm{D}(f \circ g)(x)[\xi] = \mathrm{D}\, f(g(x))[\mathrm{D}\, g(x)[\xi]] \tag{2.4}$$

where $(f \circ g)(x) = f(g(x))$. We give the directional derivates of some classical maps on matrices in the following example.

Example 3. *In this example, some directional derivatives of classical maps are computed.*

- *Constant function:* $f(\boldsymbol{X}) = \boldsymbol{C}$*, for* $\boldsymbol{C} \in \mathbb{R}^{p\times n}$*.*

$$\mathrm{D}\, f(\boldsymbol{X})[\boldsymbol{\xi}] = \lim_{t\to 0} \frac{\boldsymbol{C} - \boldsymbol{C}}{t} = \boldsymbol{0}.$$

- *Identity function:* $f(\boldsymbol{X}) = \boldsymbol{X}$*, for* $\boldsymbol{X} \in \mathbb{R}^{p\times n}$*.*

$$\mathrm{D}\, f(\boldsymbol{X})[\boldsymbol{\xi}] = \lim_{t\to 0} \frac{\boldsymbol{X} + t\boldsymbol{\xi} - \boldsymbol{X}}{t} = \boldsymbol{\xi}.$$

- *Trace function:* $f(\boldsymbol{X}) = \mathrm{Tr}(\boldsymbol{X})$*, for* $\boldsymbol{X} \in \mathbb{R}^{p\times p}$*.*

$$\mathrm{D}\, f(\boldsymbol{X})[\boldsymbol{\xi}] = \lim_{t\to 0} \frac{\mathrm{Tr}(\boldsymbol{X} + t\boldsymbol{\xi}) - \mathrm{Tr}(\boldsymbol{X})}{t} = \mathrm{Tr}(\boldsymbol{\xi}).$$

- *Inverse function:* $f(\boldsymbol{X}) = \boldsymbol{X}^{-1}$*, for* $\boldsymbol{X} \in \mathsf{GL}_p$.
  To compute the directional derivative of f*, we use the directive derivative of the constant function* $f(\boldsymbol{X})\boldsymbol{X} = \boldsymbol{I}_p$ *and the product rule,*

$$\mathrm{D}(\boldsymbol{X} \mapsto f(\boldsymbol{X})\boldsymbol{X})[\boldsymbol{\xi}] = \mathrm{D}\, f(\boldsymbol{X})[\boldsymbol{\xi}]\boldsymbol{X} + f(\boldsymbol{X})\boldsymbol{\xi} = \boldsymbol{0}.$$

  Thus, we get the directional derivative of f

$$\mathrm{D}\, f(\boldsymbol{X})[\boldsymbol{\xi}] = -\boldsymbol{X}^{-1}\boldsymbol{\xi}\boldsymbol{X}^{-1}.$$

- *Quadratic function:* $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$*, for* $\boldsymbol{x} \in \mathbb{R}^p$ *and* $\boldsymbol{A} \in \mathcal{S}_p$ *the set of* $p \times p$ *symmetric matrices.*
  First, we remark that

$$\begin{aligned} f(\boldsymbol{x} + t\boldsymbol{\xi}) - f(\boldsymbol{x}) &= \frac{1}{2}(\boldsymbol{x} + t\boldsymbol{\xi})^T\boldsymbol{A}(\boldsymbol{x} + t\boldsymbol{\xi}) - \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \\ &= t\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{\xi} + \mathcal{O}(t^2). \end{aligned}$$

  Thus, we get the directional derivative of f

$$\mathrm{D}\, f(\boldsymbol{x})[\boldsymbol{\xi}] = \lim_{t\to 0} \frac{f(\boldsymbol{x} + t\boldsymbol{\xi}) - f(\boldsymbol{x})}{t} = \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{\xi}.$$

- *Log-det function:* $f(\boldsymbol{\Sigma}) = \log|\boldsymbol{\Sigma}|$*, for* $\boldsymbol{\Sigma} \in \mathcal{S}_p^{++}$ *the set of* $p \times p$ *symmetric positive definite matrices.*
  First, we notice that for $\boldsymbol{\Sigma} \in \mathcal{S}_p$

$$|\boldsymbol{\Sigma} + t\boldsymbol{\xi}| = |\boldsymbol{\Sigma}||\boldsymbol{I}_p + t\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\xi}\boldsymbol{\Sigma}^{-\frac{1}{2}}|.$$

  Thus, we get that for t *small enough*

$$\begin{aligned} \log|\boldsymbol{\Sigma} + t\boldsymbol{\xi}| - \log|\boldsymbol{\Sigma}| &= \log|\boldsymbol{\Sigma}| + \log|\boldsymbol{I}_p + t\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\xi}\boldsymbol{\Sigma}^{-\frac{1}{2}}| - \log|\boldsymbol{\Sigma}| \\ &= \log|\boldsymbol{I}_p + t\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\xi}\boldsymbol{\Sigma}^{-\frac{1}{2}}| \\ &= \sum_i \log(1 + t\lambda_i) \end{aligned}$$

  where the λ_i *are the (real) eigenvalues of* $\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\xi}\boldsymbol{\Sigma}^{-\frac{1}{2}}$*. It follows that*

$$\begin{aligned} \log|\boldsymbol{\Sigma} + t\boldsymbol{\xi}| - \log|\boldsymbol{\Sigma}| &= \sum_i \log(1 + t\lambda_i) \\ &= t\sum_i \lambda_i + \mathcal{O}(t^2) \\ &= t\,\mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}) + \mathcal{O}(t^2). \end{aligned}$$

  Hence, we get the directional derivative of the log-det function

$$\mathrm{D}\, f(\boldsymbol{\Sigma})[\boldsymbol{\xi}] = \lim_{t\to 0} \frac{\log|\boldsymbol{\Sigma} + t\boldsymbol{\xi}| - \log|\boldsymbol{\Sigma}|}{t} = \mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}).$$

- *Determinant function:* $f(\boldsymbol{\Sigma}) = |\boldsymbol{\Sigma}|$*, for* $\boldsymbol{\Sigma} \in \mathcal{S}_p^{++}$*.*
  To compute this directional derivative, we use the classical chain rule applied on the log-det function. For all $\boldsymbol{\xi} \in \mathcal{S}_p$*, we have*

$$\mathrm{D} \log |\boldsymbol{\Sigma}|[\boldsymbol{\xi}] = \frac{\mathrm{D}\, f(\boldsymbol{\Sigma})[\boldsymbol{\xi}]}{|\boldsymbol{\Sigma}|} = \mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}).$$

  Thus, we get the desired directional derivative

$$\mathrm{D}\, f(\boldsymbol{\Sigma})[\boldsymbol{\xi}] = |\boldsymbol{\Sigma}|\, \mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}).$$

2.1.1 . Smooth embedded submanifold of a linear space

We move on to the definition of smooth embedded submanifolds, denoted by $\mathcal{M}$, of linear spaces $\mathcal{E}$. Informally, these are subsets of $\mathcal{E}$ that are either opens or defined by constraints $h : \mathcal{E} \to \mathbb{R}^k$ with $k > 0$. In the latter case, a point $x \in \mathcal{E}$ belong to $\mathcal{M}$ if and only if $h(x) = 0$. Two important remarks are made about h. First, it should be a smooth function. Second, its *rank* should be constant and maximal, *i.e.* $\mathrm{span}(\mathrm{D}\, h(x)) = \mathbb{R}^k$. This last property is enforced so that $\ker(\mathrm{D}\, h(x))$ is a *linearization* (latter called *tangent space*) of $\mathcal{M}$ at x and thus $\mathcal{M}$ is locally *diffeomorphic* to $\mathbb{R}^k$.

Definition 11 (Definition 3.10 of [19])**.** *Let* $\mathcal{E}$ *be a linear space of dimension* d*. A nonempty subset* $\mathcal{M}$ *of* $\mathcal{E}$ *is a smooth embedded submanifold of* $\mathcal{E}$ *of dimension* q *if either*

1. $q = d$ *and* $\mathcal{M}$ *is open in* $\mathcal{E}$ *- we also call this an* open submanifold *or*
2. $q = d - k$ *for some* $k \geq 1$ *and, for each* $x \in \mathcal{M}$*, there exists a neighborhood* U *of* x *in* $\mathcal{E}$ *and a smooth function* $h : U \to \mathbb{R}^k$ *such that*
   (a) *If* y *is in* U*, then* $h(y) = 0$ *if only if* $y \in \mathcal{M}$*; and*
   (b) $\mathrm{rank}(\mathrm{D}\, h(x)) = k$*.*

   Such a function h *is called a local defining function for* $\mathcal{M}$ *at* x*.*

If $\mathcal{M}$ *is a linear subspace, we also call it a linear manifold.*

Then, *smooth curves* $c : I \to \mathcal{M}$, with I an open interval of $\mathbb{R}$, are defined on $\mathcal{M}$. Collecting velocities of the curves passing through $x \in \mathcal{M}$, we get the *tangent space* at x. Informally, it corresponds to a linearization of $\mathcal{M}$ at x. The tangent spaces are of utmost importance. Indeed, we will see later that they are linear spaces. Thus, classical operations such as addition or multiplication, and operations related to inner products are possible on the tangent space contrary to the manifold (which often is *not* a linear space !).

Definition 12 (Definition 3.14 of [19]). *Let $\mathcal{M}$ be a subset of $\mathcal{E}$. For all $x \in \mathcal{M}$, define*

$$T_x\mathcal{M} = \{\dot{c}(0) | c : I \to \mathcal{M} \text{ is smooth and } c(0) = x\} \tag{2.5}$$

where I is any open interval containing $t = 0$ and $\dot{c}(t) = \frac{d}{dt}c(t)$. That is, v is in $T_x\mathcal{M}$ if and only if there exists a smooth curve on $\mathcal{M}$ passing through x with velocity v.

Definition 12 is not of practical interest. Thus, another characterization of the tangent space is given in Theorem 3. Indeed, this theorem gives a way to compute the tangent space at x that is directly related to Definition 11 of embedded submanifolds of linear spaces.

Theorem 3 (Theorem 3.15 of [19]). *Let $\mathcal{M}$ be an embedded submanifold of $\mathcal{E}$. Consider $x \in \mathcal{M}$ and the set $T_x\mathcal{M}$ (2.5). If $\mathcal{M}$ is an open submanifold, then*

$$T_x\mathcal{M} = \mathcal{E}.$$

Otherwise,

$$T_x\mathcal{M} = \ker(\mathrm{D}\, h(x)) = \{\xi \in \mathcal{E} : \mathrm{D}\, h(x)[\xi] = 0\}$$

with h any local defining function at x.

Two examples are given. Example 4 states that $\mathbb{R}^d$ is a smooth manifold which is obvious using Definition 11. Its tangent spaces are also $\mathbb{R}^d$ which is also obvious using Theorem 3. Then, the sphere in $\mathbb{R}^d$, denoted by S^{d-1}, is presented as a smooth manifold in Example 5. A two dimensional illustration of this example is presented in Figure 2.1.

Example 4 (Example 3.17 from [19]). *The set $\mathbb{R}^d$ is a linear manifold of dimension d with tangent spaces $T_{\boldsymbol{x}}\mathcal{M} = \mathbb{R}^d$ for all $\boldsymbol{x} \in \mathbb{R}^d$.*

Example 5 (Example 3.18 from [19]). *The sphere $\mathsf{S}^{d-1} = \left\{\boldsymbol{x} \in \mathbb{R}^d : \boldsymbol{x}^T\boldsymbol{x} = 1\right\}$ is the zero level set of $h(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{x} - 1$, smooth from $\mathbb{R}^d$ to $\mathbb{R}$. Since $\mathrm{D}\, h(\boldsymbol{x})[\boldsymbol{\xi}] = 2\boldsymbol{x}^T\boldsymbol{\xi}$, it is clear that $\operatorname{rank}(\mathrm{D}\, h(\boldsymbol{x})) = 1$ for all $\boldsymbol{x} \in \mathsf{S}^{d-1}$. As a result, S^{d-1} is an embedded submanifold of $\mathbb{R}^d$ of dimension $d - 1$. Furthermore, its tangent spaces are given by $T_{\boldsymbol{x}}\mathsf{S}^{d-1} = \ker(\mathrm{D}\, h(\boldsymbol{x})) = \left\{\boldsymbol{\xi} \in \mathbb{R}^d : \boldsymbol{x}^T\boldsymbol{\xi} = 0\right\}$.*

2.1.2 . Riemannian structure

So far, we only have presented embedded submanifolds $\mathcal{M}$ of linear spaces $\mathcal{E}$. In order to be a Riemannian manifold, $\mathcal{M}$ must be endowed with a *Riemannian metric* on its tangent spaces. Before introducing Riemannian metrics, we first define the *tangent bundle* of $\mathcal{M}$ in Definition 13.

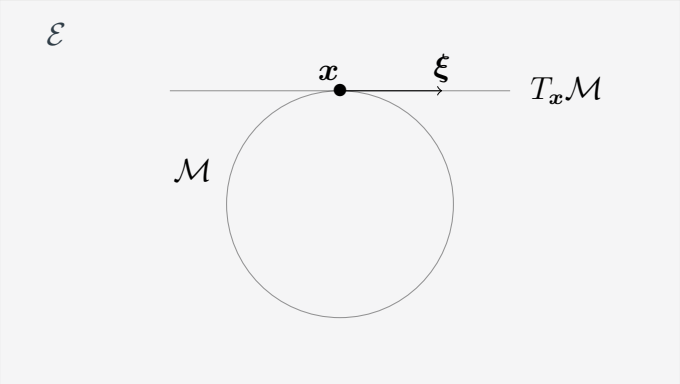


Figure 2.1: Illustration of a smooth embedded submanifold with the circle $\mathcal{M} = \mathsf{S}^1 \subset \mathcal{E} = \mathbb{R}^2$, its tangent space $T_{\boldsymbol{x}}\mathcal{M} = \left\{\boldsymbol{\xi} \in \mathbb{R}^2 : \boldsymbol{x}^T\boldsymbol{\xi} = 0\right\}$ at a given $\boldsymbol{x} \in \mathcal{M}$ and a tangent vector $\boldsymbol{\xi} \in T_{\boldsymbol{x}}\mathcal{M}$.

It is the *disjoint union* of the tangent spaces of $\mathcal{M}$ in the sense that every $\xi \in T_x\mathcal{M}$ is paired with x. The disjoint union is of first importance when ξ belongs to several tangent spaces such the tangent vectors of the manifold $\mathbb{R}^d$.

Definition 13 (Definition 3.42 from [19])**.** *The tangent bundle of a manifold $\mathcal{M}$ is the disjoint union of the tangent spaces of $\mathcal{M}$:*

$$T\mathcal{M} = \{(x, \xi) : x \in \mathcal{M} \text{ and } \xi \in T_x\mathcal{M}\}. \tag{2.6}$$

It remains to define *vector fields* and *inner products* before defining *Riemannian metrics*. Vector fields are introduced in Definition 14. These are map from $\mathcal{M}$ onto the tangent bundle $T\mathcal{M}$. An easy to visualize example is the wind map over the Earth (here assumed to be spherical): at each given point $x \in \mathcal{M}$, a vector $\xi \in T_x\mathcal{M}$ gives the orientation and the magnitude of the wind.

Definition 14 (Definition 3.44 from [19])**.** *A vector field on a manifold $\mathcal{M}$ is a map $\xi : \mathcal{M} \to T\mathcal{M}$ such that $\xi(x)$ is in $T_x\mathcal{M}$ for all $x \in \mathcal{M}$. If ξ is a smooth map, we say it is a smooth vector field. The set of smooth vector fields is denoted by $\mathfrak{X}(\mathcal{M})$.*

Then, inner products are defined on tangent spaces of the manifold. The choice on inner products on the different tangent spaces $T_x\mathcal{M}$ is called the *metric*.

Definition 15 (Definition 3.51 from [19])**.** *An inner product on $T_x\mathcal{M}$ is a bilinear, symmetric, positive definite function $\langle ., . \rangle_x : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$. It induces a norm for tangent vectors : $\|\xi\|_x = \sqrt{\langle \xi, \xi \rangle_x}$. A metric on $\mathcal{M}$ is a choice of inner product $\langle ., . \rangle_x$ for each $x \in \mathcal{M}$.*

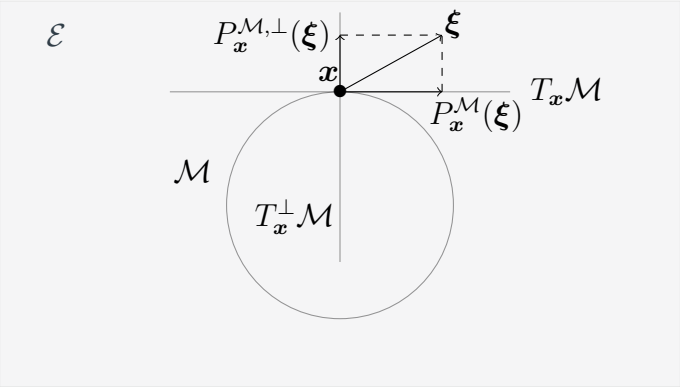


Figure 2.2: Illustration of the embedded submanifold $\mathcal{M} = \mathsf{S}^1$, its tangent space $T_{\boldsymbol{x}}\mathcal{M}$ at a given $\boldsymbol{x} \in \mathcal{M}$, a vector $\boldsymbol{\xi} \notin T_{\boldsymbol{x}}\mathcal{M}$ and its orthogonal projection $P_{\boldsymbol{x}}^{\mathcal{M}}(\boldsymbol{\xi})$ onto $T_{\boldsymbol{x}}\mathcal{M}$.

Thanks to the previous definitions, we can introduce the concept of Riemannian metric. This concept is very important since it is the basis of many other Riemannian objects. For example, geodesics, gradients, hessians and distances on manifolds are defined with respect to this metric. The Riemannian metric is defined in Definition 16 and is simply a metric that varies smoothly between tangent spaces. A manifold endowed with a Riemannian metric is a *Riemannian manifold*. The sphere S^{d-1} presented in Example 5 is turned into Riemannian manifold in Example 6.

Definition 16 (Definition 3.52 from [19])**.** *A metric $\langle ., . \rangle_x$ on $\mathcal{M}$ is a Riemannian metric if it varies smoothly with x, in the sense that for all smooth vector fields ξ, η on $\mathcal{M}$ the function $x \mapsto \langle \xi(x), \eta(x) \rangle_x$ is smooth from $\mathcal{M}$ to $\mathbb{R}$.*

Example 6 (Example 3.56 from [19])**.** *Endow $\mathbb{R}^d$ with the standard metric $\langle \boldsymbol{\xi}, \boldsymbol{\eta} \rangle = \boldsymbol{\xi}^T \boldsymbol{\eta}$ and consider the sphere S^{d-1} embedded in $\mathbb{R}^d$. With the inherited metric $\langle \boldsymbol{\xi}, \boldsymbol{\eta} \rangle_{\boldsymbol{x}} = \langle \boldsymbol{\xi}, \boldsymbol{\eta} \rangle = \boldsymbol{\xi}^T \boldsymbol{\eta}$ on each tangent space $T_{\boldsymbol{x}}\mathsf{S}^{d-1}$, the sphere S^{d-1} becomes a Riemannian manifold.*

2.1.3 . Orthogonal projection

So far, we defined inner products $\langle ., . \rangle_x$ on tangent spaces $T_x\mathcal{M}$ of $\mathcal{M}$. If this inner product also defines an inner product in the ambient space, *i.e.* if $(\xi, \eta) \in \mathcal{E} \times \mathcal{E} \mapsto \langle \xi, \eta \rangle_x$ is a bilinear, symmetric, positive definite function for all $x \in \mathcal{M}$, $\xi, \eta \in \mathcal{E}$, then an *orthogonal projection* $P_x^{\mathcal{M}} : \mathcal{E} \to T_x\mathcal{M}$ can be defined. Indeed, since an inner product is defined on all the ambient space $\mathcal{E}$, the subspace $T_x\mathcal{M} \subset \mathcal{E}$ admits an orthogonal complement which is the *normal space* and defined as

$$T_x^{\perp}\mathcal{M} = \{\xi \in \mathcal{E} : \langle \xi, \eta \rangle_x = 0 \quad \forall \eta \in T_x\mathcal{M}\}\,. \tag{2.7}$$

Thus, every element $\xi \in \mathcal{E}$ is uniquely decomposed as

$$\xi = P_x^{\mathcal{M}}(\xi) + P_x^{\mathcal{M},\perp}(\xi) \tag{2.8}$$

with the orthogonal projectors $P_x^{\mathcal{M}} : \mathcal{E} \to T_x\mathcal{M}$ and $P_x^{\mathcal{M},\perp} : \mathcal{E} \to T_x^{\perp}\mathcal{M}$.

Example 7. *Consider $\mathbb{R}^d$ with the standard metric $\langle \boldsymbol{\xi}, \boldsymbol{\eta} \rangle = \boldsymbol{\xi}^T \boldsymbol{\eta}$ and the sphere S^{d-1} embedded in $\mathbb{R}^d$. The ambient space $\mathbb{R}^d$ is the sum of two complementary and orthogonal spaces*

$$\mathbb{R}^d = T_{\boldsymbol{x}}\mathsf{S}^{d-1} + T_{\boldsymbol{x}}^{\perp}\mathsf{S}^{d-1}$$

with

$$T_{\boldsymbol{x}}^{\perp}\mathsf{S}^{d-1} = \{\alpha \boldsymbol{x} : \alpha \in \mathbb{R}\}.$$

To project $\boldsymbol{\xi} \in \mathcal{E}$ onto $T_{\boldsymbol{x}}\mathsf{S}^{d-1}$, it suffices to remove its component in $T_{\boldsymbol{x}}^{\perp}\mathsf{S}^{d-1}$, i.e.

$$P_{\boldsymbol{x}}^{\mathsf{S}^{d-1}}(\boldsymbol{\xi}) = \boldsymbol{\xi} - (\boldsymbol{x}^T\boldsymbol{\xi})\boldsymbol{x} = (\boldsymbol{I}_d - \boldsymbol{x}\boldsymbol{x}^T)\boldsymbol{\xi}.$$

2.1.4 . Levi-Civita connection

To define *affine connections*, we introduce *scalar fields*. A simple and illustrative example of the latter is the temperature on Earth (assumed to be spherical). Indeed, at each point of the earth corresponds a temperature, and thus defines a scalar fields. The definition of scalar fields is important since in the next chapters, functions on manifolds are minimized.

Definition 17 (Definition 3.32 from [19])**.** *A scalar field on a manifold $\mathcal{M}$ is a function $f : \mathcal{M} \to \mathbb{R}$. If f is a smooth function, we say it is a smooth scalar field. The set of smooth scalar fields on $\mathcal{M}$ is denoted by $\mathfrak{F}(\mathcal{M})$.*

Since vector and scalar fields are now defined, we can move on to *affine connections*. Affine connections are central in Riemannian geometry since they define the acceleration along a curve on a manifold and this acceleration defines geodesics. Definition 18 of affine connections is axiomatic: desired properties are specified and then the object, if it exists, is studied.

Definition 18 (Definition from [1])**.** *Let $\mathfrak{X}(\mathcal{M})$ denote the set of smooth vector fields on $\mathcal{M}$. An affine connection ∇ on a manifold $\mathcal{M}$ is a mapping*

$$\nabla : \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) \to \mathfrak{X}(\mathcal{M}),$$

which is denoted by $(\xi, \eta) \overset{\nabla}{\to} \nabla_\xi \eta$ and satisfies the following properties:

1. *$\mathfrak{F}(\mathcal{M})$-linearity in ξ: $\nabla_{f\xi+g\chi}\,\eta = f\,\nabla_\xi\,\eta + g\,\nabla_\chi\,\eta$,*
2. *$\mathbb{R}$-linearity in η: $\nabla_\xi(a\eta + b\zeta) = a\,\nabla_\xi\,\eta + b\,\nabla_\xi\,\zeta$,*

3. *Product rule (Leibniz' law):* $\nabla_\xi(f\eta) = (\xi f)\eta + f\,\nabla_\xi\,\eta$,

in which $\eta, \chi, \xi, \zeta \in \mathfrak{X}(\mathcal{M})$, $f, g \in \mathfrak{F}(\mathcal{M})$, *and* $a, b \in \mathbb{R}$. ξf *is the vector field such that* $(\xi f)(x) = \mathrm{D}\, f(x)[\xi(x)]$. *The vector field* $\nabla_\xi\,\eta$ *is called the covariant derivative of* η *with respect to* ξ *for the affine connection* ∇.

First of all, it should be noted that no Riemannian metric is mentioned in Definition 18. Thus, a smooth embedded submanifold is enough to define affine connections. A second remark is that Definition 18 extends the classical derivative of vector fields on a linear space $\mathcal{E}$. Indeed, for $\xi, \eta \in \mathfrak{X}(\mathcal{E})$

$$(\nabla_\xi\,\eta)_x = \lim_{t\to 0} \frac{\eta(x + t\xi(x)) - \eta(x)}{t} = \mathrm{D}\,\eta(x)[\xi(x)] \tag{2.9}$$

is an affine connection on $\mathcal{E}$. In practice, for a given manifold, many affine connections exist. The *fundamental theorem of Riemannian geometry* states that, given a Riemannian manifold, there is a unique affine connection that is *torsion-free* and is *compatible with the Riemannian metric*. Furthermore, this theorem gives an explicit formula, the *Koszul formula*, to compute this connection called the *Levi-Civita connection*.

Theorem 4 (Theorem 5.3.1 from [1])**.** *On a Riemannian manifold* $\mathcal{M}$ *there exists a unique connection* ∇ *that satisfies*

1. $\nabla_\eta\,\xi - \nabla_\xi\,\eta = [\eta, \xi]$ *(symmetry or tension-free), and*
2. $\chi\langle\eta, \xi\rangle = \langle\nabla_\chi\,\eta, \xi\rangle + \langle\eta, \nabla_\chi\,\xi\rangle$ *(compatibility with the Riemannian metric),*

for all $\chi, \eta, \xi \in \mathfrak{X}(\mathcal{M})$. $[.,.]$ *is the Lie bracket,* i.e. $[\xi, \eta] = \xi\eta - \eta\xi$. *This affine connection* ∇*, called the Levi-Civita connection or the Riemannian connection of* $\mathcal{M}$*, is characterized by the Koszul formula*

$$2\langle\nabla_\chi\,\eta, \xi\rangle = \chi\langle\eta, \xi\rangle + \eta\langle\xi, \chi\rangle - \xi\langle\chi, \eta\rangle - \langle\chi, [\eta, \xi]\rangle + \langle\eta, [\xi, \chi]\rangle + \langle\xi, [\chi, \eta]\rangle. \tag{2.10}$$

For all $\xi, \eta, \chi \in \mathcal{E}$, a Euclidean space (linear space endowed with the classical Euclidean metric), we get $\xi\langle\eta, \chi\rangle = \langle\xi\eta, \chi\rangle + \langle\eta, \xi\chi\rangle$ and the Koszul formula (2.10) reduces to $\langle\chi\eta, \xi\rangle = \langle\nabla_\chi\,\eta, \xi\rangle$. Thus, the affine connection of Equation 2.9 is the Levi-Civita connection.

Example 8. *The sphere* S^{d-1} *is a Riemannian manifold thus it has a unique Levi-Civita connection. Using the Koszul formula* (2.10)*, for all* $\boldsymbol{\eta}, \boldsymbol{\xi} \in \mathfrak{X}(\mathsf{S}^{d-1})$ *and* $\boldsymbol{x} \in \mathsf{S}^{d-1}$*, it is*

$$(\nabla^{\mathsf{S}^{d-1}}_{\boldsymbol{\xi}}\,\boldsymbol{\eta})_{\boldsymbol{x}} = P^{\mathsf{S}^{d-1}}_{\boldsymbol{x}}(\mathrm{D}\,\boldsymbol{\eta}(\boldsymbol{x})[\boldsymbol{\xi}(\boldsymbol{x})]) = (\boldsymbol{I}_d - \boldsymbol{x}\boldsymbol{x}^T)\,\mathrm{D}\,\boldsymbol{\eta}(\boldsymbol{x})[\boldsymbol{\xi}(\boldsymbol{x})].$$

All the Riemannian manifolds defined in the following are equipped with their Levi-Civita connections.

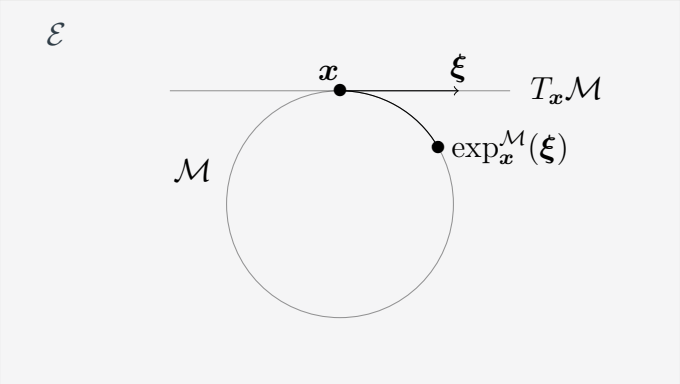


Figure 2.3: Illustration of the Riemannian manifold $\mathcal{M} = \mathsf{S}^1$, its tangent space $T_{\boldsymbol{x}}\mathcal{M}$ at a given $\boldsymbol{x} \in \mathcal{M}$, a tangent vector $\boldsymbol{\xi} \in T_{\boldsymbol{x}}\mathcal{M}$ and the exponential mapping $\exp_{\boldsymbol{x}}^{\mathcal{M}}$.

2.1.5 . Acceleration, geodesic and exponential map

This Levi-Civita connection allows us to introduce *geodesics*. In a linear space $\mathcal{E}$, the geodesic $\gamma : I \to \mathcal{E}$, I an open interval of $\mathbb{R}$, with initial conditions $\gamma(0) = x$ and $\dot{\gamma}(0) = \xi$ is the classical straight line $\gamma(t) = x + t\xi$. A characteristic of these straight lines is their zero *acceleration*: $\ddot{\gamma}(t) = 0$ for all $t \in I$. This acceleration is extended to Riemannian manifolds in Definition 19. Then, geodesics on manifolds are introduced in Definition 20 as $\mathcal{C}^2$ curves with zero acceleration.

Definition 19 (Definition 2.19 from [21])**.** *Let $\mathcal{M}$ be a Riemannian manifold with its Levi-Civita connection ∇. Let $\gamma : I \to \mathcal{M}$ with I an open interval of $\mathbb{R}$ be a $\mathcal{C}^2$ curve on $\mathcal{M}$. The acceleration along γ is given by:*

$$t \mapsto \nabla_{\dot{\gamma}(t)} \dot{\gamma}(t) \in T_{\gamma(t)}\mathcal{M}.$$

In order to respect Definition 18 of the connection, $\dot{\gamma}$ is supposed to be smoothly extended to an arbitrary vector field $X \in \mathfrak{X}(\mathcal{M})$ such that $X(\gamma(t)) = \dot{\gamma}(t)$ for all t.

Definition 20 (Definition 2.20 from [21])**.** *A smooth curve $\gamma : I \to \mathcal{M}$, with I an open interval of $\mathbb{R}$, is a geodesic if and only if it has zero acceleration on all its domain.*

Then, the *exponential mapping* is defined. This smooth map retracts a tangent vector ξ onto the manifold by following a geodesic with an initial direction ξ. The exponential mapping is illustrated in Figure 2.3.

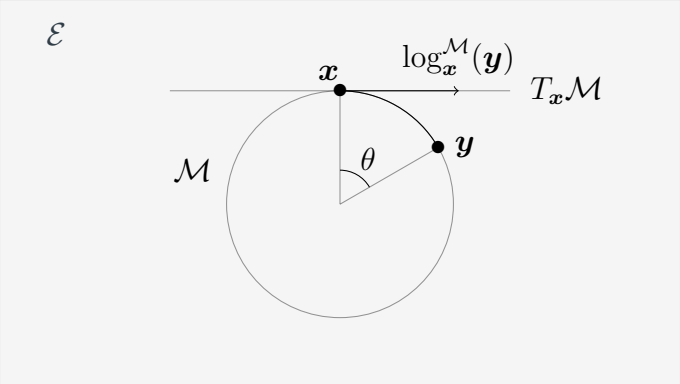


Figure 2.4: Illustration of the Riemannian manifold $\mathcal{M} = \mathsf{S}^1$, the logarithmic mapping $\log_{\boldsymbol{x}}^{\mathcal{M}}$ and the Riemannian distance $d_{\mathcal{M}}(\boldsymbol{x}, \boldsymbol{y}) = \left\|\log_{\boldsymbol{x}}^{\mathcal{M}}(\boldsymbol{y})\right\|_{\boldsymbol{x}} = \theta$.

Definition 21 (Definition 10.16 from [19])**.** *For every $(x, \xi) \in T\mathcal{M}$, let $\gamma_\xi : I \to \mathcal{M}$ be the unique geodesic with $\gamma_\xi(0) = x$, $\dot{\gamma}_\xi(0) = \xi$ and I as large as possible. Consider the following subset of the tangent bundle:*

$$\mathcal{O} = \{(x, \xi) \in T\mathcal{M} : \gamma_\xi \text{ is defined on an interval containing } [0, 1]\}.$$

The exponential map $\exp^{\mathcal{M}} : \mathcal{O} \to \mathcal{M}$ is defined by

$$\exp^{\mathcal{M}}(x, \xi) = \exp_x^{\mathcal{M}}(\xi) = \gamma_\xi(1).$$

The restriction $\exp_x^{\mathcal{M}}$ is defined on $\mathcal{O}_x = \{\xi \in T_x\mathcal{M} : (x, \xi) \in \mathcal{O}\}$.

Proposition 6 (Proposition 10.17 from [19])**.** *The exponential map is smooth on its domain $\mathcal{O}$, which is an open in $T\mathcal{M}$.*

Example 9. *Let S^{d-1} be the Riemannian manifold of the sphere in $\mathbb{R}^d$. The geodesic $\boldsymbol{\gamma} : \mathbb{R} \to \mathsf{S}^{d-1}$ with initial conditions $\boldsymbol{\gamma}(0) = \boldsymbol{x}$ and $\dot{\boldsymbol{\gamma}}(0) = \boldsymbol{\xi} \neq \mathbf{0}$ is*

$$\boldsymbol{\gamma}(t) = \cos(t \left\|\boldsymbol{\xi}\right\|)\boldsymbol{x} + \sin(t \left\|\boldsymbol{\xi}\right\|)\frac{\boldsymbol{\xi}}{\left\|\boldsymbol{\xi}\right\|}.$$

Indeed, for all $t \in \mathbb{R}$ $\boldsymbol{\gamma}(t)^T\boldsymbol{\gamma}(t) = 1$, $\boldsymbol{\gamma}$ respect the initial conditions and it has a zero acceleration. To verify this last assertion, we compute the second derivative of $\boldsymbol{\gamma}$ at t

$$\ddot{\boldsymbol{\gamma}}(t) = -\left\|\boldsymbol{\xi}\right\|^2 \boldsymbol{\gamma}(t).$$

Then, we check the zero acceleration

$$\nabla^{\mathsf{S}^{d-1}}_{\dot{\boldsymbol{\gamma}}(t)} \dot{\boldsymbol{\gamma}}(t) = (\boldsymbol{I}_d - \boldsymbol{\gamma}(t)\boldsymbol{\gamma}(t)^T)\ddot{\boldsymbol{\gamma}}(t) = -\left\|\boldsymbol{\xi}\right\|^2 (\boldsymbol{I}_d - \boldsymbol{\gamma}(t)\boldsymbol{\gamma}(t)^T)\boldsymbol{\gamma}(t) = \mathbf{0}.$$

It follows that the exponential mapping $\exp_{\boldsymbol{x}}^{\mathsf{S}^{d-1}} : T_{\boldsymbol{x}}\mathsf{S}^{d-1} \setminus \{\mathbf{0}\} \to \mathsf{S}^{d-1}$ *is*

$$\exp_{\boldsymbol{x}}^{\mathsf{S}^{d-1}}(\boldsymbol{\xi}) = \cos(\|\boldsymbol{\xi}\|)\boldsymbol{x} + \sin(\|\boldsymbol{\xi}\|)\frac{\boldsymbol{\xi}}{\|\boldsymbol{\xi}\|}.$$

In the case $\boldsymbol{\xi} = \mathbf{0}$, $\exp_{\boldsymbol{x}}^{\mathsf{S}^{d-1}}$ *is smoothly extended using the limit* $\frac{\sin(x)}{x} \to 1$ *when* $x \to 0$.

2.1.6 . Injectivity radius, logarithmic map and distance

Remarkably, given $x \in \mathcal{M}$, the exponential mapping $\exp_x^{\mathcal{M}}$ is locally a diffeomorphism around the origin 0_x of $T_x\mathcal{M}$. This means that there exists a neighborhood $U \subset T_x\mathcal{M}$ around 0_x such that $\exp_x^{\mathcal{M}}$ is a smooth one-to-one correspondence between U and $\exp_x^{\mathcal{M}}(U) \subset \mathcal{M}$. When it exists, the inverse map of the exponential mapping is called the *logarithmic map*. Given $x, y \in \mathcal{M}$, it returns the tangent vector $\xi \in T_x\mathcal{M}$ such that $\exp_x^{\mathcal{M}}(\xi) = y$. It is introduced in Definition 22 and is illustrated in Figure 2.4.

Definition 22 (Definition 10.20 from [19])**.** *For* $x \in \mathcal{M}$*, let* $\log_x^{\mathcal{M}}$ *denote the logarithmic map at* x*,*

$$\log_x^{\mathcal{M}}(y) = \underset{\xi \in \mathcal{O}_x}{\arg\min} \ \|\xi\|_x \ \text{subject to} \ \exp_x^{\mathcal{M}}(\xi) = y,$$

with domain such that this is uniquely defined.

Then, the *length of a curve* on a Riemannian manifold as well as the *Riemannian distance* are defined.

Definition 23 (Definitions 2.21 and 2.22 from [21])**.** *The length of a* $\mathcal{C}^1$ *curve,* $\gamma : [a, b] \to \mathcal{M}$*, on a Riemannian manifold is defined by*

$$\text{length}(\gamma) = \int_a^b \sqrt{\langle \dot{\gamma}(t), \dot{\gamma}(t) \rangle_{\gamma(t)}} dt = \int_a^b \|\dot{\gamma}(t)\|_{\gamma(t)} \, dt.$$

The geodesic distance on $\mathcal{M}$ *is given by*

$$d_{\mathcal{M}}(p, q) = \inf_{\gamma \in \Gamma} \text{length}(\gamma) \tag{2.11}$$

where Γ *is the set of* $\mathcal{C}^1$ *curves* $\gamma : [0, 1] \to \mathcal{M}$ *such that* $\gamma(0) = p$ *and* $\gamma(1) = q$*.*

A curve achieving the infimum (2.11) is called a *minimizing curve*. It should be noted that Definition 23 of the *Riemannian distance* does not make use of geodesics. Previously, we defined geodesics $\gamma : I \to \mathcal{M}$ as $\mathcal{C}^2$ curves with zero acceleration and remarkably these geodesics are locally minimizing curves. This means that for all $t \in I$, there exists a neighborhood $U \subset I$

containing t such that the geodesic restricted to U is a minimizing curve. Before going further, we define an open ball $B(x, \mathrm{inj}(x)) \subset T_x\mathcal{M}$ where the exponential mapping is a diffeomorphism. This domain is important for the following since, for all $\xi \in B(x, \mathrm{inj}(x))$, the curve $t \mapsto \exp_x^{\mathcal{M}}(t\xi)$ is a minimizing curve. This open ball as well as the *injectivity radius* $x \mapsto \mathrm{inj}(x)$ are introduced in Definition 24.

Definition 24 (Definition 10.19 from [19])**.** *The injectivity radius of a Riemannian manifold $\mathcal{M}$ at a point x, denoted by $\mathrm{inj}(x)$, is the supremum over radii $r > 0$ such that $\exp_x^{\mathcal{M}}$ is defined and is a diffeomorphism on the open ball*

$$B(x, r) = \{\xi \in T_x\mathcal{M} : \|\xi\|_x < r\}.$$

We now have all the tools to establish a link between the geodesic, the exponential map, the logarithmic map and the Riemannian distance This relationship is presented in Proposition 7.

Proposition 7 (Proposition 10.22 from [19])**.** *If $\|\xi\|_x < \mathrm{inj}(x)$, the geodesic $c(t) = \exp_x^{\mathcal{M}}(t\xi)$ on the interval $[0, 1]$ is the minimizing curve connecting x to $y = \exp_x^{\mathcal{M}}(\xi)$, unique up to parametrization. In particular, $d_{\mathcal{M}}(x, y) = \|\xi\|_x$, and $\log_x^{\mathcal{M}}(y) = \xi$.*

Example 10. *Let S^{d-1} be the Riemannian manifold of the sphere in $\mathbb{R}^d$. The objective is to find the logarithmic mapping. To do so, for $\boldsymbol{x}, \boldsymbol{y} \in \mathsf{S}^{d-1}$ such that $\boldsymbol{y} \neq \pm\boldsymbol{x}$, we look for $\boldsymbol{\xi} \in T_{\boldsymbol{x}}\mathsf{S}^{d-1}$ satisfying $\exp_{\boldsymbol{x}}^{\mathsf{S}^{d-1}}(\boldsymbol{\xi}) = \boldsymbol{y}$. First of all, we have*

$$\boldsymbol{x}^T\boldsymbol{y} = \cos(\|\boldsymbol{\xi}\|).$$

Thus, we get that

$$\boldsymbol{y} = (\boldsymbol{x}^T\boldsymbol{y})\boldsymbol{x} + \sin(\|\boldsymbol{\xi}\|)\frac{\boldsymbol{\xi}}{\|\boldsymbol{\xi}\|}.$$

This implies that the orthogonal projection of $\boldsymbol{y}$ onto $T_{\boldsymbol{x}}\mathsf{S}^{d-1}$ is proportional to $\boldsymbol{\xi}$

$$(\boldsymbol{I}_d - \boldsymbol{x}\boldsymbol{x}^T)\boldsymbol{y} = P_{\boldsymbol{x}}^{\mathsf{S}^{d-1}}(\boldsymbol{y}) = \sin(\|\boldsymbol{\xi}\|)\frac{\boldsymbol{\xi}}{\|\boldsymbol{\xi}\|}.$$

Thus the normalized projection is

$$\frac{P_{\boldsymbol{x}}^{\mathsf{S}^{d-1}}(\boldsymbol{y})}{\left\|P_{\boldsymbol{x}}^{\mathsf{S}^{d-1}}(\boldsymbol{y})\right\|} = \mathrm{sign}(\sin(\|\boldsymbol{\xi}\|))\frac{\boldsymbol{\xi}}{\|\boldsymbol{\xi}\|}$$

where sign is the sign function. Furthermore, if the domain of $\exp_{\boldsymbol{x}}^{\mathsf{S}^{d-1}}$ is restricted to $\boldsymbol{\xi}$ such that $\|\boldsymbol{\xi}\| < \pi$ we get that $\mathrm{sign}(\sin(\|\boldsymbol{\xi}\|)) = 1$, and

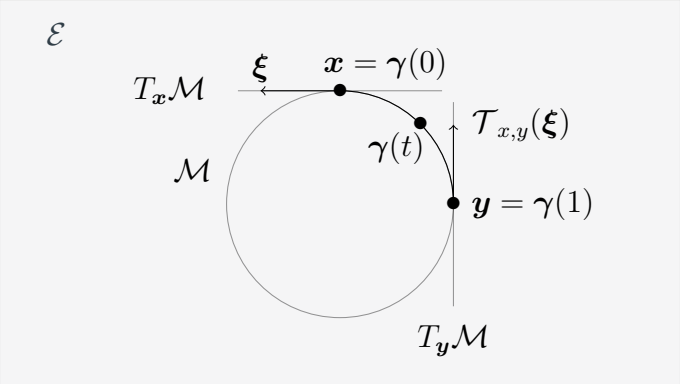


Figure 2.5: Illustration of the Riemannian manifold $\mathcal{M} = \mathsf{S}^1$ and the parallel transport of $\xi \in T_x\mathcal{M}$ to another tangent space $T_y\mathcal{M}$ along the geodesic γ.

$\boldsymbol{x}^T\boldsymbol{y} = \cos(\|\boldsymbol{\xi}\|)$ has a unique solution which is $\|\boldsymbol{\xi}\| = \arccos(\boldsymbol{x}^T\boldsymbol{y})$ where $\arccos : [-1, 1] \to [0, \pi]$. This implies that

$$\boldsymbol{\xi} = \arccos(\boldsymbol{x}^T\boldsymbol{y}) \frac{P_{\boldsymbol{x}}^{\mathsf{S}^{d-1}}(\boldsymbol{y})}{\left\| P_{\boldsymbol{x}}^{\mathsf{S}^{d-1}}(\boldsymbol{y}) \right\|}.$$

Since $\boldsymbol{\xi}$ is the unique solution,

$$\log_{\boldsymbol{x}}^{\mathsf{S}^{d-1}}(\boldsymbol{y}) = \arccos(\boldsymbol{x}^T\boldsymbol{y}) \frac{P_{\boldsymbol{x}}^{\mathsf{S}^{d-1}}(\boldsymbol{y})}{\left\| P_{\boldsymbol{x}}^{\mathsf{S}^{d-1}}(\boldsymbol{y}) \right\|}. \tag{2.12}$$

is the logarithmic mapping for $\boldsymbol{y} \neq \pm\boldsymbol{x}$ and $\log_{\boldsymbol{x}}^{\mathsf{S}^{d-1}}(\boldsymbol{y}) = \mathbf{0}$ for $\boldsymbol{y} = \boldsymbol{x}$. One can check that for all $\boldsymbol{y} \in \mathsf{S}^{d-1} \setminus \{-\boldsymbol{x}\}$

$$\exp_{\boldsymbol{x}}^{\mathsf{S}^{d-1}}(\log_{\boldsymbol{x}}^{\mathsf{S}^{d-1}}(\boldsymbol{y})) = \boldsymbol{y} \tag{2.13}$$

and conversely. Thus, $\exp_{\boldsymbol{x}}^{\mathsf{S}^{d-1}} : B(\boldsymbol{x}, \pi) \to \mathsf{S}^{d-1} \setminus \{-\boldsymbol{x}\}$ is a diffeomorphism and $d_{\mathsf{S}^{d-1}}(\boldsymbol{x}, \boldsymbol{y}) = \left\| \log_{\boldsymbol{x}}^{\mathsf{S}^{d-1}}(\boldsymbol{y}) \right\| = \arccos(\boldsymbol{x}^T\boldsymbol{y})$. Finally, it should be noted that $\boldsymbol{y} = -\boldsymbol{x}$ is the antipodal point of $\boldsymbol{x}$. Thus, there is an infinite number of $\boldsymbol{\xi}$ such that $\exp_{\boldsymbol{x}}^{\mathsf{S}^{d-1}}(\boldsymbol{\xi}) = \boldsymbol{y}$ and there is no logarithmic map for $\boldsymbol{y} = -\boldsymbol{x}$.

2.1.7 . Parallel transport

In Euclidean spaces, we are used to compare vectors, *e.g.* by computing an angle between them. However, these operations of comparison are not relevant on Riemannian manifolds for vectors that belong to different tangent spaces. Indeed, a vector ξ belonging to a given tangent space $T_x\mathcal{M}$ does

not necessarily belong to another tangent space $T_y\mathcal{M}$. Thus, it must first be "moved" to $T_y\mathcal{M}$ along a curve γ such that $\gamma(0) = x$ and $\gamma(1) = y$ before being compared to vectors of $T_y\mathcal{M}$. To do so, a vector field $\eta \in \mathfrak{X}(\mathcal{M})$ such that $\eta(\gamma(0)) = \xi$ that is parallel along γ is computed. By "constant" we mean that its covariant derivative with respect to $\dot{\gamma}$ along γ is zero. This property and the resulting parallel transport are formally presented in the next two definitions.

Definition 25. *Given a smooth curve γ on $\mathcal{M}$ with $\gamma(0) = x$ and $\gamma(1) = y$, the vector field $\eta \in \mathfrak{X}(\mathcal{M})$ is parallel along γ if for all $t \in [0, 1]$*

$$\nabla_{\dot{\gamma}(t)} \eta(\gamma(t)) = 0.$$

Definition 26. *Given a smooth curve γ on $\mathcal{M}$ with $\gamma(0) = x$ and $\gamma(1) = y$, the parallel transport of tangent vectors at x to the tangent space $T_y\mathcal{M}$ along γ is the map*

$$\mathcal{T}^{\mathcal{M}}_{x,y} : T_x\mathcal{M} \to T_y\mathcal{M}$$

defined by $\mathcal{T}^{\mathcal{M}}_{x,y}(\xi) = \eta(\gamma(1))$, where $\eta \in \mathfrak{X}(\mathcal{M})$ is a constant vector field along γ such that $\eta(\gamma(0)) = \xi$.

It should be noted that the application $\mathcal{T}^{\mathcal{M}}_{x,y}$ depends on the chosen curve γ ! In the following, when $\mathcal{M}$ is a Riemannian manifold, the chosen curve is the geodesic between x and y. This notion of parallel transport is illustrated with the sphere in $\mathbb{R}^d$ in Example 11 and in Figure 2.5.

Example 11. *Let S^{d-1} be the Riemannian manifold of the sphere in $\mathbb{R}^d$. From [114], the parallel transport of $\boldsymbol{\xi} \in T_{\boldsymbol{x}}\mathcal{M}$ along the geodesic $\boldsymbol{\gamma}$ such that $\boldsymbol{\gamma}(0) = \boldsymbol{x}, \boldsymbol{\gamma}(1) = \boldsymbol{y}$ and $\dot{\boldsymbol{\gamma}}(0) = \boldsymbol{\eta} = \log^{\mathsf{S}^{d-1}}_{\boldsymbol{x}}(\boldsymbol{y})$ is*

$$\mathcal{T}^{\mathsf{S}^{d-1}}_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{\xi}) = \left(\boldsymbol{I}_d + (\cos(\|\boldsymbol{\eta}\|) - 1)\frac{\boldsymbol{\eta}\boldsymbol{\eta}^T}{\|\boldsymbol{\eta}\|^2} - \sin(\|\boldsymbol{\eta}\|)\frac{\boldsymbol{x}\boldsymbol{\eta}^T}{\|\boldsymbol{\eta}\|} \right) \boldsymbol{\xi}.$$

2.2 . Elements of optimization on manifolds

In this section, we present algorithms to minimize smooth functions on manifolds, *i.e.*

$$\underset{x \in \mathcal{M}}{\text{minimize}} \; h(x) \tag{2.14}$$

where $\mathcal{M}$ is a manifold and h is a smooth scalar field, *i.e.* a smooth function from $\mathcal{M}$ to $\mathbb{R}$, called the *cost function*. In the following, we assume that h is lower bounded on $\mathcal{M}$. This assumption is met for most well posed optimization problems, otherwise this means that h has no minimum.

Assumption 1. *There exists $h^* \in \mathbb{R}$ such that $h(x) \geq h^*$ for all $x \in \mathcal{M}$.*

A problem (2.14) on the sphere manifold is presented in the following example.

Example 12. *Let $\boldsymbol{A}$ be a $p \times p$ symmetric matrix. A minimization problem on the sphere is*

$$\underset{\boldsymbol{x} \in \mathsf{S}^{d-1}}{\text{minimize}} \left\{ h(\boldsymbol{x}) = \frac{1}{2} \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} \right\}.$$

The cost function h respects Assumption 1 since it is continuous and S^{d-1} is compact.

Generally speaking, we are interested in problems (2.14) for which no closed form formula exists or is of no practical interest, *e.g.* for computational cost reasons. Since h is smooth, the general idea is to adapt classical gradient-based optimization algorithms developed for Euclidean spaces, such as *gradient descent* or *conjugate gradient*, to Riemannian manifolds.

2.2.1 . Gradient based optimization on manifolds

We begin with the definition of the *Riemannian gradient* which extends the definition of the gradient on Euclidean spaces to Riemannian manifolds.

Definition 27 (Definition 3.58 from [19])**.** *Let $h : \mathcal{M} \to \mathbb{R}$ be smooth on a Riemannian manifold $\mathcal{M}$. The Riemannian gradient of h is the vector field $\operatorname{grad}_{\mathcal{M}} h$ on $\mathcal{M}$ uniquely defined by the following identities:*

$$\forall (x, \xi) \in T\mathcal{M}, \quad \mathrm{D}\, h(x)[\xi] = \langle \operatorname{grad}_{\mathcal{M}} h(x), \xi \rangle_x, \tag{2.15}$$

where $\mathrm{D}\, h(x)$ is as in Equation (2.1) and $\langle ., . \rangle_x$ is the Riemannian metric.

So far, we only said we want to tackle the minimization problem (2.14) without additional specifications. The ideal objective would be to find the *global minimum* of h. However, this is a difficult task (as on Euclidean spaces) without any further assumption on h. A more realizable objective is to find the *critical points* of h on $\mathcal{M}$. This goal is classical in optimization on Euclidean spaces and here extended to Riemannian manifolds.

Definition 28. *A point $x \in \mathcal{M}$ is critical (or stationary) for a smooth function $h : \mathcal{M} \to \mathbb{R}$ if*

$$\operatorname{grad}_{\mathcal{M}} h(x) = 0.$$

Targeting these points gives a *necessary condition* for a point x to be a local minimizer. This is called the *first-order necessary optimality condition*. Indeed, any *local minimizer* of h is also a critical point.

Definition 29. *A point $x \in \mathcal{M}$ is a local minimizer of a function $h : \mathcal{M} \to \mathbb{R}$ if there exists a neighborhood U of x in $\mathcal{M}$ such that $h(y) \geq h(x)$ for all $y \in U$.*

Proposition 8 (Proposition 4.5 from [19]). *Any local minimizer of a smooth function $h : \mathcal{M} \to \mathbb{R}$ is a critical point of h.*

This necessary condition is applied on the cost function of Example 12 to find that the minimum is an eigenvector of the symmetric matrix $\boldsymbol{A}$.

Example 13. *We continue Example 12. To find the critical points, we compute the gradient of h. Given $\boldsymbol{\xi} \in T_{\boldsymbol{x}}\mathsf{S}^{d-1}$*

$$\begin{aligned} \mathrm{D}\, h(\boldsymbol{x})[\boldsymbol{\xi}] &= \boldsymbol{\xi}^T \boldsymbol{A}\boldsymbol{x} \\ &= \langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{\xi} \rangle \\ &= \langle (\boldsymbol{I}_d - \boldsymbol{x}\boldsymbol{x}^T)\boldsymbol{A}\boldsymbol{x}, \boldsymbol{\xi} \rangle. \end{aligned}$$

Thus, the gradient of h at $\boldsymbol{x}$ is $\operatorname{grad}_{\mathsf{S}^{d-1}} h(\boldsymbol{x}) = (\boldsymbol{I}_d - \boldsymbol{x}\boldsymbol{x}^T)\boldsymbol{A}\boldsymbol{x}$. By cancelling this gradient, we get the following necessary condition for $\boldsymbol{x} \in \mathsf{S}^{d-1}$ to be a minimum of h

$$\boldsymbol{A}\boldsymbol{x} = (\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x})\boldsymbol{x}.$$

Therefore, the minimum of h is met at the eigenvector of unit norm $\boldsymbol{x}$ associated with the lowest eigenvalue $\lambda = \boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x}$ of $\boldsymbol{A}$.

To find these critical points, gradient-based optimization algorithms on Euclidean spaces are adapted to Riemannian manifolds $\mathcal{M}$. The main difficulty comes from the non-linearity (in general) of $\mathcal{M}$. Indeed, a gradient descent step does not necessarily returns a point on $\mathcal{M}$, *i.e.* for a given iterate $x^{(l)} \in \mathcal{M}$ and a step size $\alpha > 0$

$$x^{(l+1)} = x^{(l)} - \alpha \operatorname{grad}_{\mathcal{M}} h(x^{(l)}) \notin \mathcal{M} \text{ (in general).} \tag{2.16}$$

To overcome this issue, we look for iterative algorithms that move along smooth curves $c : I \to \mathcal{M}$, with I an open interval of $\mathbb{R}$ around 0. At a given iterate $x^{(l)} \in \mathcal{M}$, if c is such that $c(0) = x^{(l)}$ and $h(c(\alpha)) < h(x^{(l)})$ for some step size $\alpha > 0$ then

$$x^{(l+1)} = c(\alpha) \in \mathcal{M} \tag{2.17}$$

and $h(x^{(l+1)}) < h(x^{(l)})$. Thus, we found a new iterate which belongs to the manifold and that decreases the value of h ! The challenge is to find such a curve c. In Section 2.1, we introduced the geodesic and the exponential map which are smooth maps to move on manifolds. Thus, they are good candidates for c. For all $\xi \in T_x\mathcal{M}$, $t \mapsto h(\exp_x^{\mathcal{M}}(t\xi))$ is smooth by composition, therefore it admits a Taylor expansion. Recalling that $\exp_x^{\mathcal{M}}(0) = x$ and $\frac{d}{dt} \exp_x^{\mathcal{M}}(t\xi)\Big|_{t=0} = \xi$, we get

$$h(\exp_x^{\mathcal{M}}(t\xi)) = h(x) + t\, \mathrm{D}\, h(x)[\xi] + \mathcal{O}(t^2). \tag{2.18}$$

Using Definition 27 of the Riemannian gradient, this Taylor expansion is rewritten

$$h(\exp_x^{\mathcal{M}}(t\xi)) = h(x) + t\langle \operatorname{grad}_{\mathcal{M}} h(x), \xi\rangle_x^{\mathcal{M}} + \mathcal{O}(t^2). \tag{2.19}$$

Thus for $\alpha > 0$ small enough, the cost function is decreased if and only if ξ is a descent direction, *i.e.*

$$h(\exp_x^{\mathcal{M}}(\alpha\xi)) - h(x) < 0 \iff \langle \operatorname{grad}_{\mathcal{M}} h(x), \xi\rangle_x^{\mathcal{M}} < 0. \tag{2.20}$$

Using (2.20), we build an iterative algorithm. At a given iterate $x^{(l)}$, let $\alpha > 0$ be a small enough step size and $\xi \in T_{x^{(l)}}\mathcal{M}$ be a descent direction, *i.e.* $\langle \operatorname{grad}_{\mathcal{M}} h(x), \xi\rangle_x^{\mathcal{M}} < 0$, the next iterate is given by

$$x^{(l+1)} = \exp_{x^{(l)}}^{\mathcal{M}}(\alpha\xi) \in \mathcal{M} \tag{2.21}$$

and $h(x^{(l+1)}) < h(x^{(l)})$. Before going further, we notice that we only used the smoothness, the initial position and speed of the exponential map to derive (2.21). Thus, any curves satisfying these properties could be used in place of the exponential mapping. This motivates the introduction of *retractions*.

Definition 30 (Definition 3.47 from [19])**.** *A retraction on a manifold $\mathcal{M}$ is a smooth map*

$$R^{\mathcal{M}} : T\mathcal{M} \to \mathcal{M} : (x, \xi) \mapsto R_x^{\mathcal{M}}(\xi)$$

such that each curve $c(t) = R_x^{\mathcal{M}}(t\xi)$ satisfies $c(0) = x$ and $\dot{c}(0) = \xi$.

It should be noted that exponential maps are retractions. Therefore, retractions generalize exponential maps and only respect the important properties to do optimization. Retractions can be developed when the exponential mapping is not available in closed form, or too expensive to compute or not stable numerically. An example of a retraction on the sphere in $\mathbb{R}^d$ is given.

Example 14. *Let $\boldsymbol{x} \in \mathsf{S}^{d-1}$ and $\boldsymbol{\xi} \in T_{\boldsymbol{x}}\mathsf{S}^{d-1}$, then a retraction is*

$$R_{\boldsymbol{x}}^{\mathsf{S}^{d-1}}(\boldsymbol{\xi}) = \frac{\boldsymbol{x} + \boldsymbol{\xi}}{\|\boldsymbol{x} + \boldsymbol{\xi}\|}.$$

Given a retraction on $\mathcal{M}$, (2.21) is rewritten

$$x^{(l+1)} = R_{x^{(l)}}^{\mathcal{M}}(\alpha\xi). \tag{2.22}$$

In order to implement (2.22), it remains to provide a descent direction $\xi \in T_{x^{(l)}}\mathcal{M}$. Many possibilities exist, two of them are presented in the following. The first one implements the *Riemannian gradient descent* algorithm and the second the *Riemannian conjugate gradient* algorithm.

Algorithm 3: Riemannian gradient descent

Input: Initialization: $x^{(0)} \in \mathcal{M}$
Output: $x^{(l)} \in \mathcal{M}$
for $l = 0$ ***to convergence*** **do**
  $\xi^{(l)} = -\operatorname{grad}_{\mathcal{M}} h(x^{(l)})$
  $\alpha = \text{Linesearch}(x^{(l)}, \xi^{(l)})$
  $x^{(l+1)} = R^{\mathcal{M}}_{x^{(l)}}\left(\alpha \xi^{(l)}\right)$

2.2.2 . Riemannian gradient descent

A first descent direction to implement (2.22) is $\xi = -\operatorname{grad}_{\mathcal{M}} h(x^{(l)})$. Indeed, it is a descent direction since

$$\langle \operatorname{grad}_{\mathcal{M}} h(x^{(l)}), -\operatorname{grad}_{\mathcal{M}} h(x^{(l)}) \rangle^{\mathcal{M}}_{x^{(l)}} = -\left\| \operatorname{grad}_{\mathcal{M}} h(x^{(l)}) \right\|^2_{x^{(l)}} < 0. \quad (2.23)$$

Using this descent direction along with (2.22) is the *Riemannian gradient descent* described in Algorithm 3. It can be proven that an iterate $x^{(l)}$ with an arbitrary small gradient can be found using this algorithm. To get this result, a second assumption is added on the decrease at each iteration of the cost function.

Assumption 2. *At each iteration, the algorithm achieves sufficient decrease for h, in that there exists a constant $c > 0$ such that, for all k,*

$$h(x^{(l+1)}) - h(x^{(l)}) \leq -c \left\| \operatorname{grad}_{\mathcal{M}} h(x^{(l)}) \right\|^2_{x^{(l)}}.$$

Some conditions on the *pullback function* $h \circ R^{\mathcal{M}} : T\mathcal{M} \to \mathbb{R}$ can be added to ensure that Assumption 2 is met; see [19, Chapter 4] for a detailed discussion. In practice, a line-search looks for a step size such that Assumption 2 is respected. Indeed, classical Euclidean line-searches such as the backtracking one have their Riemannian counterparts; see [1, Chapter 4]. Then, when both Assumptions 1 and 2 are met, the next proposition states that, as desired, the norm of the gradient tends to zero as the iteration number tends to the infinite. Furthermore, it gives a non-asymptotic convergence rate of $\mathcal{O}\left(\frac{1}{\sqrt{L}}\right)$ for the gradient norm. It should be noted that both results are without conditions on the initialization.

Proposition 9 (Proposition 4.7 from [19])**.** *Let h be a smooth function satisfying Assumption 1 on a Riemmannian manifold $\mathcal{M}$. Let $x^{(0)}, x^{(1)}, x^{(2)}, \cdots$ be iterates satisfying Assumption 2 with constant c. Then,*

$$\lim_{k \to +\infty} \left\| \operatorname{grad}_{\mathcal{M}} h(x^{(l)}) \right\|_{x^{(l)}} = 0.$$

Furthermore, for all $L \geq 1$, there exists l in $0, \cdots, L-1$ such that

$$\left\| \operatorname{grad}_{\mathcal{M}} h(x^{(l)}) \right\|_{x^{(l)}} \leq \sqrt{\frac{h(x^{(0)}) - h^*}{c}} \frac{1}{\sqrt{L}}.$$

Algorithm 4: Riemannian conjugate gradient

Input: Initialization: $x^{(0)} \in \mathcal{M}$, $\xi^{(0)} = -g^{(0)} = -\operatorname{grad}_{\mathcal{M}} h(x^{(0)})$
Output: $x^{(l)} \in \mathcal{M}$
for $l = 0$ ***to convergence*** **do**
  if $\langle g^{(l)}, \xi^{(l)} \rangle_{x^{(l)}} \geq 0$ **then**
    $\xi^{(l)} = -g^{(l)}$
  $\alpha = \text{Linesearch}(x^{(l)}, \xi^{(l)})$
  $x^{(l+1)} = R^{\mathcal{M}}_{x^{(l)}}\left(\alpha \xi^{(l)}\right)$
  $g^{(l+1)} = \operatorname{grad}_{\mathcal{M}} h(x^{(l+1)})$
  $\xi^{(l)}_{\mathcal{T}} = \mathcal{T}^{\mathcal{M}}_{x^{(l)}, x^{(l+1)}}(\xi^{(l)})$
  $g^{(l)}_{\mathcal{T}} = \mathcal{T}^{\mathcal{M}}_{x^{(l)}, x^{(l+1)}}(g^{(l)})$
  $\beta = \max\left(0, \frac{\left\langle g^{(l+1)} - g^{(l)}_{\mathcal{T}}, g^{(l+1)} \right\rangle_{x^{(l+1)}}}{\left\langle g^{(l+1)} - g^{(l)}_{\mathcal{T}}, \xi^{(l)}_{\mathcal{T}} \right\rangle_{x^{(l+1)}}}\right)$
  $\xi^{(l+1)} = -g^{(l+1)} + \beta \xi^{(l)}_{\mathcal{T}}$

2.2.3 . Riemannian conjugate gradient

In the previous subsection, we presented the Riemannian gradient descent algorithm which gives the following iterate for a given $x^{(l)}$,

$$x^{(l+1)} = R^{\mathcal{M}}_{x^{(l)}}(\alpha \xi^{(l)}) \tag{2.24}$$

where α is a small enough step size and $\xi^{(l)} = -\operatorname{grad}_{\mathcal{M}} h(x^{(l)})$. The Riemannian gradient descent is the simplest gradient-based optimization algorithm on manifold but empirically suffers from a convergence that can be slow. To alleviate this problem, other descent directions can be used in (2.24). For example, the *Riemannian conjugate gradient* proposes to add some inertia. As in (2.16), the non-linearity of the Riemannian manifold requires to adapt the classical conjugate gradient. Indeed, on a Euclidean space, the conjugate gradient linearly combines the gradient of h at $x^{(l)}$ and the descent direction $\xi^{(l-1)}$. This cannot be done on a Riemannian manifold since, in general, $T_{x^{(l)}}\mathcal{M} \neq T_{x^{(l-1)}}\mathcal{M}$. Thus, the descent direction $\xi^{(l-1)}$ is first transported to $T_{x^{(l)}}\mathcal{M}$ using the parallel transport $\mathcal{T}^{\mathcal{M}}_{x^{(l-1)}, x^{(l)}} : T_{x^{(l-1)}}\mathcal{M} \to T_{x^{(l)}}\mathcal{M}$ and then linearly combined to the gradient of h at $x^{(l)}$

$$\xi^{(l)} = -\operatorname{grad}_{\mathcal{M}} h(x^{(l)}) + \beta \, \mathcal{T}^{\mathcal{M}}_{x^{(l-1)}, x^{(l)}}(\xi^{(l-1)}) \tag{2.25}$$

where $\beta > 0$. It should be noted that $\xi^{(l)}$ is not necessarily a descent direction. In this case, β is set to 0 and thus $\xi^{(l)} = -\operatorname{grad}_{\mathcal{M}} h(x^{(l)})$ which is the descent direction of the Riemannian gradient descent. This Riemannian conjugate gradient is presented in Algorithm 4. In this algorithm, the inertia

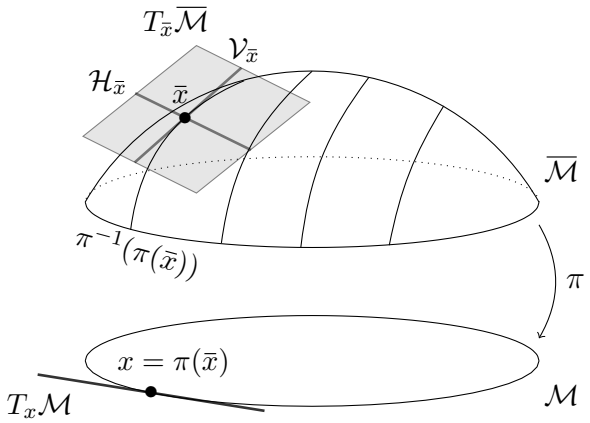


Figure 2.6: Illustration of the quotient manifold $\mathcal{M}$ represented by elements of $\overline{\mathcal{M}}$. The set of all representations of $x = \pi(\bar{x}) \in \mathcal{M}$ is the equivalence class $\pi^{-1}(\pi(\bar{x})) \subset \overline{\mathcal{M}}$. The tangent space $T_{\bar{x}}\overline{\mathcal{M}}$ is decomposed into the vertical space $\mathcal{V}_{\bar{x}} = T_x\pi^{-1}(\pi(\bar{x}))$ and its orthogonal complement, the horizontal space $\mathcal{H}_{\bar{x}}$, which provides proper representatives for tangent vectors in $T_x\mathcal{M}$.

parameter β is computed using the *Hestenes-Stiefel rule*. Others could be used, *e.g.* see the survey [61]. A last remark is that we used the parallel transport in Equation (2.25), however it is not always available in closed form or can be expensive to compute. Previously, we said that the retraction generalizes the exponential mapping. This leads to cheaper and easier to derive formulas to move on the manifold while keeping the important properties for optimization. In the same way, a generalization of the parallel transport is the *vector transport*; see [1, Chapter 8] for more details.

2.3 . Riemannian quotient manifolds

Riemannian quotient manifolds are a ubiquitous tool when optimizing functions with symmetries on Riemannian manifolds. A classical problem is the estimation of a subspace whose orthogonal basis is defined up to rotations. This problem is presented later on in this section; see [151] for other functions with symmetries. Riemannian quotient manifolds are an advanced topic and thus requires the introduction of many other concepts of Riemannian geometry to be well defined. The reader is referred to [1, Chapter 3] or [19, Chapters 8 and 9] for a proper introduction. In this section, we focus on the practical aspect, *i.e.* how to recognize these manifolds and how to manipulate their elements.

2.3.1 . Some elements on Riemannian quotient manifolds

Let $\overline{\mathcal{M}}$ be a Riemannian manifold embedded in a linear space $\mathcal{E}$ with the Riemannian metric at $\bar{x} \in \overline{\mathcal{M}}$, $(\bar{\xi}, \bar{\eta}) \in T_{\bar{x}}\overline{\mathcal{M}} \times T_{\bar{x}}\overline{\mathcal{M}} \mapsto \langle \bar{\xi}, \bar{\eta} \rangle_{\bar{x}}^{\overline{\mathcal{M}}}$. Riemannian quotient manifolds arise when some points are "equivalent", *e.g.* rotations of $\overline{\mathcal{M}}$ that leave the values of a function $\bar{h} : \overline{\mathcal{M}} \to \mathbb{R}$ unchanged. In this case, points of $\overline{\mathcal{M}}$ can be grouped together to form a new set $\mathcal{M}$. We begin by recalling some definitions related to *quotient sets*. Let $\sim$ be an *equivalence relation*, *i.e.* a *binary relation* on $\overline{\mathcal{M}}$ that is *reflexive*, *symmetric*, and *transitive*. Then the corresponding *equivalence classes* are

$$[\bar{x}] = \left\{ \bar{y} \in \overline{\mathcal{M}} : \bar{y} \sim \bar{x} \right\}, \tag{2.26}$$

and the associated quotient set is defined by

$$\mathcal{M} = \overline{\mathcal{M}} / \sim = \left\{ [\bar{x}] \in \overline{\mathcal{M}} : \bar{x} \in \overline{\mathcal{M}} \right\}. \tag{2.27}$$

The *natural* (or *canonical*) *projection* π associates points of $\overline{\mathcal{M}}$ to those of $\mathcal{M}$

$$\pi(\bar{x}) = [\bar{x}]. \tag{2.28}$$

If $\mathcal{M}$ respect some properties, then it admits a unique structure that turns it into a *quotient manifold*. Elements of a quotient manifold, such as points and tangent vectors, are abstracts. Thus, we leverage the elements of $\overline{\mathcal{M}}$, that are easily handled, to work with elements of $\mathcal{M}$. For example, an equivalence class $x \in \mathcal{M}$ is represented by a point $\bar{x} \in \overline{\mathcal{M}}$ such that $\pi(\bar{x}) = x$. Admitting that the tangent space $T_x\mathcal{M}$ is properly defined, this asks the question of the representation of its points. To do so, the tangent space $T_{\bar{x}}\overline{\mathcal{M}}$ is decomposed into two orthogonal subspaces. First of all, the vertical space is defined as the tangent space to the equivalent class $\pi^{-1}(\pi(\bar{x})) \subset \overline{\mathcal{M}}$ and thus collects tangent vectors that "have no effect on $\pi(\bar{x})$"

$$\mathcal{V}_{\bar{x}} = T_{\bar{x}}\pi^{-1}(\pi(\bar{x})) \subset T_{\bar{x}}\overline{\mathcal{M}}. \tag{2.29}$$

Then, the horizontal space is defined as the orthogonal complement of $\mathcal{V}_{\bar{x}}$ in $T_{\bar{x}}\overline{\mathcal{M}}$

$$\mathcal{H}_{\bar{x}} = \{ \bar{\xi} \in T_{\bar{x}}\overline{\mathcal{M}} : \langle \bar{\xi}, \bar{\eta} \rangle_{\bar{x}}^{\overline{\mathcal{M}}} = 0 \text{ for all } \bar{\eta} \in \mathcal{V}_{\bar{x}} \}. \tag{2.30}$$

Thus, $\mathcal{V}_{\bar{x}}$ and $\mathcal{H}_{\bar{x}}$ are in direct sum in the tangent space of $\overline{\mathcal{M}}$ at $\bar{x}$, *i.e.* $T_{\bar{x}}\overline{\mathcal{M}} = \mathcal{V}_{\bar{x}} + \mathcal{H}_{\bar{x}}$ and orthogonal projections $P_{\bar{x}}^{\mathcal{V}} : \mathcal{E} \to \mathcal{V}_{\bar{x}}$ and $P_{\bar{x}}^{\mathcal{H}} : \mathcal{E} \to \mathcal{H}_{\bar{x}}$ can be defined. Figure 2.6 illustrates these concepts. Then, it can be shown that for each element $\xi \in T_{\pi(\bar{x})}\mathcal{M}$, there exists a unique $\bar{\xi} \in \mathcal{H}_{\bar{x}}$ such that $\xi = \mathrm{D}\,\pi(\bar{x})[\bar{\xi}]$ (admitting that this directional derivative is well defined). $\bar{\xi}$ is called the *horizontal lift* of ξ at the *lifting point* $\bar{x}$ and can also be denoted $\mathrm{lift}_{\bar{x}}(\xi)$. Thus, every $\xi \in T_{\pi(\bar{x})}\mathcal{M}$ is represented by a unique element

$\bar{\xi} \in \mathcal{H}_{\bar{x}}$. If for every $x \in \mathcal{M}$ and every $\xi, \eta \in T_x\mathcal{M}$, the inner product $(\bar{\xi}, \bar{\eta}) \mapsto \langle \bar{\xi}, \bar{\eta} \rangle_{\bar{x}}^{\overline{\mathcal{M}}}$ does not depend on the lifting point $\bar{x}$, *i.e.*

$$\bar{x} \sim \bar{y} \implies \langle \mathrm{lift}_{\bar{x}}(\xi), \mathrm{lift}_{\bar{x}}(\eta) \rangle_{\bar{x}}^{\overline{\mathcal{M}}} = \langle \mathrm{lift}_{\bar{y}}(\xi), \mathrm{lift}_{\bar{y}}(\eta) \rangle_{\bar{y}}^{\overline{\mathcal{M}}} \tag{2.31}$$

then

$$\langle \xi, \eta \rangle_x^{\mathcal{M}} = \langle \mathrm{lift}_{\bar{x}}(\xi), \mathrm{lift}_{\bar{x}}(\eta) \rangle_{\bar{x}}^{\overline{\mathcal{M}}} \tag{2.32}$$

defines a Riemannian metric on $\mathcal{M}$ and $\mathcal{M}$ becomes a Riemannian quotient manifold of $\overline{\mathcal{M}}$.

2.3.2 . Optimization on Riemannian quotient manifolds

In the following chapters, we are mainly focused on optimization when dealing with Riemannian quotient manifolds. These minimization problems write

$$\underset{\bar{x} \in \overline{\mathcal{M}}}{\mathrm{minimize}}\ \bar{h}(\bar{x}) \tag{2.33}$$

for a cost function $\bar{h} : \overline{\mathcal{M}} \to \mathbb{R}$ invariant along equivalence classes *i.e.*

$$\bar{h}(\bar{x}) = \bar{h}(\bar{y}) \text{ for all } \bar{x} \sim \bar{y}. \tag{2.34}$$

Thus, we are only interested in the equivalence classes $x = \pi(\bar{x})$ and not in the elements $\bar{x}$ of $\overline{\mathcal{M}}$. Formally, (2.33) is rewritten

$$\underset{x \in \mathcal{M}}{\mathrm{minimize}}\ h(x). \tag{2.35}$$

with h such that $\bar{h} = h \circ \pi : \overline{\mathcal{M}} \to \mathbb{R}$. The next example illustrates the presented tools with a subspace estimation problem.

Example 15. *The objective of this example is to motivate the introduction to Riemannian quotient manifolds and to show how problems on these sets can arise. First of all, the previously presented sphere* $\mathsf{S}^{p-1} \subset \mathbb{R}^p$ *can be seen be seen as the set of orthogonal bases of* 1*-dimensional subspaces in* $\mathbb{R}^p$*. A natural extension is the set of orthogonal bases of* k*-dimensional subspaces in* $\mathbb{R}^p$

$$\mathsf{St}_{p,k} = \left\{ \boldsymbol{U} \in \mathbb{R}^{p \times k} : \boldsymbol{U}^T \boldsymbol{U} = \boldsymbol{I}_k \right\}.$$

This set can be endowed with a Riemannian structure and thus becomes a Riemannian manifold called the Stiefel manifold*. It should be noted that for* $k = 1$*, the Stiefel manifold coincides with the Riemannian manifold of the sphere* S^{p-1} *and for* $k = p$*, it coincides with the* orthogonal group $\mathcal{O}_p$*. Then, for* $n > p$*, we assume having a data matrix* $\boldsymbol{X} \in \mathbb{R}^{p \times n}$ *(*n *data vectors in* $\mathbb{R}^p$ *concatenated). A classical problem, in machine learning and signal processing, is to look for a* k*-dimensional subspace represented by*

an orthogonal basis $\boldsymbol{U} \in \mathsf{St}_{p,k}$ for which projected data are close to the original ones, i.e.*,*

$$\underset{\boldsymbol{u}\in\mathsf{St}_{p,k}}{\text{minimize}} \left\{ \bar{h}(\boldsymbol{U}) = \frac{1}{2} \left\| \boldsymbol{X} - \boldsymbol{U}\boldsymbol{U}^T\boldsymbol{X} \right\|_2^2 \right\}. \tag{2.36}$$

This minimization problem is solved by the Principal Component Analysis *and has a closed form solution: the k orthogonal eigenvectors of $\boldsymbol{X}\boldsymbol{X}^T$ associated with the k highest eigenvalues. In this example, we are interested in the parameter space of $\bar{h}$. A first remark is that $\bar{h}$ has a symmetry, indeed for all $\boldsymbol{U} \in \mathsf{St}_{p,k}$*

$$\bar{h}(\boldsymbol{U}\boldsymbol{O}) = \bar{h}(\boldsymbol{U}) \text{ for all } \boldsymbol{O} \in \mathcal{O}_k.$$

Thus, it is interesting to consider the following equivalence relation

$$\boldsymbol{U} \sim \boldsymbol{U}' \iff \boldsymbol{U}\boldsymbol{O} = \boldsymbol{U}' \text{ for some } \boldsymbol{O} \in \mathcal{O}_k,$$

which induces the following equivalent classes in $\mathsf{St}_{p,k}$

$$[\boldsymbol{U}] = \{\boldsymbol{U}' \in \mathsf{St}_{p,k} : \boldsymbol{U}' \sim \boldsymbol{U}\}.$$

Then, the associated quotient set is

$$\mathsf{Gr}_{p,k} = \mathsf{St}_{p,k}/\sim = \{[\boldsymbol{U}] : \boldsymbol{U} \in \mathsf{St}_{p,k}\}.$$

This set is a Riemannian quotient manifold, is presented in details in Section 2.4 and is called the Grassmann manifold*. Considering the canonical projection $\pi : \boldsymbol{U} \mapsto [\boldsymbol{U}]$, the symmetry of $\bar{h}$ is removed with $h : \mathsf{Gr}_{p,k} \to \mathbb{R}$ such that $\bar{h} = h \circ \pi$. Finally, the optimization problem* (2.36) *can be rewritten as a minimization over equivalence classes*

$$\underset{\pi(\boldsymbol{U})\in\mathsf{Gr}_{p,k}}{\text{minimize}}\ h(\pi(\boldsymbol{U})). \tag{2.37}$$

We will see later that the dimension of the parameter space in (2.36) *is $pk - \frac{k(k+1)}{2}$ whereas in* (2.37) *it is $(p-k)k$. This way, $\frac{k(k-1)}{2}$ dimensions have been removed (which is the dimension of $\mathcal{O}_k$).*

As presented in previous sections, to perform first order Riemannian optimization, we essentially need two tools: the Riemannian gradient and a retraction. The Riemannian gradient of h at $x = \pi(\bar{x})$, denoted $\operatorname{grad}_{\mathcal{M}} h(x)$ and that belongs to $T_x\mathcal{M}$, is represented by the Riemannian gradient $\operatorname{grad}_{\overline{\mathcal{M}}} \bar{h}(\bar{x}) \in \mathcal{H}_{\bar{x}}$ of $\bar{h}$ at $\bar{x}$. By definition, the gradient is the only tangent vector in $T_{\bar{x}}\overline{\mathcal{M}}$ satisfying

$$\mathrm{D}\,\bar{h}(\bar{x})[\bar{\xi}] = \langle \operatorname{grad}_{\overline{\mathcal{M}}} \bar{h}(\bar{x}), \bar{\xi} \rangle_{\bar{x}}^{\overline{\mathcal{M}}} \text{ for all } \bar{\xi} \in T_{\bar{x}}\overline{\mathcal{M}}. \tag{2.38}$$

Note that this vector always belongs to the horizontal space $\mathcal{H}_{\bar{x}}$ due to the invariance of $\bar{h}$ along equivalence classes. To obtain a point on $\mathcal{M}$ from a descent direction (represented by a vector in $\mathcal{H}_{\bar{x}}$), we need a retraction, *i.e.*, a map $R^{\mathcal{M}} : T\mathcal{M} \to \mathcal{M}$. Let $\overline{R}^{\overline{\mathcal{M}}}$ be a retraction on $\overline{\mathcal{M}}$ such that for all $x \in \mathcal{M}$ and $\xi \in T_x\mathcal{M}$

$$\pi(\overline{R}_{\bar{x}}^{\overline{\mathcal{M}}}(\mathrm{lift}_{\bar{x}}(\xi))) = \pi(\overline{R}_{\bar{y}}^{\overline{\mathcal{M}}}(\mathrm{lift}_{\bar{y}}(\xi))) \tag{2.39}$$

for all $\bar{x}, \bar{y} \in \pi^{-1}(x)$ with $\mathrm{lift}_{\bar{x}}(\xi)$ and $\mathrm{lift}_{\bar{y}}(\xi)$ being the horizontal lifts of ξ at $\bar{x}$ and $\bar{y}$ respectively. Then,

$$R_x^{\mathcal{M}}(\xi) = \pi(\overline{R}_{\bar{x}}^{\overline{\mathcal{M}}}(\mathrm{lift}_{\bar{x}}(\xi))) \tag{2.40}$$

defines a retraction on $\mathcal{M}$.

We conclude this section by pointing out that all the Riemannian tools defined in the previous sections such as the Levi-Civita connection, the exponential map, the logarithmic map and the Riemannian distance can be extended to Riemannian quotient manifolds. These extensions are made in the same spirit as what we have just done for the retraction: the Levi-Civita connection on $\mathcal{M}$ is represented by a vector field on $\overline{\mathcal{M}}$, the geodesic on $\mathcal{M}$ is represented by a geodesic on $\overline{\mathcal{M}}$ and so on. We do not go into more details since, for these Riemannian quotient manifolds, we use already established results for these tools; for example see the Grassmann manifold presented in Section 2.4.

2.4 . Some important Riemannian manifolds: $\mathcal{S}_p^{++}$, $\mathcal{SS}_p^{++}$, $(\mathbb{R}_*^+)^n$ and $\mathsf{Gr}_{p,k}$

This section aims to present examples of Riemannian manifolds that will be used throughout the manuscript. Each manifold will be presented in details as well as the tools of interest for optimization. The presented manifolds are:

- the manifold of $p \times p$ symmetric positive definite matrices with the affine invariance Riemannian metric denoted $\mathcal{S}_p^{++}$,
- the manifold of $p \times p$ symmetric positive definite matrices with unit determinant denoted $\mathcal{SS}_p^{++}$,
- the manifold of n-dimensional strictly positive vectors denoted $(\mathbb{R}_*^+)^n$,
- the compact Stiefel manifold denoted $\mathsf{St}_{p,k}$,
- and the Grassmann quotient manifold of k-dimensional subspaces in $\mathbb{R}^p$ denoted $\mathsf{Gr}_{p,k}$.

2.4.1 . $\mathcal{S}_p^{++}$: manifold of symmetric positive definite matrices

An important Riemannian manifold is the one of symmetric positive definite matrices with the affine invariant metric. Before introducing this Riemannian manifold, we motivate its increasing use in the literature over the years. First of all, an abundant literature has developed around optimization on this Riemannian manifold. Indeed, recent theoretical advances have shown that some non-convex problems are geodesically convex on this manifold (*i.e.* convex along the geodesics). This geodesic convexity, abbreviated g-convexity, gives interesting properties on first order stationary points (zero gradient) similar to convexity and thus allows global optimization. Detailed presentations of the concept of g-convexity are made in [19, Chapter 11] and [122]. Moreover, this g-convexity gives fast optimization algorithms and therefore convenient to use [148]. Examples of applications of these g-convexity properties are covariance estimation [140], Gaussian mixtures estimation [68, 67], metric learning [146] and geometric mean computation [94]. In addition to its relevance in optimization, the Riemannian manifold of symmetric positive definite matrices has been successfully used in many covariance-based applications such as brain-computer interface classification [8] and detection of pedestrians [134] or in signal processing for diffusion tensor magnetic resonance imaging [53]. We now turn to the description of this Riemannian manifold. Only the main tools are presented. Detailed descriptions can be found in [120, 113, 15].

First of all, the sets of $p \times p$ symmetric matrices and $p \times p$ symmetric positive definite matrices are defined as

$$\mathcal{S}_p = \left\{ \boldsymbol{\Sigma} \in \mathbb{R}^{p \times p} : \boldsymbol{\Sigma}^T = \boldsymbol{\Sigma} \right\}, \tag{2.41}$$

and

$$\mathcal{S}_p^{++} = \left\{ \boldsymbol{\Sigma} \in \mathcal{S}_p : \forall \boldsymbol{x} \in \mathbb{R}^p \setminus \{\mathbf{0}\}, \, \boldsymbol{x}^T \boldsymbol{\Sigma} \boldsymbol{x} > 0 \right\} \tag{2.42}$$

respectively. Thus, $\mathcal{S}_p$ is a linear space in the ambient space $\mathbb{R}^{p \times p}$ and $\mathcal{S}_p^{++}$ is an open in $\mathcal{S}_p$. Thus and by definition, $\mathcal{S}_p^{++}$ is a smooth embedded submanifold of $\mathcal{S}_p$. This induces that the tangent space at $\boldsymbol{\Sigma} \in \mathcal{S}_p^{++}$ is

$$T_{\boldsymbol{\Sigma}} \mathcal{S}_p^{++} = \mathcal{S}_p = \left\{ \boldsymbol{\xi} \in \mathbb{R}^{p \times p} : \boldsymbol{\xi}^T = \boldsymbol{\xi} \right\}. \tag{2.43}$$

Then, every tangent space $T_{\boldsymbol{\Sigma}} \mathcal{S}_p^{++}$ is equipped with the following Riemannian metric, for all $\boldsymbol{\xi}, \boldsymbol{\eta} \in T_{\boldsymbol{\Sigma}} \mathcal{S}_p^{++}$

$$\langle \boldsymbol{\xi}, \boldsymbol{\eta} \rangle_{\boldsymbol{\Sigma}}^{\mathcal{S}_p^{++}} = \mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1} \boldsymbol{\xi} \boldsymbol{\Sigma}^{-1} \boldsymbol{\eta} \right). \tag{2.44}$$

It is sometimes referred to the affine invariant Riemannian metric due to its invariance to affine transformations, *i.e.*

$$\langle \mathrm{D}\, \phi_{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma})[\boldsymbol{\xi}], \mathrm{D}\, \phi_{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma})[\boldsymbol{\eta}] \rangle_{\phi_{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma})}^{\mathcal{S}_p^{++}} = \langle \boldsymbol{\xi}, \boldsymbol{\eta} \rangle_{\boldsymbol{\Sigma}}^{\mathcal{S}_p^{++}} \tag{2.45}$$

where $\phi_{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma}) = \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^T$ with $\boldsymbol{A} \in \mathrm{GL}_p$. It should be noted that many other Riemannian metrics exist on $\mathcal{S}_p^{++}$ such as the log-Euclidean metric, the Bures-Wasserstein metric or the Bogoliubov-Kubo-Mori metric [6, 130]. However, the affine invariant metric (2.44) is proportional to the *Fisher information metric* [1] associated with the centered Gaussian distributions and thus is closely related to the statistical models we study in this manuscript. The presented geometry is sometimes referred as the *information geometry* of the centered Gaussian distributions, see [3] for a presentation of the information geometry. An additional remark is that it is a particular case of a class of affine invariant metrics, see [112] for more details. Then the orthogonal projection from $\mathbb{R}^{p\times p}$ onto $T_{\boldsymbol{\Sigma}}\mathcal{S}_p^{++}$ is

$$P_{\boldsymbol{\Sigma}}^{\mathcal{S}_p^{++}}(\boldsymbol{\xi}) = \mathrm{sym}(\boldsymbol{\xi}) \tag{2.46}$$

where $\mathrm{sym}(\boldsymbol{\xi}) = \frac{1}{2}\left(\boldsymbol{\xi} + \boldsymbol{\xi}^T\right)$. For two smooth vector fields $\boldsymbol{\xi}, \boldsymbol{\eta} \in \mathfrak{X}(\mathcal{S}_p^{++})$, the Levi-Civita connection on $\mathcal{S}_p^{++}$ is

$$\nabla_{\boldsymbol{\xi}}^{\mathcal{S}_p^{++}} \boldsymbol{\eta} = \mathrm{D}\,\boldsymbol{\eta}[\boldsymbol{\xi}] - \mathrm{sym}\left(\boldsymbol{\eta}\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}\right). \tag{2.47}$$

The corresponding geodesic $\gamma^{\mathcal{S}_p^{++}}$ with initial conditions $\gamma^{\mathcal{S}_p^{++}}(0) = \boldsymbol{\Sigma}$ and $\dot{\gamma}^{\mathcal{S}_p^{++}}(0) = \boldsymbol{\xi}$ is

$$\gamma^{\mathcal{S}_p^{++}}(t) = \boldsymbol{\Sigma}^{\frac{1}{2}} \exp\left(t\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\xi}\boldsymbol{\Sigma}^{-\frac{1}{2}}\right)\boldsymbol{\Sigma}^{\frac{1}{2}} \tag{2.48}$$

where $\exp$ is the matrix exponential and for all $t \in \mathbb{R}$ and $\boldsymbol{A} \in \mathbb{R}^{p\times p}$, $\boldsymbol{A}^t = \exp(t\log(\boldsymbol{A}))$ when the matrix logarithm $\log(\boldsymbol{A})$ exists[2]. The geodesic $\gamma^{\mathcal{S}_p^{++}}$ with endpoints conditions $\gamma^{\mathcal{S}_p^{++}}(0) = \boldsymbol{\Sigma}_1$ and $\gamma^{\mathcal{S}_p^{++}}(1) = \boldsymbol{\Sigma}_2$ is

$$\gamma^{\mathcal{S}_p^{++}}(t) = \boldsymbol{\Sigma}_1^{\frac{1}{2}}\left(\boldsymbol{\Sigma}_1^{-\frac{1}{2}}\boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_1^{-\frac{1}{2}}\right)^t\boldsymbol{\Sigma}_1^{\frac{1}{2}} \tag{2.49}$$

Then, the exponential mapping on $\mathcal{S}_p^{++}$ at $\boldsymbol{\Sigma}$ is

$$\exp_{\boldsymbol{\Sigma}}^{\mathcal{S}_p^{++}}(\boldsymbol{\xi}) = \boldsymbol{\Sigma}^{\frac{1}{2}} \exp\left(\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\xi}\boldsymbol{\Sigma}^{-\frac{1}{2}}\right)\boldsymbol{\Sigma}^{\frac{1}{2}}. \tag{2.50}$$

The parallel transport between $\boldsymbol{\Sigma}_1 \in \mathcal{S}_p^{++}$ and $\boldsymbol{\Sigma}_2 \in \mathcal{S}_p^{++}$ moves vectors $\boldsymbol{\xi} \in T_{\boldsymbol{\Sigma}_1}\mathcal{S}_p^{++}$ onto the tangent space $T_{\boldsymbol{\Sigma}_2}\mathcal{S}_p^{++}$ while preserving the Riemannian metric and has the following formula [122]

$$\mathcal{T}_{\boldsymbol{\Sigma}_1,\boldsymbol{\Sigma}_2}^{\mathcal{S}_p^{++}}(\boldsymbol{\xi}) = \left(\boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_1^{-1}\right)^{\frac{1}{2}}\boldsymbol{\xi}\left(\left(\boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_1^{-1}\right)^{\frac{1}{2}}\right)^T. \tag{2.51}$$

[1] The notion of Fisher information metric is defined in Section 2.5.

[2] If $\boldsymbol{A} \in \mathcal{S}_p^{++}$ then the matrix logarithm $\log(\boldsymbol{A})$ exists and is unique.

Then, the logarithm mapping of $\boldsymbol{\Sigma}_2 \in \mathcal{S}_p^{++}$ at $\boldsymbol{\Sigma}_1 \in \mathcal{S}_p^{++}$ is

$$\log_{\boldsymbol{\Sigma}_1}^{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma}_2) = \boldsymbol{\Sigma}_1^{\frac{1}{2}} \log\left(\boldsymbol{\Sigma}_1^{-\frac{1}{2}} \boldsymbol{\Sigma}_2 \boldsymbol{\Sigma}_1^{-\frac{1}{2}}\right) \boldsymbol{\Sigma}_1^{\frac{1}{2}}. \tag{2.52}$$

Finally, the Riemannian distance on $\mathcal{S}_p^{++}$ is

$$d_{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2) = \left\| \log\left(\boldsymbol{\Sigma}_2^{-\frac{1}{2}} \boldsymbol{\Sigma}_1 \boldsymbol{\Sigma}_2^{-\frac{1}{2}}\right) \right\|_2. \tag{2.53}$$

Since, this distance is associated to the Riemannian metric (2.44), it is also invariant to affine transformations, *i.e.*

$$d_{\mathcal{S}_p^{++}}\left(\phi_{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma}_1), \phi_{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma}_2)\right) = d_{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2). \tag{2.54}$$

We now detail how to minimize a smooth function $h : \mathcal{S}_p^{++} \to \mathbb{R}$. Indeed, there are two tools left to minimize h:

1. the Riemannian gradient of h at any given point on $\mathcal{S}_p^{++}$,
2. a retraction defined on any tangent space $T_{\boldsymbol{\Sigma}}\mathcal{S}_p^{++}$.

The Riemannian gradient of h at $\boldsymbol{\Sigma}$ is given as a transformation of $\boldsymbol{G} \in \mathbb{R}^{p \times p}$, the Euclidean gradient of h at $\boldsymbol{\Sigma}$. Indeed, the Euclidean gradient is easily computed using automatic differentiation libraries such as Autograd [79] or JAX [25]. Thus, using this transformation, the Riemannian gradient can be automatically computed. The Riemannian gradient of h at $\boldsymbol{\Sigma}$ is

$$\operatorname{grad}_{\mathcal{S}_p^{++}} h(\boldsymbol{\Sigma}) = \boldsymbol{\Sigma} \operatorname{sym}(\boldsymbol{G}) \boldsymbol{\Sigma}. \tag{2.55}$$

It remains to provide a retraction $R_{\boldsymbol{\Sigma}}^{\mathcal{S}_p^{++}} : T_{\boldsymbol{\Sigma}}\mathcal{S}_p^{++} \to \mathcal{S}_p^{++}$. The exponential mapping (2.50) is of course a valid one. However, in practice we will use the following retraction for its numerical stability:

$$R_{\boldsymbol{\Sigma}}^{\mathcal{S}_p^{++}}(\boldsymbol{\xi}) = \boldsymbol{\Sigma} + \boldsymbol{\xi} + \frac{1}{2}\boldsymbol{\xi}\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}. \tag{2.56}$$

This retraction is a second order approximation of the exponential mapping (2.50):

$$\exp_{\boldsymbol{\Sigma}}^{\mathcal{S}_p^{++}}(t\boldsymbol{\xi}) = R_{\boldsymbol{\Sigma}}^{\mathcal{S}_p^{++}}(t\boldsymbol{\xi}) + \mathcal{O}(t^3). \tag{2.57}$$

It can also be seen as a second order retraction in the sense that $\nabla_{\dot{r}(t)}^{\mathcal{S}_p^{++}} \dot{r}(t)\Big|_{t=0} = \mathbf{0}$ with $r(t) = R_{\boldsymbol{\Sigma}}^{\mathcal{S}_p^{++}}(t\boldsymbol{\xi})$ and $\dot{r}(t) = \frac{d}{dt} r(t)$.

2.4.2 . $\mathcal{SS}_p^{++}$: manifold of symmetric positive definite matrices with unit determinant

An important and related manifold to $\mathcal{S}_p^{++}$ is $\mathcal{SS}_p^{++}$, the manifold of $p \times p$ symmetric positive definite matrices of unitary determinant. An example of application is the estimation of the scatter matrix of the compound Gaussian distribution [18]. This manifold is a Riemannian geodesic submanifold of $\mathcal{S}_p^{++}$: the geodesics of $\mathcal{SS}_p^{++}$ are geodesics of $\mathcal{S}_p^{++}$. Thus, knowing $\mathcal{S}_p^{++}$, the geometry of $\mathcal{SS}_p^{++}$ is easily derived.

We begin with the formal definition of $\mathcal{SS}_p^{++}$, the set of $p \times p$ symmetric positive definite matrices with a unit determinant,

$$\mathcal{SS}_p^{++} = \left\{ \boldsymbol{\Sigma} \in \mathcal{S}_p^{++} : |\boldsymbol{\Sigma}| = 1 \right\}. \tag{2.58}$$

By denoting $h(\boldsymbol{\Sigma}) = |\boldsymbol{\Sigma}| - 1$, we get that $\mathrm{D}\, h(\boldsymbol{\Sigma})[\boldsymbol{\xi}] = \mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi})$ for all $\boldsymbol{\xi} \in \mathcal{S}_p$. By taking $\boldsymbol{\xi} = \frac{\alpha}{p}\boldsymbol{\Sigma}$, we get that $\mathrm{D}\, h(\boldsymbol{\Sigma})[\boldsymbol{\xi}] = \alpha$ for all $\alpha \in \mathbb{R}$. Thus $\mathrm{rank}(\mathrm{D}\, h(\boldsymbol{\Sigma})) = 1$ for all $\boldsymbol{\Sigma} \in \mathcal{SS}_p^{++}$ and $\mathcal{SS}_p^{++}$ is a smooth embedded submanifold of $\mathcal{S}_p$. This induces that the tangent space at $\boldsymbol{\Sigma} \in \mathcal{SS}_p^{++}$ is

$$T_{\boldsymbol{\Sigma}}\mathcal{SS}_p^{++} = \left\{ \boldsymbol{\xi} \in \mathcal{S}_p : \mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}) = 0 \right\}. \tag{2.59}$$

Then, every tangent space $T_{\boldsymbol{\Sigma}}\mathcal{SS}_p^{++}$ is equipped with the affine invariant metric defined in (2.44), *i.e.* for all $\boldsymbol{\xi}, \boldsymbol{\eta} \in T_{\boldsymbol{\Sigma}}\mathcal{S}_p^{++}$

$$\langle \boldsymbol{\xi}, \boldsymbol{\eta} \rangle_{\boldsymbol{\Sigma}}^{\mathcal{SS}_p^{++}} = \langle \boldsymbol{\xi}, \boldsymbol{\eta} \rangle_{\boldsymbol{\Sigma}}^{\mathcal{S}_p^{++}} = \mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta} \right). \tag{2.60}$$

The orthogonal projection from $\mathbb{R}^{p \times p}$ onto $T_{\boldsymbol{\Sigma}}\mathcal{SS}_p^{++}$ is

$$P_{\boldsymbol{\Sigma}}^{\mathcal{SS}_p^{++}}(\boldsymbol{\xi}) = \mathrm{sym}(\boldsymbol{\xi}) - \frac{1}{p}\mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi})\boldsymbol{\Sigma} \tag{2.61}$$

where $\mathrm{sym}(\boldsymbol{\xi}) = \frac{1}{2}\left(\boldsymbol{\xi} + \boldsymbol{\xi}^T\right)$. For two smooth vector fields $\boldsymbol{\xi}, \boldsymbol{\eta} \in \mathfrak{X}(\mathcal{SS}_p^{++})$, the Levi-Civita connection on $\mathcal{S}_p^{++}$ is

$$\nabla_{\boldsymbol{\xi}}^{\mathcal{SS}_p^{++}} \boldsymbol{\eta} = P_{\boldsymbol{\Sigma}}^{\mathcal{SS}_p^{++}} \left(\nabla_{\boldsymbol{\xi}}^{\mathcal{S}_p^{++}} \boldsymbol{\eta} \right) = P_{\boldsymbol{\Sigma}}^{\mathcal{SS}_p^{++}} \left(\mathrm{D}\, \boldsymbol{\eta}[\boldsymbol{\xi}] - \mathrm{sym}\left(\boldsymbol{\eta}\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi} \right) \right). \tag{2.62}$$

Remarkably, the geodesic $\gamma^{\mathcal{S}_p^{++}}$ on $\mathcal{S}_p^{++}$ with initial conditions $\gamma^{\mathcal{S}_p^{++}}(0) = \boldsymbol{\Sigma} \in \mathcal{SS}_p^{++}$ and $\dot{\gamma}^{\mathcal{SS}_p^{++}}(0) = \boldsymbol{\xi} \in T_{\boldsymbol{\Sigma}}\mathcal{SS}_p^{++}$ has a unit determinant for all $t \in \mathbb{R}$. Indeed, we have

$$\left| \gamma^{\mathcal{SS}_p^{++}}(t) \right| = \left| \boldsymbol{\Sigma}^{\frac{1}{2}} \exp\left(t\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\xi}\boldsymbol{\Sigma}^{-\frac{1}{2}} \right) \boldsymbol{\Sigma}^{\frac{1}{2}} \right| = \exp\left(\mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi} \right) \right) = 1. \tag{2.63}$$

Thus, the corresponding geodesic $\gamma^{\mathcal{SS}_p^{++}}$ with initial conditions $\gamma^{\mathcal{SS}_p^{++}}(0) = \boldsymbol{\Sigma}$ and $\dot{\gamma}^{\mathcal{SS}_p^{++}}(0) = \boldsymbol{\xi}$ is

$$\gamma^{\mathcal{SS}_p^{++}}(t) = \boldsymbol{\Sigma}^{\frac{1}{2}} \exp\left(t\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\xi}\boldsymbol{\Sigma}^{-\frac{1}{2}} \right) \boldsymbol{\Sigma}^{\frac{1}{2}}. \tag{2.64}$$

The geodesic $\gamma^{\mathcal{SS}_p^{++}}$ with endpoints conditions $\gamma^{\mathcal{SS}_p^{++}}(0) = \boldsymbol{\Sigma}_1$ and $\gamma^{\mathcal{SS}_p^{++}}(1) = \boldsymbol{\Sigma}_2$ is

$$\gamma^{\mathcal{SS}_p^{++}}(t) = \boldsymbol{\Sigma}_1^{\frac{1}{2}} \left(\boldsymbol{\Sigma}_1^{-\frac{1}{2}} \boldsymbol{\Sigma}_2 \boldsymbol{\Sigma}_1^{-\frac{1}{2}} \right)^t \boldsymbol{\Sigma}_1^{\frac{1}{2}}. \tag{2.65}$$

Then, the exponential mapping on $\mathcal{SS}_p^{++}$ at $\boldsymbol{\Sigma}$ is

$$\exp_{\boldsymbol{\Sigma}}^{\mathcal{SS}_p^{++}}(\boldsymbol{\xi}) = \boldsymbol{\Sigma}^{\frac{1}{2}} \exp\left(\boldsymbol{\Sigma}^{-\frac{1}{2}} \boldsymbol{\xi} \boldsymbol{\Sigma}^{-\frac{1}{2}} \right) \boldsymbol{\Sigma}^{\frac{1}{2}}. \tag{2.66}$$

The parallel transport between $\boldsymbol{\Sigma}_1 \in \mathcal{SS}_p^{++}$ and $\boldsymbol{\Sigma}_2 \in \mathcal{SS}_p^{++}$ moves vectors $\boldsymbol{\xi} \in T_{\boldsymbol{\Sigma}_1}\mathcal{SS}_p^{++}$ onto the tangent space $T_{\boldsymbol{\Sigma}_2}\mathcal{SS}_p^{++}$ and has the following formula

$$\mathcal{T}_{\boldsymbol{\Sigma}_1,\boldsymbol{\Sigma}_2}^{\mathcal{SS}_p^{++}}(\boldsymbol{\xi}) = \left(\boldsymbol{\Sigma}_2 \boldsymbol{\Sigma}_1^{-1} \right)^{\frac{1}{2}} \boldsymbol{\xi} \left(\left(\boldsymbol{\Sigma}_2 \boldsymbol{\Sigma}_1^{-1} \right)^{\frac{1}{2}} \right)^T. \tag{2.67}$$

Then, the logarithm mapping of $\boldsymbol{\Sigma}_2 \in \mathcal{SS}_p^{++}$ at $\boldsymbol{\Sigma}_1 \in \mathcal{SS}_p^{++}$ is

$$\log_{\boldsymbol{\Sigma}_1}^{\mathcal{SS}_p^{++}}(\boldsymbol{\Sigma}_2) = \boldsymbol{\Sigma}_1^{\frac{1}{2}} \log\left(\boldsymbol{\Sigma}_1^{-\frac{1}{2}} \boldsymbol{\Sigma}_2 \boldsymbol{\Sigma}_1^{-\frac{1}{2}} \right) \boldsymbol{\Sigma}_1^{\frac{1}{2}}. \tag{2.68}$$

Finally, the Riemannian distance on $\mathcal{SS}_p^{++}$ is

$$d_{\mathcal{SS}_p^{++}}(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2) = \left\| \log\left(\boldsymbol{\Sigma}_2^{-\frac{1}{2}} \boldsymbol{\Sigma}_1 \boldsymbol{\Sigma}_2^{-\frac{1}{2}} \right) \right\|_2. \tag{2.69}$$

We now detail how to minimize a smooth function $h : \mathcal{SS}_p^{++} \to \mathbb{R}$. Indeed, there are two tools left to minimize h:

1. the Riemannian gradient of h at any given point on $\mathcal{SS}_p^{++}$,
2. a retraction defined on any tangent space $T_{\boldsymbol{\Sigma}}\mathcal{SS}_p^{++}$.

The Riemannian gradient of h at $\boldsymbol{\Sigma}$ is given as a transformation of $\boldsymbol{G} \in \mathbb{R}^{p \times p}$, the Euclidean gradient of h at $\boldsymbol{\Sigma}$,

$$\operatorname{grad}_{\mathcal{SS}_p^{++}} h(\boldsymbol{\Sigma}) = P_{\boldsymbol{\Sigma}}^{\mathcal{SS}_p^{++}}(\boldsymbol{\Sigma} \boldsymbol{G} \boldsymbol{\Sigma}). \tag{2.70}$$

It remains to provide a retraction $R_{\boldsymbol{\Sigma}}^{\mathcal{S}_p^{++}} : T_{\boldsymbol{\Sigma}}\mathcal{SS}_p^{++} \to \mathcal{SS}_p^{++}$. The exponential mapping (2.50) is of course a valid one. However, in practice we will use the following retraction for its numerical stability:

$$R_{\boldsymbol{\Sigma}}^{\mathcal{SS}_p^{++}}(\boldsymbol{\xi}) = \frac{\boldsymbol{\Sigma} + \boldsymbol{\xi} + \frac{1}{2}\boldsymbol{\xi}\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}}{\left| \boldsymbol{\Sigma} + \boldsymbol{\xi} + \frac{1}{2}\boldsymbol{\xi}\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi} \right|^{\frac{1}{p}}}. \tag{2.71}$$

It is a second order retraction in the sense that $\nabla_{\dot{r}(t)}^{\mathcal{SS}_p^{++}} \dot{r}(t)\Big|_{t=0} = \mathbf{0}$ with $r(t) = R_{\boldsymbol{\Sigma}}^{\mathcal{SS}_p^{++}}(t\boldsymbol{\xi})$ and $\dot{r}(t) = \frac{d}{dt}r(t)$. Indeed, by differentiating twice, we get that $\frac{d}{dt}r(0) = \boldsymbol{\xi}$ and $\frac{d^2}{dt^2}r(0) = \boldsymbol{\xi}\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}$. Thus, we have the desired property $\nabla_{\dot{r}(t)}^{\mathcal{SS}_p^{++}} \dot{r}(t)\Big|_{t=0} = 0$.

2.4.3 . $(\mathbb{R}_*^+)^n$: manifold of vectors with strictly positive entries

Another manifold of interest is the one of matrices with strictly positive entries. This manifold has recently gain some interest in applications with constraints of positivity such as robust covariance estimation [18] or optimal transport [93]. In the rest of the chapters, we will only handle constraints of positivity in vectors (and not in general rectangle matrices). Thus, this subsection presents the manifold of vectors with strictly positive entries.

We now present this manifold. First of all, the set of n-dimensional vectors with strictly positive entries is

$$(\mathbb{R}_*^+)^n = \{\boldsymbol{\tau} \in \mathbb{R}^n : \tau_i > 0\} \tag{2.72}$$

where τ_i is the i-th component of $\boldsymbol{\tau}$. $\mathbb{R}^n$ is a linear space and $(\mathbb{R}_*^+)^n$ is an open in $\mathbb{R}^n$. Thus and by definition, $(\mathbb{R}_*^+)^n$ is a smooth embedded submanifold of $\mathbb{R}^n$. This induces that the tangent space at $\boldsymbol{\tau} \in (\mathbb{R}_*^+)^n$ is

$$T_{\boldsymbol{\tau}}(\mathbb{R}_*^+)^n = \mathbb{R}^n. \tag{2.73}$$

Then, every tangent space $T_{\boldsymbol{\tau}}(\mathbb{R}_*^+)^n$ is equipped with the following Riemannian metric, for all $\boldsymbol{\xi}, \boldsymbol{\eta} \in T_{\boldsymbol{\tau}}(\mathbb{R}_*^+)^n$

$$\langle \boldsymbol{\xi}, \boldsymbol{\eta} \rangle_{\boldsymbol{\tau}}^{(\mathbb{R}_*^+)^n} = \left(\boldsymbol{\tau}^{\odot -1} \odot \boldsymbol{\xi}\right)^T \left(\boldsymbol{\tau}^{\odot -1} \odot \boldsymbol{\eta}\right) \tag{2.74}$$

where $.^{\odot -1}$ and $\odot$ are the elementwise inverse and multiplication respectively. With this Riemannian metric $(\mathbb{R}_*^+)^n$ becomes a Riemannian manifold. It is tightly linked to $\mathcal{S}_n^{++}$ described in Subsection 2.4.1. Indeed, there is a one-to-one correspondence between $(\mathbb{R}_*^+)^n$ and $\mathcal{D}_n^{++}$ (set of n-dimensional positive definite matrices) using the diffeomorphism $\operatorname{diag} : (\mathbb{R}_*^+)^n \to \mathcal{D}_n^{++}$ that puts elements of a vector onto the diagonal of the $n \times n$ zero matrix. $\mathcal{D}_n^{++}$ itself is a geodesically submanifold of $\mathcal{S}_n^{++}$. Thus, all the following formulas are counterparts of formulas from Subsection 2.4.1. This means that $\mathcal{D}_n^{++}$ is a submanifold of $\mathcal{S}_n^{++}$ and its geodesics are geodesics of $\mathcal{S}_n^{++}$. It is sometimes referred to the affine invariant Riemannian metric due to its invariance to affine transformations, *i.e.*

$$\langle \mathrm{D}\, \phi_{(\mathbb{R}_*^+)^n}(\boldsymbol{\tau})[\boldsymbol{\xi}], \mathrm{D}\, \phi_{(\mathbb{R}_*^+)^n}(\boldsymbol{\tau})[\boldsymbol{\eta}] \rangle_{\phi_{(\mathbb{R}_*^+)^n}(\boldsymbol{\tau})}^{(\mathbb{R}_*^+)^n} = \langle \boldsymbol{\xi}, \boldsymbol{\eta} \rangle_{\boldsymbol{\tau}}^{(\mathbb{R}_*^+)^n} \tag{2.75}$$

where $\phi_{(\mathbb{R}_*^+)^n}(\boldsymbol{\tau}) = \boldsymbol{a} \odot \boldsymbol{\tau}$, $\boldsymbol{a} \in \mathbb{R}^n$ with nonzero elements. The orthogonal projection from $\mathbb{R}^{p \times p}$ onto $T_{\boldsymbol{\tau}}(\mathbb{R}_*^+)^n$ is the identity mapping

$$P_{\boldsymbol{\tau}}^{(\mathbb{R}_*^+)^n}(\boldsymbol{\xi}) = \boldsymbol{\xi}. \tag{2.76}$$

For two smooth vector fields $\boldsymbol{\xi}, \boldsymbol{\eta} \in \mathfrak{X}((\mathbb{R}_*^+)^n)$, the Levi-Civita connection on $(\mathbb{R}_*^+)^n$ is

$$\nabla_{\boldsymbol{\xi}}^{(\mathbb{R}_*^+)^n} \boldsymbol{\eta} = \mathrm{D}\, \boldsymbol{\eta}[\boldsymbol{\xi}] - \boldsymbol{\eta} \odot \boldsymbol{\tau}^{\odot -1} \odot \boldsymbol{\xi}. \tag{2.77}$$

The corresponding geodesic $\gamma^{(\mathbb{R}_*^+)^n}$ with initial conditions $\gamma^{(\mathbb{R}_*^+)^n}(0) = \boldsymbol{\tau}$ and $\dot{\gamma}^{(\mathbb{R}_*^+)^n}(0) = \boldsymbol{\xi}$ is

$$\gamma^{(\mathbb{R}_*^+)^n}(t) = \boldsymbol{\tau} \odot \exp\left(t\boldsymbol{\tau}^{\odot -1} \odot \boldsymbol{\xi}\right). \tag{2.78}$$

The geodesic $\gamma^{(\mathbb{R}_*^+)^n}$ with endpoints conditions $\gamma^{(\mathbb{R}_*^+)^n}(0) = \boldsymbol{\tau}_1$ and $\gamma^{(\mathbb{R}_*^+)^n}(1) = \boldsymbol{\tau}_2$ is

$$\gamma^{(\mathbb{R}_*^+)^n}(t) = \boldsymbol{\tau}_1^{\odot(1-t)} \odot \boldsymbol{\tau}_2^{\odot t}. \tag{2.79}$$

Then, the exponential mapping on $(\mathbb{R}_*^+)^n$ at $\boldsymbol{\tau}$ is

$$\exp_{\boldsymbol{\tau}}^{(\mathbb{R}_*^+)^n}(\boldsymbol{\xi}) = \boldsymbol{\tau} \odot \exp\left(\boldsymbol{\tau}^{\odot -1} \odot \boldsymbol{\xi}\right). \tag{2.80}$$

The parallel transport between $\boldsymbol{\tau}_1 \in (\mathbb{R}_*^+)^n$ and $\boldsymbol{\tau}_2 \in (\mathbb{R}_*^+)^n$ moves vectors $\boldsymbol{\xi} \in T_{\boldsymbol{\tau}_1}(\mathbb{R}_*^+)^n$ onto the tangent space $T_{\boldsymbol{\tau}_2}(\mathbb{R}_*^+)^n$ and has the following formula

$$\mathcal{T}_{\boldsymbol{\tau}_1,\boldsymbol{\tau}_2}^{(\mathbb{R}_*^+)^n}(\boldsymbol{\xi}) = \boldsymbol{\tau}_2 \odot \boldsymbol{\tau}_1^{\odot -1} \odot \boldsymbol{\xi}. \tag{2.81}$$

Then, the logarithm mapping of $\boldsymbol{\tau}_2 \in (\mathbb{R}_*^+)^n$ at $\boldsymbol{\tau}_1 \in (\mathbb{R}_*^+)^n$ is

$$\log_{\boldsymbol{\tau}_1}^{(\mathbb{R}_*^+)^n}(\boldsymbol{\tau}_2) = \boldsymbol{\tau}_1 \odot \log\left(\boldsymbol{\tau}_1^{\odot -1} \odot \boldsymbol{\tau}_2\right). \tag{2.82}$$

Finally, the Riemannian distance on $(\mathbb{R}_*^+)^n$ is

$$d_{(\mathbb{R}_*^+)^n}(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2) = \left\|\log(\boldsymbol{\tau}_1) - \log(\boldsymbol{\tau}_2)\right\|_2. \tag{2.83}$$

Since, this distance is associated to the Riemannian metric (2.74), it is also invariant to affine transformations, *i.e.*

$$d_{(\mathbb{R}_*^+)^n}\left(\phi_{(\mathbb{R}_*^+)^n}(\boldsymbol{\tau}_1), \phi_{(\mathbb{R}_*^+)^n}(\boldsymbol{\tau}_2)\right) = d_{(\mathbb{R}_*^+)^n}(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2). \tag{2.84}$$

We now detail how to minimize a smooth function $h : (\mathbb{R}_*^+)^n \to \mathbb{R}$. Indeed, there are two tools left to minimize h:

1. the Riemannian gradient of h at any given point on $(\mathbb{R}_*^+)^n$,
2. a retraction defined on any tangent space $T_{\boldsymbol{\tau}}(\mathbb{R}_*^+)^n$.

The Riemannian gradient of h at $\boldsymbol{\tau}$ is given as a transformation of $\boldsymbol{g} \in \mathbb{R}^n$, the Euclidean gradient of h at $\boldsymbol{\tau}$:

$$\operatorname{grad}_{(\mathbb{R}_*^+)^n} h(\boldsymbol{\tau}) = \boldsymbol{\tau}^{\odot 2} \odot \boldsymbol{g}. \tag{2.85}$$

It remains to provide a retraction $R_{\boldsymbol{\tau}}^{(\mathbb{R}_*^+)^n} : T_{\boldsymbol{\tau}}(\mathbb{R}_*^+)^n \to (\mathbb{R}_*^+)^n$. The exponential mapping (2.80) is of course a valid one. However, in practice we will use the following retraction for its numerical stability:

$$R_{\boldsymbol{\tau}}^{(\mathbb{R}_*^+)^n}(\boldsymbol{\xi}) = \boldsymbol{\tau} + \boldsymbol{\xi} + \frac{1}{2}\boldsymbol{\tau}^{\odot -1} \odot \boldsymbol{\xi}^{\odot 2}. \tag{2.86}$$

This retraction is a second order approximation of the exponential mapping (2.80):

$$\exp_{\boldsymbol{\tau}}^{(\mathbb{R}_*^+)^n}(t\boldsymbol{\xi}) = R_{\boldsymbol{\tau}}^{(\mathbb{R}_*^+)^n}(t\boldsymbol{\xi}) + \mathcal{O}(t^3). \tag{2.87}$$

It can also be seen as a second order retraction in the sense that $\nabla_{\dot{r}(t)}^{(\mathbb{R}_*^+)^n} \dot{r}(t)\Big|_{t=0} = \mathbf{0}$ with $r(t) = R_{\boldsymbol{\tau}}^{(\mathbb{R}_*^+)^n}(t\boldsymbol{\xi})$ and $\dot{r}(t) = \frac{d}{dt}r(t)$.

2.4.4 . $\mathsf{Gr}_{p,k}$: manifold of subspaces

Many signal processing and machine learning algorithms rely on linear subspace estimation or classification. A standard subspace estimation algorithm is the Principal Component Analysis (PCA) [72]. This method computes an orthogonal basis of a linear subspace where most of the variance of the original data lies in. PCA is fast and easy to implement which makes it a very common algorithm and applied in numerous applications. Thus, an rich literature has developed since the original formulation of PCA. For example, the variance based cost function can be tweaked to enforce desired properties such as sparsity [153, 73] or robustness against outliers [88, 96]. This leads us to introduce the manifold of subspaces: the Grassmann manifold [2, 52, 1, 11]. Indeed, this Riemannian manifold enables the minimization of cost functions that rely on subspaces such the PCA-based ones. Furthermore, it describes the geometry of subspaces and thus, geodesics, distances and barycentres between subspaces can be computed. From a theoretical point of view, this has enabled the development of Intrinsic Cramér-Rao bounds (bounds on manifold) for the estimation of subspaces [121]. These bounds have shown in numerical experiments the efficiency of the PCA algorithm. Since the Grassmann manifold describes the geometry of subspaces, it is of broader interest than PCA. Indeed, many other applications rely on this manifold. We mention some of them: low rank completion for recommender systems [23, 22], dictionary learning [62] and video based face recognition [143, 69].

We now described the required tools of the Grassmann manifold for the next chapters. First of all, the Grassmann manifold is the set of k-dimensional linear subspaces of $\mathbb{R}^p$

$$\mathsf{Gr}_{p,k} = \left\{\mathrm{span}(\boldsymbol{U}) : \boldsymbol{U} \in \mathsf{St}_{p,k}\right\}. \tag{2.88}$$

Its elements can be represented by orthonormal basis. This leads us to introduce the set of orthonormal basis that spans k-dimensional subspaces in $\mathbb{R}^p$ called the Stiefel manifold and denoted $\mathsf{St}_{p,k}$. It is the zero level set of $h(\boldsymbol{U}) = \boldsymbol{U}^T\boldsymbol{U} - \boldsymbol{I}_k$ (smooth map from $\mathbb{R}^{p\times k}$ to $\mathbb{R}^{k\times k}$)

$$\mathsf{St}_{p,k} = \left\{\boldsymbol{U} \in \mathbb{R}^{p\times k} : \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_k\right\}. \tag{2.89}$$

It should be noted that for $k = p$, the Stiefel manifold amounts to the orthogonal group

$$\mathcal{O}_k = \left\{ \boldsymbol{U} \in \mathbb{R}^{k \times k} : \boldsymbol{U}^T \boldsymbol{U} = \boldsymbol{I}_k \right\}. \tag{2.90}$$

We briefly explain why $\mathsf{St}_{p,k}$ is a smooth manifold. For $\boldsymbol{\xi} \in \mathbb{R}^{p \times k}$, the linear map $\mathrm{D}\, h(\boldsymbol{U}) : \mathbb{R}^{p \times k} \to \mathcal{S}_k$ is

$$\mathrm{D}\, h(\boldsymbol{U})[\boldsymbol{\xi}] = \boldsymbol{U}^T \boldsymbol{\xi} + \boldsymbol{\xi}^T \boldsymbol{U}. \tag{2.91}$$

Thus, the kernel of $\mathrm{D}\, h(\boldsymbol{U})$ is

$$\ker(\mathrm{D}\, h(\boldsymbol{U})) = \left\{ \boldsymbol{\xi} \in \mathbb{R}^{p \times k} : \boldsymbol{U}^T \boldsymbol{\xi} + \boldsymbol{\xi}^T \boldsymbol{U} = \boldsymbol{0}_{k \times k} \right\}. \tag{2.92}$$

Then the linear map $\mathrm{D}\, h(\boldsymbol{U})$ is surjective. Indeed, for all $\boldsymbol{\eta} \in \mathcal{S}_k$, it suffices to take the direction $\boldsymbol{\xi} = \frac{1}{2}\boldsymbol{U}\boldsymbol{\eta}$ in order to get

$$\mathrm{D}\, h(\boldsymbol{U})[\boldsymbol{\xi}] = \boldsymbol{U}^T \left(\frac{1}{2} \boldsymbol{U} \boldsymbol{\eta} \right) + \left(\frac{1}{2} \boldsymbol{U} \boldsymbol{\eta} \right)^T \boldsymbol{U} = \boldsymbol{\eta}. \tag{2.93}$$

Thus, we get that $\mathrm{span}(\mathrm{D}\, h(\boldsymbol{U})) = \mathcal{S}_k$ which induces a constant and maximal rank: $\mathrm{rank}(\mathrm{D}\, h(\boldsymbol{U})) = \frac{k(k+1)}{2}$. Using the rank-nullity theorem, it follows that

$$\dim(\ker(\mathrm{D}(h(\boldsymbol{U})))) = pk - \frac{k(k+1)}{2}. \tag{2.94}$$

This shows that $\mathsf{St}_{p,k}$ is a smooth embedded submanifold in $\mathbb{R}^{p \times k}$ of dimension $pk - \frac{k(k+1)}{2}$ with the following tangent space at $\boldsymbol{U}$

$$T_{\boldsymbol{U}} \mathsf{St}_{p,k} = \ker(\mathrm{D}\, h(\boldsymbol{U})) = \left\{ \boldsymbol{\xi} \in \mathbb{R}^{p \times k} : \boldsymbol{U}^T \boldsymbol{\xi} + \boldsymbol{\xi}^T \boldsymbol{U} = \boldsymbol{0}_{k \times k} \right\}. \tag{2.95}$$

Another parametrization of $T_{\boldsymbol{U}} \mathsf{St}_{p,k}$ that is useful in the following is

$$T_{\boldsymbol{U}} \mathsf{St}_{p,k} = \left\{ \boldsymbol{U}\boldsymbol{A} + \boldsymbol{U}_\perp \boldsymbol{B} : \boldsymbol{A} \in \mathcal{A}_k, \boldsymbol{B} \in \mathbb{R}^{(p-k) \times k} \right\} \tag{2.96}$$

where $\mathcal{A}_k$ is the set $k \times k$ skew-symmetric matrices

$$\mathcal{A}_k = \left\{ \boldsymbol{X} \in \mathbb{R}^{k \times k} : \boldsymbol{X}^T = -\boldsymbol{X} \right\}. \tag{2.97}$$

and $\boldsymbol{U}_\perp \in \mathsf{St}_{p,p-k}$ is such that $\boldsymbol{U}^T \boldsymbol{U}_\perp = \boldsymbol{0}_{k \times k}$. It can be verified that the right part of (2.96) is indeed $T_{\boldsymbol{U}} \mathsf{St}_{p,k}$ by checking that its dimension is $pk - \frac{k(k+1)}{2}$ and that each of its elements $\boldsymbol{\xi}$ is such that $\mathrm{D}\, h(\boldsymbol{U})[\boldsymbol{\xi}] = \boldsymbol{0}_{k \times k}$. Then, $\mathsf{St}_{p,k}$ is turned into a Riemannian manifold by endowing it with the Euclidean metric on its tangent spaces, for all $\boldsymbol{\xi}, \boldsymbol{\eta} \in T_{\boldsymbol{U}} \mathsf{St}_{p,k}$

$$\langle \boldsymbol{\xi}, \boldsymbol{\eta} \rangle_{\boldsymbol{U}}^{\mathsf{St}_{p,k}} = \mathrm{Tr}\left(\boldsymbol{\xi}^T \boldsymbol{\eta} \right). \tag{2.98}$$

Since the Stiefel manifold is defined, we move on with the Grassmann manifold. As said previously, the Grassmann manifold $\mathsf{Gr}_{p,k}$ is the set of k-dimensional linear subspaces of $\mathbb{R}^p$ and its elements will be represented with elements of $\mathsf{St}_{p,k}$. Indeed, two orthogonal basis $\boldsymbol{U}, \boldsymbol{U}' \in \mathsf{St}_{p,k}$ represent the same subspace *i.e.* $\operatorname{span}(\boldsymbol{U}) = \operatorname{span}(\boldsymbol{U}')$ if and only if $\boldsymbol{UO} = \boldsymbol{U}'$ for some $\boldsymbol{O} \in \mathcal{O}_k$. This brings us to define the equivalence relation $\sim$ on $\mathsf{St}_{p,k}$

$$\boldsymbol{U} \sim \boldsymbol{U}' \iff \boldsymbol{UO} = \boldsymbol{U}' \text{ for some } \boldsymbol{O} \in \mathcal{O}_k. \tag{2.99}$$

Remarkably, it can be shown that there is a one-to-one correspondence between subspaces $\operatorname{span}(\boldsymbol{U})$ and the equivalence classes $\{\boldsymbol{UO} : \boldsymbol{O} \in \mathcal{O}_k\} \subset \mathsf{St}_{p,k}$. This leads us to give another definition of the Grassmann manifold, this time as a smooth quotient manifold of $\mathsf{St}_{p,k}$,

$$\mathsf{Gr}_{p,k} = \mathsf{St}_{p,k}/\mathcal{O}_k = \{\pi(\boldsymbol{U}) : \boldsymbol{U} \in \mathsf{St}_{p,k}\}, \tag{2.100}$$

where $\pi : \mathsf{St}_{p,k} \to \mathsf{Gr}_{p,k}$ is the map $\pi(\boldsymbol{U}) = \{\boldsymbol{UO} : \boldsymbol{O} \in \mathcal{O}_k\}$. We refer the reader to [19, Chapter 9] for a proof that $\mathsf{Gr}_{p,k}$ (2.100) is indeed a smooth quotient manifold. It should be noted that, using Definition (2.100), every element $\pi(\boldsymbol{U})$ of $\mathsf{Gr}_{p,k}$ can be represented by an arbitrary $\boldsymbol{U}' \in \mathsf{St}_{p,k}$ such that $\boldsymbol{U} \sim \boldsymbol{U}'$. Then, the dimension of $\mathsf{Gr}_{p,k}$ is

$$\dim(\mathsf{Gr}_{p,k}) = \dim(\mathsf{St}_{p,k}) - \dim(\mathcal{O}_k) = (p-k)k. \tag{2.101}$$

In order to represent elements of the tangent space of $\mathsf{Gr}_{p,k}$ at $\pi(\boldsymbol{U})$, the tangent space $T_{\boldsymbol{U}}\mathsf{St}_{p,k}$ is decomposed into two complementary subspaces, the vertical one $\mathcal{V}_{\boldsymbol{U}}$ and the horizontal one $\mathcal{H}_{\boldsymbol{U}}$

$$T_{\boldsymbol{U}}\mathsf{St}_{p,k} = \mathcal{V}_{\boldsymbol{U}} + \mathcal{H}_{\boldsymbol{U}}. \tag{2.102}$$

Using the definition of the map π and the tangent space of $\mathcal{O}_k$ at $\boldsymbol{I}_k$, the vertical space is

$$\mathcal{V}_{\boldsymbol{U}} = T_{\boldsymbol{U}}\pi^{-1}(\pi(\boldsymbol{U})) = \{\boldsymbol{UA} : \boldsymbol{A} \in \mathcal{A}_k\}. \tag{2.103}$$

From (2.96), every $\boldsymbol{\xi} \in T_{\boldsymbol{U}}\mathsf{St}_{p,k}$ can be parametrized as $\boldsymbol{\xi} = \boldsymbol{UA} + \boldsymbol{U}_\perp\boldsymbol{B}$ with $\boldsymbol{A} \in \mathcal{A}_k, \boldsymbol{B} \in \mathbb{R}^{(p-k)\times k}$. Let $\boldsymbol{\xi} \in \mathcal{V}_{\boldsymbol{U}}$, thus there exists $\boldsymbol{A} \in \mathcal{A}_k$ such that $\boldsymbol{\xi} = \boldsymbol{UA}$. Since for all $\boldsymbol{B} \in \mathbb{R}^{(p-k)\times k}$ we have $\langle \boldsymbol{UA}, \boldsymbol{U}_\perp\boldsymbol{B}\rangle_{\boldsymbol{U}}^{\mathsf{St}_{p,k}} = 0$, the horizontal space at $\boldsymbol{U}$ is

$$\mathcal{H}_{\boldsymbol{U}} = \left\{\boldsymbol{\xi} \in \mathbb{R}^{p\times k} : \boldsymbol{U}^T\boldsymbol{\xi} = \boldsymbol{0}_{k\times k}\right\}. \tag{2.104}$$

The associated orthogonal projection of $\boldsymbol{\xi} \in \mathbb{R}^{p\times k}$ onto $\mathcal{H}_{\boldsymbol{U}}$ is

$$P_{\boldsymbol{U}}^{\mathsf{Gr}_{p,k}}(\boldsymbol{\xi}) = (\boldsymbol{I} - \boldsymbol{UU}^T)\boldsymbol{\xi}. \tag{2.105}$$

As explained in Section 2.3, every element $\xi \in T_{\pi(\boldsymbol{U})}\mathsf{Gr}_{p,k}$ is represented by a unique element $\mathrm{lift}_{\boldsymbol{U}}(\xi) \in \mathcal{H}_{\boldsymbol{U}}$, called the horizontal lift, such that $\xi = \mathrm{D}\,\pi(\boldsymbol{U})[\mathrm{lift}_{\boldsymbol{U}}(\xi)]$. Remarkably on $\mathsf{Gr}_{p,k}$, there is an explicit relationship between horizontal lifts taken at different lifting points of a same tangent vector,

$$\mathrm{lift}_{\boldsymbol{UQ}}(\xi) = \mathrm{lift}_{\boldsymbol{U}}(\xi)\boldsymbol{Q} \text{ for all } \boldsymbol{Q} \in \mathcal{O}_k. \tag{2.106}$$

To prove this assertion, it suffices to take two different smooth curves on $\mathsf{St}_{p,k}$. The first one is such that $c(0) = \boldsymbol{U}$ and $c'(0) = \mathrm{lift}_{\boldsymbol{U}}(\xi)$ while the second one is $\tilde{c}(t) = c(t)\boldsymbol{Q}$. Thus, we get that $\tilde{c}(0) = \boldsymbol{UQ}$, $\tilde{c}'(0) = \mathrm{lift}_{\boldsymbol{U}}(\xi)\boldsymbol{Q}$ and $(\pi \circ c)(t) = (\pi \circ \tilde{c})(t)$ for all t where c and $\tilde{c}$ are defined. This induces that $(\pi \circ c)'(0) = (\pi \circ \tilde{c})'(0)$. By applying the chain rule, it follows that $\xi = \mathrm{D}\,\pi(\boldsymbol{UQ})[\mathrm{lift}_{\boldsymbol{U}}(\xi)\boldsymbol{Q}]$. Since $\mathrm{lift}_{\boldsymbol{U}}(\xi)\boldsymbol{Q} \in \mathcal{H}_{\boldsymbol{UQ}}$ and by uniqueness, $\mathrm{lift}_{\boldsymbol{U}}(\xi)\boldsymbol{Q}$ is the horizontal lift of ξ at $\boldsymbol{UQ}$ which proves (2.106). It remains to endow $\mathsf{Gr}_{p,k}$ with a Riemannian metric. A candidate is to leverage the Riemannian metric of $\mathsf{St}_{p,k}$, *i.e.*, for all $\xi, \eta \in T_{\pi(\boldsymbol{U})}\mathsf{Gr}_{p,k}$, $(\xi, \eta) \mapsto \langle \mathrm{lift}_{\boldsymbol{U}}(\xi), \mathrm{lift}_{\boldsymbol{U}}(\eta)\rangle_{\boldsymbol{U}}^{\mathsf{St}_{p,k}}$. To be a Riemannian metric it remains to prove that it is invariant to the lifting point, *i.e.*, for all $\boldsymbol{Q} \in \mathcal{O}_k$, we must have $\langle \mathrm{lift}_{\boldsymbol{UQ}}(\xi), \mathrm{lift}_{\boldsymbol{UQ}}(\eta)\rangle_{\boldsymbol{UQ}}^{\mathsf{St}_{p,k}} = \langle \mathrm{lift}_{\boldsymbol{U}}(\xi), \mathrm{lift}_{\boldsymbol{U}}(\eta)\rangle_{\boldsymbol{U}}^{\mathsf{St}_{p,k}}$. This is readily checked using Equation (2.106). Thus,

$$\langle \xi, \eta\rangle_{\pi(\boldsymbol{U})}^{\mathsf{Gr}_{p,k}} = \langle \mathrm{lift}_{\boldsymbol{U}}(\xi), \mathrm{lift}_{\boldsymbol{U}}(\eta)\rangle_{\boldsymbol{U}}^{\mathsf{St}_{p,k}} \tag{2.107}$$

is a Riemannian metric on $\mathsf{Gr}_{p,k}$. Hence, $\mathsf{Gr}_{p,k}$ becomes a Riemannian quotient manifold of $\mathsf{St}_{p,k}$. Then, classical tools of Riemannian geometry for the Grassmann manifold are given in the following. Given two smooth vector fields $\xi, \eta \in \mathfrak{X}(\mathsf{Gr}_{p,k})$, two associated smooth vector fields $\boldsymbol{\xi}, \boldsymbol{\eta} \in \mathfrak{X}(\mathsf{St}_{p,k})$ are derived using the horizontal lift, $\boldsymbol{\xi}(\boldsymbol{U}) = \mathrm{lift}_{\boldsymbol{U}}(\xi(\pi(\boldsymbol{U})))$ and $\boldsymbol{\eta}(\boldsymbol{U}) = \mathrm{lift}_{\boldsymbol{U}}(\eta(\pi(\boldsymbol{U})))$. Using these vector fields, the Levi-Civita on $\mathsf{Gr}_{p,k}$ is represented by its horizontal lift which is

$$\nabla_{\boldsymbol{\xi}}^{\mathsf{Gr}_{p,k}}\,\boldsymbol{\eta} = P^{\mathsf{Gr}_{p,k}}(\mathrm{D}\,\boldsymbol{\eta}[\boldsymbol{\xi}]) \tag{2.108}$$

where, for a given $\boldsymbol{\chi} \in \mathfrak{X}(\mathbb{R}^{p\times k})$, $P^{\mathsf{Gr}_{p,k}}(\boldsymbol{\chi}) : \boldsymbol{U} \in \mathsf{St}_{p,k} \mapsto P_{\boldsymbol{U}}^{\mathsf{Gr}_{p,k}}(\boldsymbol{\chi}(\boldsymbol{U})) \in \mathcal{H}_{\boldsymbol{U}}$. The corresponding geodesic with initial position $\pi(\boldsymbol{U})$ and initial speed ξ, of horizontal lift $\boldsymbol{\xi}$ at $\boldsymbol{U}$, is represented by

$$\gamma^{\mathsf{Gr}_{p,k}}(t) = \boldsymbol{UY}\cos(t\boldsymbol{\Sigma}) + \boldsymbol{X}\sin(t\boldsymbol{\Sigma}) \tag{2.109}$$

where $\boldsymbol{\xi} = \boldsymbol{X\Sigma Y}^T$ is the thin Singular Value Decomposition (SVD). Then, the exponential mapping on $\mathsf{Gr}_{p,k}$ of ξ at $\pi(\boldsymbol{U})$ is represented by

$$\exp_{\boldsymbol{U}}^{\mathsf{Gr}_{p,k}}(\boldsymbol{\xi}) = \boldsymbol{UY}\cos(\boldsymbol{\Sigma}) + \boldsymbol{X}\sin(\boldsymbol{\Sigma}) \tag{2.110}$$

where $\boldsymbol{\xi} = \boldsymbol{X}\boldsymbol{\Sigma}\boldsymbol{Y}^T$ is the thin SVD. The associated logarithmic mapping of $\pi(\boldsymbol{U}')$ at $\pi(\boldsymbol{U})$ is represented by its horizontal lift at $\boldsymbol{U}$,

$$\log_{\boldsymbol{U}}^{\mathsf{Gr}_{p,k}}(\boldsymbol{U}') = \boldsymbol{X}\boldsymbol{\Theta}\boldsymbol{Y}^T \tag{2.111}$$

where $\boldsymbol{X}\boldsymbol{\Theta}\boldsymbol{Y}^T$ is computed using the SVD of $(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^T)\boldsymbol{U}'(\boldsymbol{U}^T\boldsymbol{U}')^{-1} = \boldsymbol{X}\tan(\boldsymbol{\Theta})\boldsymbol{Y}^T$. The matrix $\boldsymbol{\Theta}$ contains the principal angles between $\mathrm{span}(\boldsymbol{U})$ and $\mathrm{span}(\boldsymbol{U}')$. It follows that the Riemannian distance between $\pi(\boldsymbol{U})$ and $\pi(\boldsymbol{U}')$ is

$$d_{\mathsf{Gr}_{p,k}}(\boldsymbol{U}, \boldsymbol{U}') = \|\boldsymbol{\Theta}\|_2 . \tag{2.112}$$

It should be noted that the principal angles between $\mathrm{span}(\boldsymbol{U})$ and $\mathrm{span}(\boldsymbol{U}')$ can also be computed using the SVD $\boldsymbol{U}^T\boldsymbol{U}' = \boldsymbol{O}_1\cos(\boldsymbol{\Theta})\boldsymbol{O}_2^T$.

We now detail how to minimize a smooth function $\bar{h} : \mathsf{St}_{p,k} \to \mathbb{R}$ that is invariant along equivalence classes, *i.e.*

$$\bar{h}(\boldsymbol{U}\boldsymbol{Q}) = \bar{h}(\boldsymbol{U}) \text{ for all } \boldsymbol{Q} \in \mathcal{O}_k. \tag{2.113}$$

Thanks to these invariances, a cost function $h : \mathsf{Gr}_{p,k} \to \mathbb{R}$ can be defined

$$h(\pi(\boldsymbol{U})) = \bar{h}(\boldsymbol{U}) \text{ for all } \pi(\boldsymbol{U}) \in \mathsf{Gr}_{p,k}. \tag{2.114}$$

It remains to define two tools to minimize h:

1. the Riemannian gradient of h at any given point on $\mathsf{Gr}_{p,k}$,
2. a retraction defined on any tangent space $T_{\pi(\boldsymbol{U})}\mathsf{Gr}_{p,k}$.

The Riemannian gradient of h at $\pi(\boldsymbol{U})$ is represented by its horizontal lift at $\boldsymbol{U}$ which is

$$\mathrm{grad}_{\mathsf{Gr}_{p,k}}\, \bar{h}(\boldsymbol{U}) = P_{\boldsymbol{U}}^{\mathsf{Gr}_{p,k}}(\boldsymbol{G}) \tag{2.115}$$

where $\boldsymbol{G} \in \mathbb{R}^{p\times k}$ is the Euclidean gradient of $\bar{h}$ at $\boldsymbol{U}$. It remains to provide a retraction. The exponential mapping (2.110) is of course a valid one. However, a more numerically stable retraction is represented by

$$\overline{R}_{\boldsymbol{U}}^{\mathsf{Gr}_{p,k}}(\boldsymbol{\xi}) = \boldsymbol{X}\boldsymbol{Y}^T \tag{2.116}$$

where $\boldsymbol{U} + \boldsymbol{\xi} = \boldsymbol{X}\boldsymbol{\Sigma}\boldsymbol{Y}^T$ is the thin SVD. Thus, given an iterate $\pi(\boldsymbol{U}^{(k)})$, an iterate of the Riemannian gradient descent is obtained with

$$\pi\left(\boldsymbol{U}^{(k+1)}\right) = \pi\left(\overline{R}_{\boldsymbol{U}^{(k)}}^{\mathsf{Gr}_{p,k}}(-\alpha\,\mathrm{grad}_{\mathsf{Gr}_{p,k}}\,\bar{h}(\boldsymbol{U}^{(k)}))\right) \tag{2.117}$$

where $\alpha > 0$ is a stepsize.

2.5 . Statistical estimation and intrinsic Cramér-Rao bounds

In Chapter 1, Section 1.3, the estimation theory is introduced. Here, we present the extension of this estimation theory on Euclidean sets to Riemannian manifolds. Given a *measurement* $\{\boldsymbol{x}_i\}_{i=1}^n$ in the *sample space* $\mathcal{X}$, we seek a *parameter* θ in the *parameter space* $\mathcal{M}$, a Riemannian manifold. To do so, an *estimate* $\hat{\theta}$ of θ is produced from the measurement $\{\boldsymbol{x}_i\}_{i=1}^n$ and the corresponding mapping from $\mathcal{X}$ to $\mathcal{M}$ is called an *estimator*.

Definition 31. *An estimator $\hat{\theta} : \mathcal{X} \to \mathcal{M}$ maps every measurement $\{\boldsymbol{x}_i\}_{i=1}^n$ to an estimate $\hat{\theta}(\{\boldsymbol{x}_i\}_{i=1}^n)$.*

In the following, some of the definitions and properties from Section 1.3 are extended to Riemannian manifolds. First of all, the *negative log-likelihood* is redefined as well as *maximum likelihood estimators*. Then *intrinsic Cramér-Rao bounds* (iCRBs) are introduced: they generalize CRBs to Riemannian manifolds. Indeed, they lower bound the variance of the estimator $\hat{\theta}$ which is measured with the Riemannian distance on $\mathcal{M}$ instead of the classical Euclidean MSE. These bounds have two interests:

- distances related with the statistical model can be used, resulting with simple and sometimes even parameters free iCRBs,
- iCRBS are intrinsic and thus take into account constraints of the estimation problem (such as orthogonality constraints) that are not easily handled with classical CRBs.

This section highlights the main results from the seminal paper [121] and mainly relies on the gentle introduction to iCRBs proposed in [21, Chapter 6]. The presented iCRBs only hold at high SNR and this for two reasons. First, the covariance matrix of the estimator uses the logarithmic map which is only defined locally. Thus, all estimates $\hat{\theta}$ must be in a neighborhood of the true parameter θ where the logarithmic map is defined. Second, the proof of the main presented result (Theorem 5) relies on Taylor expansions that are valid only when the curvature of the Riemannian manifold $\mathcal{M}$ is not too high.

2.5.1 . Some definitions for statistical estimation

First of all, we give some classical definitions from the estimation theory. Let a measurement $\{\boldsymbol{x}_i\}_{i=1}^n \in \mathcal{X}$ be a realization of a random variable X following a probability density function f parametrized by $\theta \in \mathcal{M}$, *i.e.*

$$X \sim f(.;\theta), \tag{2.118}$$

then, the negative log-likelihood function $\mathcal{L}$ is defined as minus the logarithm of f. In the following, we assume that $\mathcal{L}$ is at least twice differentiable on $\mathcal{M}$.

Definition 32. *Given $\{\boldsymbol{x}_i\}_{i=1}^n \in \mathcal{X}$, the negative log-likelihood function $\mathcal{L} : \mathcal{M} \to \mathbb{R}$ is defined by*

$$\mathcal{L}(\theta|\{\boldsymbol{x}_i\}_{i=1}^n) = -\log f(\{\boldsymbol{x}_i\}_{i=1}^n; \theta).$$

Then, the maximum likelihood estimator is defined as the minimizer of the negative log-likelihood on the Riemannian manifold $\mathcal{M}$.

Definition 33. *Given $\{\boldsymbol{x}_i\}_{i=1}^n \in \mathcal{X}$, the maximum likelihood estimator $\hat{\theta} \in \mathcal{M}$ is a minimizer of the negative log-likelihood function*

$$\hat{\theta} = \underset{\theta \in \mathcal{M}}{\arg\min}\ \mathcal{L}(\theta|\{\boldsymbol{x}_i\}_{i=1}^n).$$

Using the negative log-likelihood function, the *Fisher information metric* is defined.

Definition 34. *For a negative log-likelihood $\mathcal{L} : \mathcal{M} \to \mathbb{R}$, the Fisher information metric is defined for all $\xi, \eta \in T_\theta\mathcal{M}$ as*

$$\langle \xi, \eta \rangle_\theta^{\mathsf{FIM}} = \mathbb{E}[\mathrm{D}\,\mathcal{L}(\theta|\{\boldsymbol{x}_i\}_{i=1}^n)[\xi]\,\mathrm{D}\,\mathcal{L}(\theta|\{\boldsymbol{x}_i\}_{i=1}^n)[\eta]] = \mathbb{E}[\mathrm{D}^2\,\mathcal{L}(\theta|\{\boldsymbol{x}_i\}_{i=1}^n)[\xi, \eta]].$$

Then, an orthonormal basis of each tangent space $T_\theta\mathcal{M}$ is needed to derive components of tangent vectors. Let $q = \dim(\mathcal{M})$, an orthonormal basis of $T_\theta\mathcal{M}$ is denoted

$$e_\theta = \{e_\theta^1, \cdots, e_\theta^q\}. \tag{2.119}$$

The *score vector*[3] is a vector whose components are the directional derivatives of the negative log-likelihood with respect to each element of e_θ. This vector has a zero mean and its covariance matrix is called the *Fisher information matrix*.

Definition 35. *The score vector $\boldsymbol{s}_\theta \in \mathbb{R}^q$ is defined with respect to the orthonormal basis e_θ as*

$$(\boldsymbol{s}_\theta)_i = \mathrm{D}\,\mathcal{L}(\theta|\{\boldsymbol{x}_i\}_{i=1}^n)[e_\theta^i].$$

Lemma 1. *The score vector has a zero mean,* i.e. $\mathbb{E}[\boldsymbol{s}_\theta] = \boldsymbol{0}$.

Definition 36. *The Fisher information matrix $\boldsymbol{F}_\theta$ is the $q \times q$ symmetric, positive semidefinite matrix defined with respect to the basis e_θ as*

$$\boldsymbol{F}_\theta = \mathbb{E}[\boldsymbol{s}_\theta \boldsymbol{s}_\theta^T].$$

Thus, the entries of $\boldsymbol{F}_\theta$ are given by

$$(\boldsymbol{F}_\theta)_{ij} = \langle e_\theta^i, e_\theta^j \rangle_\theta^{\mathsf{FIM}}.$$

[3] Usually, the score vector is defined with the log-likelihood instead of the negative log-likelihood. However, since we only use its outer product, both definitions are equivalent.

Now that the Fisher information matrix is defined, we move on to the variance. As indicated in the introduction of this section, the variance is defined using the Riemannian distance. To do so, the *error vector* between θ and $\hat{\theta}$ is defined as

$$\xi_\theta = \log_\theta^{\mathcal{M}}(\hat{\theta}) \in T_\theta\mathcal{M}. \tag{2.120}$$

If $\mathcal{M}$ is a vector space, we recover the classical error vector $\xi_\theta = \hat{\theta} - \theta$. Using the coordinates of $T_\theta\mathcal{M}$, the error vector is defined as follows.

Definition 37. *The error vector between $\hat{\theta}$ and θ is denoted $\boldsymbol{\xi}_\theta \in \mathbb{R}^q$ and its coordinates, with respect to the basis e_θ of $T_\theta\mathcal{M}$, are*

$$(\boldsymbol{\xi}_\theta)_i = \left\langle \log_\theta^{\mathcal{M}}(\hat{\theta}), e_\theta^i \right\rangle_\theta^{\mathcal{M}}.$$

It should be noted that the norm of the error vector $\boldsymbol{\xi}_\theta$ is equal to the Riemannian distance on $\mathcal{M}$ between θ and $\hat{\theta}$. Indeed, we have

$$\begin{aligned}
\|\boldsymbol{\xi}_\theta\|^2 = \boldsymbol{\xi}_\theta^T\boldsymbol{\xi}_\theta &= \sum_i \left(\left\langle \log_\theta^{\mathcal{M}}(\hat{\theta}), e_\theta^i \right\rangle_\theta^{\mathcal{M}} \right)^2 \\
&= \left\langle \log_\theta^{\mathcal{M}}(\hat{\theta}), \sum_i \left\langle \log_\theta^{\mathcal{M}}(\hat{\theta}), e_\theta^i \right\rangle_\theta^{\mathcal{M}} e_\theta^i \right\rangle_\theta^{\mathcal{M}} \\
&= \left\| \log_\theta^{\mathcal{M}}(\hat{\theta}) \right\|_\theta^2 = d_{\mathcal{M}}^2(\theta, \hat{\theta}).
\end{aligned}$$

Then, the *bias vector* is defined: it is the mean of the error vector. If the bias vector is zero everywhere on $\mathcal{M}$, then $\hat{\theta}$ is called an *unbiased estimator*. In the following, it is assumed that $\hat{\theta}$ is unbiased. Finally, the covariance matrix of $\boldsymbol{\xi}_\theta$ is defined.

Definition 38. *The bias of an estimator $\hat{\theta} \in \mathcal{M}$ for a given parameter $\theta \in \mathcal{M}$ is the mean error vector*

$$\boldsymbol{b}_\theta = \mathbb{E}[\boldsymbol{\xi}_\theta].$$

An estimator is unbiased if its bias is zero everywhere

$$\boldsymbol{b}_\theta = \mathbf{0} \text{ for all } \theta \in \mathcal{M}.$$

Definition 39. *For an unbiased estimator $\hat{\theta}$, the covariance matrix $\boldsymbol{C}_\theta \in \mathbb{R}^{q\times q}$ with respect to the basis e_θ of $T_\theta\mathcal{M}$ is a symmetric, positive semidefinite matrix defined by*

$$\boldsymbol{C}_\theta = \mathbb{E}[\boldsymbol{\xi}_\theta\boldsymbol{\xi}_\theta^T].$$

Thus, the trace of $\boldsymbol{C}_\theta$ is the variance of the estimator $\hat{\theta}$

$$\mathrm{Tr}(\boldsymbol{C}_\theta) = \mathbb{E}\left[d_{\mathcal{M}}^2(\theta, \hat{\theta}) \right].$$

2.5.2 . Intrinsic Cramér-Rao bounds

With the tools defined previously, we are now able to present the main Theorem of intrinsic Cramér-Rao bounds for unbiased estimators.

Theorem 5. *Let $\mathcal{M}$ be a Riemannian manifold, let $\theta \in \mathcal{M}$ and let e_θ be an orthonormal basis of $T_\theta\mathcal{M}$. Consider an estimation problem on $\mathcal{M}$ such that the Fisher information matrix $\boldsymbol{F}_\theta$ is invertible. Then, for any unbiased estimator, the covariance matrix $\boldsymbol{C}_\theta$ obeys the following matrix inequality, where both $\boldsymbol{F}_\theta$ and $\boldsymbol{C}_\theta$ are expressed with respect to the basis e_θ*

$$\boldsymbol{C}_\theta \succeq \boldsymbol{F}_\theta^{-1} + \text{ curvature terms}$$

where $\succeq$ is the Loewner inequality.

In Theorem 5 the curvature terms are not specified for simplicity and they will be considered negligible in the following. From Theorem 5 and neglecting the curvature terms, we get the following iCRB

$$\mathbb{E}\left[d^2_{\mathcal{M}}(\theta, \hat{\theta})\right] \geq \operatorname{Tr}\left(\boldsymbol{F}_\theta^{-1}\right). \tag{2.121}$$

To illustrate the presented tools and Equation (2.121), we finish this section with an example on the iCRB of the covariance matrix estimation problem of the centered multivariate Gaussian distribution.

Example 16. *Let $\{\boldsymbol{x}_i\}_{i=1}^n \subset \mathbb{R}^p$, a set of independent and identically distributed realizations of a random variable $\boldsymbol{x}$ following a centered multivariate Gaussian distribution*

$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}) \tag{2.122}$$

where $\boldsymbol{\Sigma} \in \mathcal{S}_p^{++}$ is called the covariance matrix and parametrizes the distribution. The goal of this example is to derive a lower bound of the variance of any unbiased estimator $\hat{\boldsymbol{\Sigma}}$ of $\boldsymbol{\Sigma}$. It should be understood that this variance can be defined using any squared Riemannian distance between the true parameter and an unbiased estimator. In this example, we consider the Riemannian manifold of symmetric positive definite matrices presented in the subsection 2.4.1. Indeed, its Riemannian metric is proportional to the Fisher information metric associated with the model (2.122) *and thus the obtained iCRB is simple. To derive this iCRB, we begin by writing the negative log-likelihood function associated with the distribution* (2.122)*,*

$$\mathcal{L}(\boldsymbol{\Sigma}|\{\boldsymbol{x}_i\}_{i=1}^n) = \frac{n}{2}\left[\log|\boldsymbol{\Sigma}| + \operatorname{Tr}\left(\boldsymbol{\Sigma}^{-1}\hat{\boldsymbol{\Sigma}}_{SCM}\right)\right] + \text{constant}$$

where $\hat{\boldsymbol{\Sigma}}_{SCM}$ is the SCM

$$\hat{\boldsymbol{\Sigma}}_{SCM} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i\boldsymbol{x}_i^T. \tag{2.123}$$

Using the expectation $\mathbb{E}[\hat{\boldsymbol{\Sigma}}_{SCM}] = \boldsymbol{\Sigma}$, the Fisher information metric writes

$$\langle \boldsymbol{\xi}, \boldsymbol{\eta} \rangle_{\boldsymbol{\Sigma}}^{\mathsf{FIM}} = \mathbb{E}\left[\mathrm{D}^2 \, \mathcal{L}(\boldsymbol{\Sigma} | \{\boldsymbol{x}_i\}_{i=1}^n)[\boldsymbol{\xi}, \boldsymbol{\eta}] \right] = \frac{n}{2} \operatorname{Tr}(\boldsymbol{\Sigma}^{-1} \boldsymbol{\xi} \boldsymbol{\Sigma}^{-1} \boldsymbol{\eta}) = \frac{n}{2} \langle \boldsymbol{\xi}, \boldsymbol{\eta} \rangle_{\boldsymbol{\Sigma}}^{\mathcal{S}_p^{++}}$$

for all $\boldsymbol{\xi}, \boldsymbol{\eta} \in T_{\boldsymbol{\Sigma}} \mathcal{S}_p^{++} = \mathcal{S}_p$. Then, an orthonormal basis of the tangent space $T_{\boldsymbol{\Sigma}} \mathcal{S}_p^{++}$ with respect to the Riemannian metric $\langle ., . \rangle_{.}^{\mathcal{S}_p^{++}}$ is

$$\left\{ e_{\boldsymbol{\Sigma}}^i \right\}_{1 \leq i \leq p(p+1)/2} = \left\{ \boldsymbol{\Sigma}^{-\frac{1}{2}} \boldsymbol{E}^{ij} \boldsymbol{\Sigma}^{-\frac{1}{2}} \text{ for all } i, j \in [\![1, p]\!] \text{ such that } j \geq i \right\}$$

where $\boldsymbol{E}^{ij} \in \mathcal{S}_p$ and its ij^{th} and ji^{th} elements are $2^{-\frac{1}{2}}$ if $i \neq j$ and 1 otherwise. All other elements of $\boldsymbol{E}^{ij}$ are equal to 0. Using this orthonormal basis, we are able to compute the elements of the Fisher information matrix

$$(\boldsymbol{F}_{\boldsymbol{\Sigma}})_{ij} = \langle e_{\boldsymbol{\Sigma}}^i, e_{\boldsymbol{\Sigma}}^j \rangle_{\boldsymbol{\Sigma}}^{\mathsf{FIM}} = \begin{cases} \frac{n}{2} \text{ if } i = j \\ 0 \text{ otherwise.} \end{cases}$$

Hence, the Fisher information matrix is proportional to the identity

$$\boldsymbol{F}_{\boldsymbol{\Sigma}} = \frac{n}{2} \boldsymbol{I}_{p(p+1)/2}.$$

This leads to a simple closed form formula of the iCRB

$$\mathbb{E}\left[d^2_{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma}, \hat{\boldsymbol{\Sigma}}) \right] \geq \operatorname{Tr}\left(\boldsymbol{F}_{\boldsymbol{\Sigma}}^{-1} \right) = \frac{p(p+1)}{n}. \tag{2.124}$$

Remarkably, this iCRB is parameter free and is in $\mathcal{O}\left(\frac{p^2}{n}\right)$. Hence, the bound is quadratic with respect to the dimension of the data and is in one over the number of data. In comparaison, the classical Euclidean CRB for any unbiased estimator $\hat{\boldsymbol{\Sigma}}$ is (see [121] for a complete derivation)

$$\mathbb{E}\left[\left\| \boldsymbol{\Sigma} - \hat{\boldsymbol{\Sigma}} \right\|_2^2 \right] \geq \frac{2 \left(\sum_{i \leq j} \boldsymbol{\Sigma}_{ij}^2 + \sum_{i<j} \boldsymbol{\Sigma}_{ii} \boldsymbol{\Sigma}_{jj} \right)}{n}. \tag{2.125}$$

Finally, in [121], it is shown that $\hat{\boldsymbol{\Sigma}}_{SCM}$ is asymptotically unbiased, i.e. *$\boldsymbol{b}_{\boldsymbol{\Sigma}} \to \mathbf{0}$ as $n \to \infty$ for all $\boldsymbol{\Sigma} \in \mathcal{S}_p^{++}$ with $\boldsymbol{b}_{\boldsymbol{\Sigma}}$ defined in Definition 38.* (2.124) *is illustrated in the Figure 2.7 in which we observe that the MSE of $\hat{\boldsymbol{\Sigma}}_{SCM}$ reaches the iCRB for n large. We also observe that for n small, there is a discrepancy between the MSE of $\hat{\boldsymbol{\Sigma}}_{SCM}$ and the iCRB. [121] shows that this discrepancy is due to the bias and the inefficiency of $\hat{\boldsymbol{\Sigma}}_{SCM}$ on $\mathcal{S}_p^{++}$. This contradicts the classical analysis on the Euclidean space derived from the Figure 2.8. Thus, studying the estimation error with an intrinsic point of view can also lead to much different results than its Euclidean counterpart.*

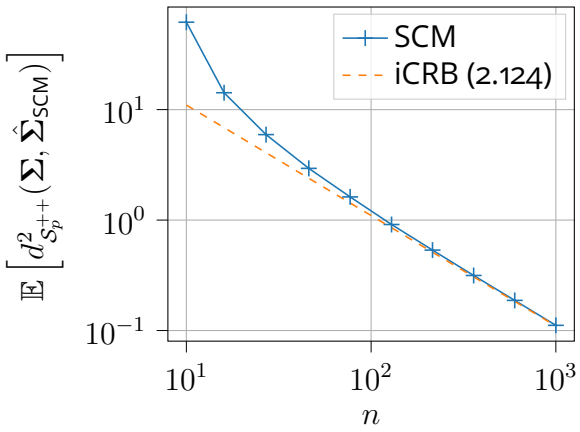


Figure 2.7: Mean Squared Error (MSE) computed as $\mathbb{E}\left[d^2_{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma}, \hat{\boldsymbol{\Sigma}}_{\text{SCM}})\right]$ with 1000 Monte-Carlo versus n, the number of samples of dimension $p = 10$ to estimate $\hat{\boldsymbol{\Sigma}}_{\text{SCM}}$.

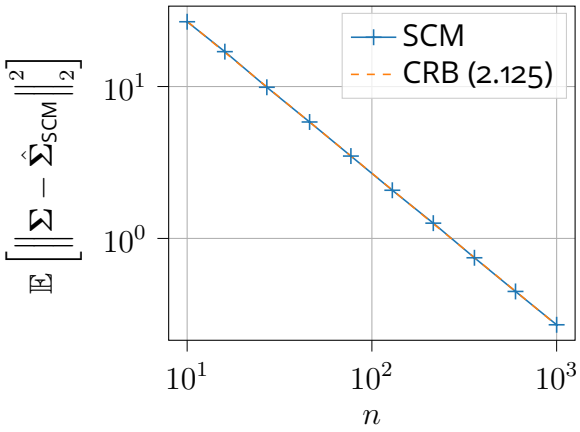


Figure 2.8: Mean Squared Error (MSE) computed as $\mathbb{E}\left[\left\|\boldsymbol{\Sigma} - \hat{\boldsymbol{\Sigma}}_{\text{SCM}}\right\|_2^2\right]$ with 1000 Monte-Carlo versus n, the number of samples of dimension $p = 10$ to estimate $\hat{\boldsymbol{\Sigma}}_{\text{SCM}}$.

2.6 . Conclusions

We began this chapter by defining Riemannian manifolds as embedded submanifolds of linear spaces with metrics that vary smoothly between tangent spaces. This Definition allowed us to simply define important Riemannian manifolds such as the sphere. Then, we introduced some tools of Riemannian manifolds: orthogonal projection, Levi-Civita connection, geodesic, exponential map, logarithmic map, geodesic distance and parallel transport. Taylor expansions on curves as well as first-order optimization algorithms

on Riemannian manifolds have been presented. Also, Assumptions on costs functions have been introduced in order to guarantee convergence to critical points. Then, we presented the usefulness of Riemannian quotient manifolds as well as their properties for signal processing and machine learning problems such as subspace estimation. Next, we detailed some Riemannian manifolds that are used in the subsequent chapters. Finally, we presented intrinsic Cramér-Rao bounds to compute the minimum variance of an unbiased statistical estimator on a given Riemannian manifold.

3 - Robust estimation and classification of non centered data

Classically, many signal processing applications or machine learning algorithms make use of the second order statistic. Indeed, a standard distribution is the multivariate centered Gaussian distribution. The latter is fully parametrized by its covariance matrix which turns out to be an interesting feature to discriminate data in machine learning problems. Recently, the Riemannian geometry $\mathcal{S}_p^{++}$ associated with the Fisher information metric (FIM) of the centered Gaussian distribution [120] has been used with great successes on classification problems, *e.g.* on EEG data [8], in detection of pedestrians [134] or in Diffusion tensor imaging [113]. These successes are described in Chapter 1 Section 1.5 and the geometry of $\mathcal{S}_p^{++}$ is presented in Chapter 2 Section 2.4.1. We recall some of its elements here since they are important for this chapter. The distance of the Riemannian manifold $\mathcal{S}_p^{++}$ between two covariance matrices $\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2 \in \mathcal{S}_p^{++}$ benefits from a simple closed form formula,

$$d_{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2) = \left\| \log\left(\boldsymbol{\Sigma}_1^{-\frac{1}{2}} \boldsymbol{\Sigma}_2 \boldsymbol{\Sigma}_1^{-\frac{1}{2}} \right) \right\|_2 . \tag{3.1}$$

Notably, this distance is affine invariant, *i.e* $\forall \boldsymbol{A} \in \mathrm{GL}_p$,

$$d_{\mathcal{S}_p^{++}}(\boldsymbol{A}\boldsymbol{\Sigma}_1\boldsymbol{A}^T, \boldsymbol{A}\boldsymbol{\Sigma}_2\boldsymbol{A}^T) = d_{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2). \tag{3.2}$$

This invariance property is of particular interest for applications based on mixing models [119, 34], *i.e* the measured signal is assumed to be a linear combination of non-measurable and discriminative source signals. In this case, the distances in the source space are equal to those in the measured signal space. Then, many classification-clustering algorithms, *e.g.* the *Nearest centroid classifier* or *K-means++*, need to compute centers of mass. The Riemannian center of mass of $\{\boldsymbol{\Sigma}_i\}_{i=1}^M$, denoted $\boldsymbol{\Sigma}$, associated with the distance (3.1), is defined as the minimizer of the variance [75, 94],

$$\boldsymbol{\Sigma} = \underset{\boldsymbol{\Sigma} \in \mathcal{S}_p^{++}}{\arg\min} \frac{1}{M} \sum_{i=1}^{M} d^2_{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma}, \boldsymbol{\Sigma}_i). \tag{3.3}$$

A gradient descent achieves this minimization, see Chapter 2 Section 2.2.

As mentioned earlier, this geometry assumes that the signal is centered. Indeed, it is the information geometry [3, 121, 120] of the centered Gaussian distribution. Hence, Equation (3.1) is the distance between centered Gaussian distributions. It does not use the mean/location whereas it can

be a discriminative feature, *e.g.* in multispectral imaging where signals are non-centered [83]. Performance could be improved if data were modeled with non-centered distributions and classified with associated Riemannian manifolds (or statistical manifolds if the FIM is considered). Furthermore, some applications such as SAR images segmentation or time series classification can benefit from other statistical models such as robust statistical models [104, 136]. Thus, the objective of this chapter is to extend the pipeline, presented in Chapter 1 Section 1.2, to other statistical models than the centered Gaussian one. In particular, we propose to use the location, in addition to the covariance matrix, as a clustering-classification feature. This has two practical consequences. The first one is the development of joint location-covariance statistical estimators. The second one is the realization of machine learning algorithms that handle jointly these two statistics.

The chapter is organized as follows. First of all, Sections 3.1, 3.2 and 3.3 present a clustering-classification pipeline for non-centered Gaussian data. The information geometry $\mathcal{M}_p$ of multivariate Gaussian distributions is leveraged to derive affine invariant divergences between couples of locations and covariance matrices. A Riemannian gradient descent is proposed to optimize functions of Gaussian distributions. In this chapter, it is used to compute centers of mass associated with the proposed divergences. The proposed pipeline is applied on the *Breizhcrops* dataset and robustness to transformations of data are presented. The rest of the chapter proposes to model data with the non-centered mixtures of scaled Gaussian distributions (NC-MSG). Section 3.4 presents the model and its parameter space $\mathcal{M}_{p,n}$. Then, Sections 3.5 and 3.6 establish two Riemannian geometries for $\mathcal{M}_{p,n}$. These geometries are developed to optimize functions of NC-MSGs such the negative log-likelihood (NLL) and variances to compute centers of mass. The first geometry uses a product metric and thus is simple to derive. However, this geometry gives optimization algorithms that are slow in practice. Hence, we derive a second Riemannian geometry that uses the FIM of the NC-MSG. This geometry is only known locally, *i.e.* geodesics and distances between arbitrary points remain unknown. Since, geodesics are unknown, we propose to classify NC-MSGs with a Kullback-Leibler (KL) divergence. The associated center of mass is derived. Finally, the proposed algorithms are extensively studied though simulations and applied on real data with the *Breizhcrops* dataset. Robustness to transformation of the data are presented.

3.1 . Non -centered multivariate Gaussian distribution

3.1.1 . Parameter space $\mathcal{M}_p$ and information geometry

Let a set of n data points $\boldsymbol{x}_i \in \mathbb{R}^p$ sampled from a random variable $\boldsymbol{x}$ following a Gaussian distribution

$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{3.4}$$

The parameters $\boldsymbol{\mu} \in \mathbb{R}^p$ and $\boldsymbol{\Sigma} \in \mathcal{S}_p^{++}$ are the location and covariance matrix respectively. The negative log-likelihood is defined on the set $\mathcal{M}_p = \mathbb{R}^p \times \mathcal{S}_p^{++}$ and given $\upsilon = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ writes (neglecting terms that do not depend on υ)

$$\mathcal{L}_{\mathsf{G}}(\upsilon) = \log|\boldsymbol{\Sigma}| + \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}). \tag{3.5}$$

The maximum likelihood estimators of the Gaussian distribution are the well known sample mean and SCM,

$$\begin{aligned}\hat{\boldsymbol{\mu}}_{\mathsf{SM}} &= \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i,\\ \hat{\boldsymbol{\Sigma}}_{\mathsf{SCM}} &= \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{\mathsf{SM}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{\mathsf{SM}})^T.\end{aligned} \tag{3.6}$$

Then, $\mathcal{M}_p$ is turned into a Riemannian manifold. The tangent space $T_\upsilon\mathcal{M}_p$ of $\mathcal{M}_p$ at υ is identified to the product space $\mathbb{R}^p \times \mathcal{S}_p$ with $\mathcal{S}_p$ the set of symmetric matrices. Moreover, $\mathcal{M}_p$ is equipped with the FIM associated with the negative log-likelihood (3.5). Let $\xi = \left(\boldsymbol{\xi}_{\boldsymbol{\mu}}, \boldsymbol{\xi}_{\boldsymbol{\Sigma}}\right)$, $\eta = \left(\boldsymbol{\eta}_{\boldsymbol{\mu}}, \boldsymbol{\eta}_{\boldsymbol{\Sigma}}\right) \in T_\upsilon\mathcal{M}_p$, this metric writes [120]

$$\langle \xi, \eta \rangle_\upsilon^{\mathcal{M}_p} = \boldsymbol{\xi}_{\boldsymbol{\mu}}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_{\boldsymbol{\mu}} + \frac{1}{2}\operatorname{Tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_{\boldsymbol{\Sigma}}). \tag{3.7}$$

Remarkably, the FIM (3.7) is invariant under affine transformations. Given $\boldsymbol{A} \in \mathsf{GL}_p$ and $\boldsymbol{\mu}_0 \in \mathbb{R}^p$ we verify that

$$\langle \mathrm{D}\,\phi_{\mathcal{M}_p}(\upsilon)[\xi], \mathrm{D}\,\phi_{\mathcal{M}_p}(\upsilon)[\eta] \rangle_{\phi_{\mathcal{M}_p}(\upsilon)}^{\mathcal{M}_p} = \langle \xi, \eta \rangle_\upsilon^{\mathcal{M}_p}, \tag{3.8}$$

where the affine transformation writes,

$$\phi_{\mathcal{M}_p}(\upsilon) = (\boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{\mu}_0, \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^T). \tag{3.9}$$

A geodesic $\gamma(t) = (\boldsymbol{\mu}(t), \boldsymbol{\Sigma}(t)) : \mathbb{R} \to \mathcal{M}_p$ associated with the FIM (3.7) must have a zero acceleration [32]

$$\begin{cases}\ddot{\boldsymbol{\mu}}(t) - \dot{\boldsymbol{\Sigma}}(t)\boldsymbol{\Sigma}(t)^{-1}\dot{\boldsymbol{\mu}}(t) = \boldsymbol{0}\\ \ddot{\boldsymbol{\Sigma}}(t) + \dot{\boldsymbol{\mu}}(t)\dot{\boldsymbol{\mu}}(t)^T - \dot{\boldsymbol{\Sigma}}(t)\boldsymbol{\Sigma}(t)^{-1}\dot{\boldsymbol{\Sigma}}(t) = \boldsymbol{0}.\end{cases} \tag{3.10}$$

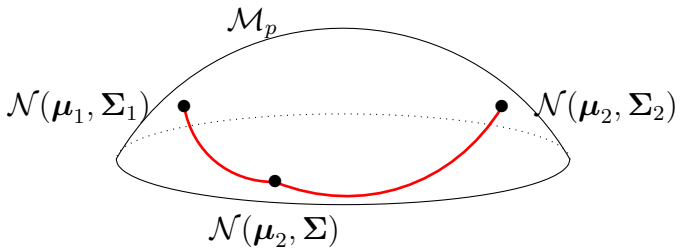


Figure 3.1: Illustration of a geodesic triangle on the Riemannian manifold $\mathcal{M}_p$. If the covariance matrix $\boldsymbol{\Sigma}$ is well chosen, then the geodesic triangle $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \rightarrow \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}) \rightarrow \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ is known with closed formed formulas. The arc length of the path in red is $\delta_{\mathcal{M}_p}$.

An explicit expression of the geodesic on $\mathcal{M}_p$ with initial position $\gamma(0) = \upsilon$ and initial velocity $\dot{\gamma}(0) = \xi$ is derived in [32],

$$\gamma(t) = (\boldsymbol{\mu}(t), \boldsymbol{\Sigma}(t)) = \Big(2\boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{R}(t)\sinh\left(\frac{t}{2}\boldsymbol{G}\right)\boldsymbol{G}^{-}\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\xi}_{\boldsymbol{\mu}} + \boldsymbol{\mu}, \boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{R}(t)\boldsymbol{R}(t)^T\boldsymbol{\Sigma}^{\frac{1}{2}}\Big) \quad (3.11)$$

where

$$\boldsymbol{G}^2 = \left(\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-\frac{1}{2}}\right)^2 + 2\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\xi}_{\boldsymbol{\mu}}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T\boldsymbol{\Sigma}^{-\frac{1}{2}},$$

$$\boldsymbol{R}(t) = \left(\cosh\left(\frac{t}{2}\boldsymbol{G}\right) - \boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{G}^{-}\sinh\left(\frac{t}{2}\boldsymbol{G}\right)\right)^{-T},$$

and $\boldsymbol{G}^{-}$ is the Moore–Penrose inverse of $\boldsymbol{G}$. However (3.11) only gives an expression of a geodesic with initial position and velocity. Unfortunately, in the general case, a closed form expression of a geodesic between two points $\upsilon_1 = (\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\upsilon_2 = (\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ remains unknown. Hence, the distance between υ_1 and υ_2 associated with the FIM (3.7) is also unknown. Using other metrics than the FIM could give closed form distances but they would not necessarily have the affine transformation invariance property. Instead, we propose to use geodesic triangles derived from (3.11).

3.1.2 . Geodesic triangles and divergences

Geodesic triangles between υ_1 and υ_2 using the expression (3.11) can be derived. Indeed, by carefully choosing intermediate points υ, geodesics are obtained between υ_1 and υ and then between υ and υ_2. Hence, we get geodesic triangles $\upsilon_1 \rightarrow \upsilon \rightarrow \upsilon_2$. The squared arc-length of one of these geodesic triangles is then measured to get a divergence denoted $\delta_{\mathcal{M}_p}$. By construction, these divergences $\delta_{\mathcal{M}_p}$ are invariant by affine transformation,

$$\delta_{\mathcal{M}_p}(\phi_{\mathcal{M}_p}(\upsilon_1), \phi_{\mathcal{M}_p}(\upsilon_2)) = \delta_{\mathcal{M}_p}(\upsilon_1, \upsilon_2). \quad (3.12)$$

To construct those triangles, we recall that the manifold with a fixed location vector $\mathcal{M}_{p_{\boldsymbol{\mu}}} = \{(\boldsymbol{\mu}, \boldsymbol{\Sigma}) : \boldsymbol{\Sigma} \in \mathcal{S}_p^{++}\}$ endowed with metric (3.7) is a geodesic submanifold of $\mathcal{M}_p$, *i.e.* the geodesics of $\mathcal{M}_{p_{\boldsymbol{\mu}}}$ are geodesics of $\mathcal{M}_p$. Hence, in the case $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2$, the squared distance on $\mathcal{M}_p$ is

$$d^2_{\mathcal{M}_p}(\upsilon_1, \upsilon_2) = \frac{1}{2} d^2_{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2) \tag{3.13}$$

Thus, to create a triangle between υ_1 and υ_2, it suffices to find an intermediate point $\upsilon = (\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is determined such that a geodesic (3.11) is known between υ_1 and υ. These geodesic triangles are represented in Figure 3.1. Based on this scheme, [32] proposed to use a rescaling of the initial covariance matrix as an intermediate point, *i.e.*

$$\upsilon_c = (\boldsymbol{\mu}_2, c\boldsymbol{\Sigma}_1), \tag{3.14}$$

with $c = \left|\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_2\right|^{\frac{1}{p}} = \underset{c \in \mathbb{R}_*^+}{\arg\min}\ d^2_{\mathcal{M}_p}(\upsilon_c, \upsilon_2)$. Using this point, a first invariant under affine transformations (3.9) divergence on $\mathcal{M}_p$ is proposed in Corollary 1.

Corollary 1 (Divergence $\delta_{c,\mathcal{M}_p}$)**.** *An invariant under affine transformations* (3.9) *divergence on* $\mathcal{M}_p$ *is*

$$\begin{aligned}\delta_{c,\mathcal{M}_p}(\upsilon_1, \upsilon_2) = 2\,\mathrm{acosh}\left(\frac{c^{-\frac{1}{2}}}{2}\left(c + 1 + \frac{1}{2}\Delta\boldsymbol{\mu}^T\boldsymbol{\Sigma}_1^{-1}\Delta\boldsymbol{\mu}\right)\right)^2 \\ + \frac{(p-1)}{2}\log(c)^2 + \frac{1}{2}\left\|\log\left(c\boldsymbol{\Sigma}_2^{-\frac{1}{2}}\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_2^{-\frac{1}{2}}\right)\right\|_2^2.\end{aligned}$$

where $\Delta\boldsymbol{\mu} = \boldsymbol{\mu}_2 - \boldsymbol{\mu}_1$ *and* $c = \left|\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_2\right|^{\frac{1}{p}}$.

Proof. Using the intermediate point $\upsilon_c = (\boldsymbol{\mu}_2, c\boldsymbol{\Sigma}_1)$, and applying the construction of triangles explained earlier, we get

$$\delta_{c,\mathcal{M}_p}(\upsilon_1, \upsilon_2) = \rho^2(\upsilon_1, \upsilon_c) + d^2_{\mathcal{M}_p}(\upsilon_c, \upsilon_2), \tag{3.15}$$

where ρ is the arc length of a geodesic (3.11) computed in Equation (18) of [32]. Then, ρ is simplified. By denoting $\widetilde{\boldsymbol{\mu}} = \boldsymbol{\Sigma}_1^{-\frac{1}{2}}\Delta\boldsymbol{\mu}$, we get

$$\frac{1}{2}\rho^2(\upsilon_1, \upsilon_c) = \left\|\mathrm{acosh}\left(\frac{c^{-\frac{1}{2}}}{2}(\boldsymbol{I}_p + c\boldsymbol{I}_p + \frac{1}{2}\widetilde{\boldsymbol{\mu}}\widetilde{\boldsymbol{\mu}}^T)\right)\right\|_2^2 \tag{3.16}$$

$$= \mathrm{acosh}\left(\frac{c^{-\frac{1}{2}}}{2}(c + 1 + \frac{1}{2}\widetilde{\boldsymbol{\mu}}^T\widetilde{\boldsymbol{\mu}})\right)^2 \tag{3.17}$$

$$+ (p-1)\,\mathrm{acosh}\left(\frac{c^{-\frac{1}{2}} + c^{\frac{1}{2}}}{2}\right)^2 \tag{3.18}$$

Using $\operatorname{acosh}\left(\frac{c^{-\frac{1}{2}}+c^{\frac{1}{2}}}{2}\right)^2 = \log(c^{\frac{1}{2}})^2 = \frac{1}{4}\log(c)^2$ and Equation (3.13), we get the divergence $\delta_{c,\mathcal{M}_p}$. □

In [129], the authors proved that the orthogonal projection of υ_1 onto $\mathcal{N}^p_{\boldsymbol{\mu}_2}$ is

$$\upsilon_\perp = \left(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1 + \frac{1}{2}\Delta\boldsymbol{\mu}\Delta\boldsymbol{\mu}^T\right). \tag{3.19}$$

The squared arc length of the geodesic between υ_1 and $\upsilon_\perp$ is also computed in [129],

$$\delta_\perp(\upsilon_1, \upsilon_\perp) = \frac{1}{2}\operatorname{acosh}\left(1 + \Delta\boldsymbol{\mu}^T\boldsymbol{\Sigma}_1^{-1}\Delta\boldsymbol{\mu}\right)^2. \tag{3.20}$$

Hence, using the intermediate point $\upsilon_\perp$ and summing Equation (3.20) with Equation (3.13) we get a second invariant under affine transformations (3.9) divergence on $\mathcal{M}_p$. This divergence is proposed in Corollary 2.

Corollary 2 (Divergence $\delta_{\perp,\mathcal{M}_p}$)**.** *An invariant under affine transformations* (3.9)*, divergence on* $\mathcal{M}_p$ *is*

$$\delta_{\perp,\mathcal{M}_p}(\upsilon_1, \upsilon_2) = \frac{1}{2}\Bigg[\operatorname{acosh}\left(1 + \Delta\boldsymbol{\mu}^T\boldsymbol{\Sigma}_1^{-1}\Delta\boldsymbol{\mu}\right)^2 + \left\|\log\left(\boldsymbol{\Sigma}_2^{-\frac{1}{2}}\left(\boldsymbol{\Sigma}_1 + \frac{1}{2}\Delta\boldsymbol{\mu}\Delta\boldsymbol{\mu}^T\right)\boldsymbol{\Sigma}_2^{-\frac{1}{2}}\right)\right\|_2^2\Bigg].$$

3.2 . Riemannian optimization on $\mathcal{M}_p$ and estimation of centers of mass

3.2.1 . Riemannian optimization

In machine learning, some important clustering-classification algorithms, *e.g. K-means++* or the *Nearest centroïd* classifier, require a divergence and an algorithm to compute centers of mass. Since we proposed two divergences in Corollaries 1 and 2, it only remains to explicit an algorithm to compute centers of mass. Such an algorithm relies on optimization on the Riemannian manifold $\mathcal{M}_p$. Hence, we begin by presenting tools to perform gradient based optimization on $\mathcal{M}_p$. In this subsection we consider a function $h : \mathcal{M}_p \mapsto \mathbb{R}$. The objective is to find the parameter υ minimizing h on $\mathcal{M}_p$,

$$\underset{\upsilon\in\mathcal{M}_p}{\text{minimize}}\ h(\upsilon). \tag{3.21}$$

Since $\mathcal{M}_p$ is a Riemannian manifold, we leverage the framework of optimization on Riemannian manifolds [1] to compute (3.21). Thus, we provide two important tools for Riemannian optimization, both associated with the metric (3.7) :

Algorithm 5: Riemannian gradient descent [1]

Input : Initial iterate $\upsilon_1 \in \mathcal{M}_p$.
Output: Sequence of iterates $\{\upsilon_k\}$.
$k := 1$;
while *no convergence* **do**
  Compute a step size α (see [1, Ch. 4]) and set
  $\upsilon_{k+1} := R_{\upsilon_k}^{\mathcal{M}_p}(-\alpha \operatorname{grad}_{\mathcal{M}_p} h(\upsilon_k))$;
  $k := k + 1$;
end

- the Riemannian gradient in the Proposition 10,
- a second order retraction in the Proposition 11 (approximation of the geodesic (3.11) with lower calculation cost and better numerical stability).

With these tools, we can apply gradient based algorithms on $\mathcal{M}_p$ to minimize h. The corresponding Riemannian gradient descent is given in the Algorithm 5.

Proposition 10 (Riemannian gradient)**.** *Let $\upsilon \in \mathcal{M}_p$, the Riemannian gradient of h at υ is*

$$\operatorname{grad}_{\mathcal{M}_p} h(\upsilon) = P_{\upsilon}^{\mathcal{M}_p}\left(\boldsymbol{\Sigma}\boldsymbol{G}_{\boldsymbol{\mu}}, 2\boldsymbol{\Sigma}\boldsymbol{G}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}\right)$$

where $\forall \xi \in \mathbb{R}^p \times \mathbb{R}^{p\times p}$, $P_{\upsilon}^{\mathcal{M}_p}(\xi) = (\boldsymbol{\xi}_{\boldsymbol{\mu}}, \operatorname{sym}(\boldsymbol{\xi}_{\boldsymbol{\Sigma}}))$, with $\operatorname{sym}(\boldsymbol{\xi}) = \frac{1}{2}(\boldsymbol{\xi} + \boldsymbol{\xi}^T)$, is the orthogonal projection according to the FIM (3.7) onto $T_{\upsilon}\mathcal{M}_p$ and $\operatorname{grad}_{\epsilon} h(\upsilon) = (\boldsymbol{G}_{\boldsymbol{\mu}}, \boldsymbol{G}_{\boldsymbol{\Sigma}})$ is the Euclidean gradient of h in $\mathbb{R}^p \times \mathbb{R}^{p\times p}$.

Proof. See Appendix 3.A.1. □

Proposition 11 (Second order retraction)**.** *A second order retraction at $\upsilon \in \mathcal{M}_p$ of $\xi \in T_{\upsilon}\mathcal{M}_p$ is,*

$$R_{\upsilon}^{\mathcal{M}_p}(\xi) = \left(\boldsymbol{\mu} + \boldsymbol{\xi}_{\boldsymbol{\mu}} + \frac{1}{2}\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\mu}}, \boldsymbol{\Sigma} + \boldsymbol{\xi}_{\boldsymbol{\Sigma}} + \frac{1}{2}\left(\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\Sigma}} - \boldsymbol{\xi}_{\boldsymbol{\mu}}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T\right)\right).$$

Proof. See Appendix 3.A.2. □

3.2.2 . Estimation of centers of mass

We now have all the elements to compute centers of mass of sets of points $S = \{\upsilon_i\}_{i=1}^M \subset \mathcal{M}_p$. These centers are associated with divergences, which in our case are the divergences $\delta_{\mathcal{M}_p} \in \{\delta_{c,\mathcal{M}_p}, \delta_{\perp,\mathcal{M}_p}\}$, defined in the

subsection 3.1.2. Similarly to (3.3), the Riemannian center of mass υ is defined as the minimizer of the variance of S

$$\upsilon = \underset{\upsilon \in \mathcal{M}_p}{\arg\min} \frac{1}{M} \sum_{i=1}^{M} \delta_{\mathcal{M}_p}(\upsilon, \upsilon_i). \tag{3.22}$$

Hence, gradient based algorithms can be applied to achieve (3.22) (*e.g.* using Algorithm 5). The only remaining element to compute is the Riemannian gradient of the variance defined in (3.22). Using the Proposition 10, computing the Riemannian gradient of the variance defined in (3.22) amounts to computing its Euclidean gradient. The latter is easily numerically computed using automatic differentiation libraries like Autograd [79] or JAX [25].

3.3 . Application

In this subsection, we provide an application of the theoretical framework developed earlier on the large-scale crop type mapping dataset *Breizhcrops* [118], presented in Chapter 1 Subsection 1.1.3. To classify these crops, we apply a *Nearest centroïd classifier* algorithm on descriptors as presented in Chapter 1. We recall that this classification algorithm works in three steps.

1. For each crop $\boldsymbol{X} \in \mathbb{R}^{p \times n}$, a descriptor is computed (*e.g.* the sample mean or the SCM (3.6)).

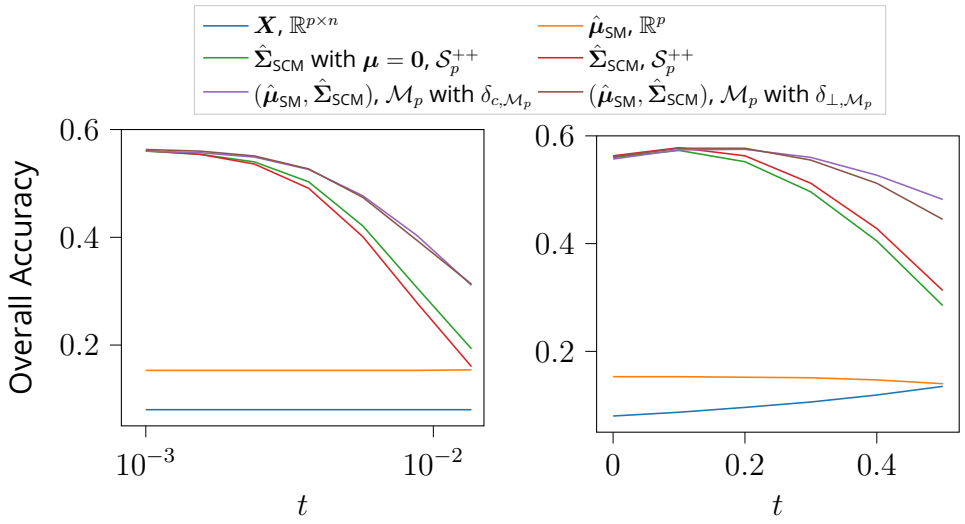
2. Then, on the training set, the center of mass of the descriptors of each class is computed.

3. Finally, on the test set, each descriptor is associated with the nearest center of mass.

Thus, we get a classification of the $\boldsymbol{X}$. The different descriptors used in the application are the following.

- Two descriptors are the batches themselves $\boldsymbol{X}$ and their sample means $\hat{\boldsymbol{\mu}}_{\text{SM}}$ (3.6). Their associated geometry is the Euclidean one with the Euclidean distance as presented in Chapter 1 Example 2. The center of mass is the classical element-wise arithmetic mean.

- Then, two estimators are the SCMs $\hat{\boldsymbol{\Sigma}}_{\text{SCM}}$ (3.6) with location assumed to be known or not. In the case of known location, the SCM is estimated as $\hat{\boldsymbol{\Sigma}}_{\text{SCM}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^T$ whereas in the case of unknown location it is estimated as $\hat{\boldsymbol{\Sigma}}_{\text{SCM}} = \frac{1}{n}\sum_{i=1}^{n} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{\text{SM}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{\text{SM}})^T$. The associated geometry is $\mathcal{S}_p^{++}$ as presented in Equations (3.1) and (3.3).



(a) Rotation transformation: $\boldsymbol{x}_i \mapsto \boldsymbol{Q}(t)^T\boldsymbol{x}_i$ with $\boldsymbol{Q}(0) = \boldsymbol{I}_p$

(b) Scale transformation: $\boldsymbol{x}_i \mapsto (1-t)\boldsymbol{x}_i$

Figure 3.2: “Overall Accuracy" versus the parameter t of two different data transformations applied to the test set of the *Breizhcrops* dataset. The different *Nearest centroïd classifiers* estimate the centers of mass on the training set. Then, the classification is performed on the test set which can undergo two transformations: a rotation transformation and a scale transformation. For $t = 0$, the test set is unchanged and then the larger the t the more the test set is transformed. Six different *Nearest centroïd classifiers* are compared: each one is a combination of an estimator, a divergence and its associated center of mass computation. The two proposed one are denoted “$(\hat{\boldsymbol{\mu}}_{\text{SM}}, \hat{\boldsymbol{\Sigma}}_{\text{SCM}})$, $\mathcal{M}_p$ with $\delta_{c,\mathcal{M}_p}$” and “$(\hat{\boldsymbol{\mu}}_{\text{SM}}, \hat{\boldsymbol{\Sigma}}_{\text{SCM}})$, $\mathcal{M}_p$ with $\delta_{\perp,\mathcal{M}_p}$".

- Finally, two descriptors use both the sample mean and the SCM, $(\hat{\boldsymbol{\mu}}_{\text{SM}}, \hat{\boldsymbol{\Sigma}}_{\text{SCM}})$ from (3.6). These estimators are used with the geometry $\mathcal{M}_p$ and the two divergences $\delta^2_{c,\mathcal{M}_p}$ and $\delta^2_{\perp,\mathcal{M}_p}$ presented in Corollaries 1 and 2 respectively. Riemannian centers of mass are computed using Algorithm 5, implemented using Pymanopt [132] (the python version of Manopt [24]).

The goal of this application is to show the robustness of the proposed methods when the test set undergoes transformations. The intuition is that using both the location and the covariance matrix, instead of only using the covariance matrix, should improve the robustness of the classifier. To do so, we estimate the centers of mass on the raw training set and then we classify a transformed version of the test set. The objective is to keep good performance while the test set is being transformed. Denoting $\boldsymbol{x}_i$ the vectors in the test set, the two continuous transformations are the following.

- The rotation transformation is: $\boldsymbol{x}_i \mapsto \boldsymbol{Q}(t)^T \boldsymbol{x}_i$ for all $t \in [0, 1]$ where $\boldsymbol{Q}(t) = \exp(t\boldsymbol{\xi})$ with $\boldsymbol{\xi}^T = -\boldsymbol{\xi}$.
- The scale transformation is: $\boldsymbol{x}_i \mapsto (1-t)\boldsymbol{x}_i$ for all $t \in [0, 1]$.

It should be noted that when $t = 0$, the test set is left unchanged and as t is increased, the test set undergoes an increasingly important transformation.

Figure 3.2 presents the Overall Accuracy results of the different descriptors and geometries on the *Breizhcrops* dataset. First of all, we observe that the estimators using $\hat{\boldsymbol{\Sigma}}_{\text{SCM}}$ along with the FIM clearly outperform the others whatever the transformation and its intensity. Furthermore, all the estimators/geometries perform equally well when $t = 0$ (no transformation of the test set). However, when t is increased, the two proposed methods that use $\delta_{\mathcal{M}_p} \in \{\delta_{c,\mathcal{M}_p}, \delta_{\perp,\mathcal{M}_p}\}$ perform better than those using only $\hat{\boldsymbol{\Sigma}}_{\text{SCM}}$ along with the geometry of $\mathcal{S}_p^{++}$. This shows the interest of considering both first and second order statistics along with the FIM for classification. A final remark is that the *Nearest centroïd classifier* using $\delta_{c,\mathcal{M}_p}$ performs slightly better than the one using $\delta_{\perp,\mathcal{M}_p}$ when the scale transformation is applied. A good point is that the difference in performance between $\delta_{c,\mathcal{M}_p}$ and $\delta_{\perp,\mathcal{M}_p}$ is marginal compared to the difference with the other methods when a transformation is applied. This means that the proposed *Nearest centroïd classifier* is robust to the chosen intermediate point of the different triangles.

So far, we have proposed two affine invariant divergences that handle both first and second order statistics of the Gaussian distribution. The Riemannian geometry associated with the FIM has been studied and an algorithm to compute Riemannian centers of mass associated with these divergences has been proposed. Finally, these tools have been applied on a classification problem to show the interest of the proposed method.

3.4 . Non centered mixture of scaled Gaussian distributions

3.4.1 . From the Gaussian distribution to the mixture of scaled Gaussian distributions

Many signal processing and machine learning tasks require estimates of the first and second order statistical moments of the sample set $\{\boldsymbol{x}_i\}_{i=1}^n$ [84, 82, 58, 117]. An example of such an application has been given in the previous section. In the latter, these first and second order moments have been estimated using the empirical mean and the SCM that correspond to the MLE of the Gaussian distribution. However, these estimates tend to perform poorly in the context of heavy-tailed distributions or when the set contains outliers, which motivates the use of robust estimation methods. In such setups, one can obtain a better fit to empirical distributions by considering more general statistical models, such as the elliptical distributions [74]. Within this broad family of distributions, M-estimators of the location and scatter [86] appear as generalized MLEs and have been leveraged for their robustness properties in many fields (see [104] for an extensive review).

An important subfamily of elliptical distributions are the compound Gaussian distributions, which models samples as $\boldsymbol{x} \overset{d}{=} \boldsymbol{\mu} + \sqrt{\tau}\boldsymbol{u}$, where $\boldsymbol{\mu} \in \mathbb{R}^p$ is the center (also referred as location) of the distribution, $\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ is the *speckle* (centered Gaussian distribution with covariance matrix $\boldsymbol{\Sigma}$), and $\tau \in \mathbb{R}^+$ is an independent random scaling factor called the *texture*. The flexibility regarding the choice of the PDF for τ results in various models for $\boldsymbol{x}$. Compound Gaussian distributions encompass for example the t-distribution (that also includes the Cauchy distribution), and the K-distribution. In practice, the underlying distribution is generally unknown, which is why the textures have often been modeled as unknown and deterministic in the centered case, *i.e.*, $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{0}, \tau_i\boldsymbol{\Sigma})$. Such models have been presented Chapter 1 Section 1.3 and are referred to as *mixture of scaled Gaussian distributions* (MSG) [141]. The MLE of the covariance matrix $\boldsymbol{\Sigma}$ of this model coincides with Tyler's M-estimator of the scatter up to a scale factor [136], which attracted considerable activity due to its robustness and distribution-free properties over the elliptical distributions family [46, 110, 55, 149]. However, its transposition to the non-centered case from the model $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \tau_i\boldsymbol{\Sigma})$ received less interest[1]. This might notably be due to the fact that the usual fixed-point iterations to evaluate its maximum likelihood may diverge in practice, which motivated the present work.

In the following sections, we tackle optimization problems related to

[1]Notice that D. E. Tyler also proposes an M-estimator of location and scatter that is a solution of a fixed point equation in [136]. While the MLE of $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{0}, \tau_i\boldsymbol{\Sigma})$ and Tyler's estimator (*i.e.*, scatter only) coincide, this is not the case for the non-centered model [40].

parameter estimation and classification for non-centered mixture of scaled Gaussian distributions (NC-MSG). The contribution are threefold:

First, we derive a Riemannian gradient descent and a Riemannian conjugate gradient algorithms based on the Riemannian manifold of the parameter space (location, covariance, textures) endowed with a product Riemannian metric. These algorithms are simple to derive and enough to cast a gradient descent applicable to any function of the parameters. However, they are slow in practice. Hence, we derive a second Riemannian gradient descent algorithm. The latter is based on the Fisher-Rao information geometry of the considered statistical model. Indeed the parameter space is endowed with the FIM of the NC-MSG and is inherently well suited to the natural geometry of the data [97]. In this scope we derive the Riemannian gradient (also referred to as natural gradient) and a second order retraction of this geometry in order to develop a Riemannian gradient descent. We focus on two main examples that are regularized maximum likelihood estimation and center of mass computation. Simulations evidence that this last algorithm allows for a fast computation of critical points, as it can converge with up to one order of magnitude less of iterations compared to the two previous Riemannian descent approaches.

The second line of contributions concerns the problem of maximum likelihood estimation, for which we propose a new class of regularization penalties. A main issue with NC-MSGs is that the existence of the maximum likelihood is not guaranteed. This is due to attraction points where the likelihood function diverges. This also explains why standard fixed-point algorithms to evaluate the solution may diverge in practice. Related issues are well known in the context of M-estimators because their existence is subject to strict conditions that are not always met in practice [86, 136, 104], for example when there is insufficient sample support ($n < p$). In such setups, it is now common to rely on regularization penalties to ensure the existence of a solution, and the stability of corresponding iterative algorithms. In the centered case of elliptical distributions, several works considered shrinkage of M-estimators to a target covariance matrix [109, 126, 103], and regularizing both the mean and the covariance for the non-centered t-distribution was studied in [125]. Other regularization formulated on the spectrum of the covariance matrix were proposed in [141, 29, 144] for the centered case. For NC-MSGs, we propose here a family of penalties that can be interpreted as a divergence between the initial model and a white Gaussian one (*i.e.*, that shrinks both the textures and eigenvalues of the covariance matrix to a pre-defined $\kappa \in \mathbb{R}_*^+$). We derive the general conditions for these penalties to ensure existence of a solution of the regularized MLE. Interestingly, we show that this existence is only conditioned to the design of the penalty, and does not depend on the size of the sample support. We also also study the

invariance properties of the resulting estimators.

Finally, we apply the proposed algorithms to perform Riemannian classification. We consider the framework where statistical features of sample batches are used to discriminate between classes [8, 134, 135, 54]. The Riemannian approach then consists in generalizing usual classification algorithms (*e.g.*, the *Nearest centroïd classifier*) by replacing the Euclidean distance and arithmetic mean by a divergence and its corresponding center of mass [75, 5, 38]. In particular, this framework has been presented in Chapter 1. In this setup, the information geometry can help in designing meaningful distances between the features, and improve the output performance [8, 54]. Unfortunately, the geodesic distance associated with the Fisher information metric of the NC-MSG remains unobtainable in closed form (it is still unknown for the non-centered multivariate Gaussian model [32, 129, 41]). Instead, we propose to rely on the Kullback-Leibler (KL) divergence and its associated center of mass (computed using the proposed Riemannian optimization algorithm). We apply such Riemannian classification framework to the *Breizhcrops* dataset [118]. Our experiments evidence that regularizing the estimation greatly improves the accuracy. Thanks to the invariance properties of the proposed estimators, we we also show that this process exhibits a good robustness to rigid transformations of the samples during the inference.

The rest of the chapter is organized as follows. The next subsection presents NC-MSGs and casts their parameter space as a manifold. Section 3.5 presents a Riemannian geometry with a product metric for the parameter space of the NC-MSG. Section 3.6 studies the Fisher-Rao information geometry of the NC-MSG. Section 3.7 discusses parameter estimation in the considered model, presents a new class of regularized estimators, and studies some of their properties (existence, invariances). Section 3.8 derives the KL divergence of the model, and its associated center of mass. Section 3.9 concludes with validation simulations, and an application to Riemannian classification of the *Breizhcrops* dataset.

3.4.2 . Non-centered mixture of scaled Gaussian distributions

Let a set of n data points $\{\boldsymbol{x}_i\}_{i=1}^n$ belonging to $\mathbb{R}^p$ and distributed according to the following statistical model

$$\boldsymbol{x}_i \overset{d}{=} \boldsymbol{\mu} + \sqrt{\tau_i}\, \boldsymbol{\Sigma}^{\frac{1}{2}}\, \boldsymbol{u}\,, \tag{3.23}$$

where $\boldsymbol{u}$ follows a centered circular Gaussian distribution *i.e.* $\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_p)$. The variables $\boldsymbol{\mu} \in \mathbb{R}^p$ and $\boldsymbol{\Sigma} \in \mathcal{S}_p^{++}$ are respectively named the location and covariance parameters. Then, the unknown texture parameters $\{\tau_i\}_{i=1}^n$ are stacked into the vector $\boldsymbol{\tau} \in (\mathbb{R}_*^+)^n$. If these textures admit a PDF, then the random variables $\boldsymbol{x}_i$ follow a Compound Gaussian distribution [105, 104]. However, in general, this PDF is unknown. Hence, to be robust to any

underlying Compound Gaussian distributions the textures are often assumed to be deterministic [108, 110]. In this case, the random variable $\boldsymbol{x}_i$ follow a NC-MSG, *i.e.*

$$\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \tau_i \boldsymbol{\Sigma}). \tag{3.24}$$

Thus $\boldsymbol{x}_i$ admits a PDF f defined from the Gaussian one f_{G}

$$f(\boldsymbol{x}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \tau_i) = f_{\mathsf{G}}(\boldsymbol{x}_i; \boldsymbol{\mu}, \tau_i \boldsymbol{\Sigma}) \tag{3.25}$$

with $\forall \boldsymbol{x} \in \mathbb{R}^p$

$$f_{\mathsf{G}}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{p}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]. \tag{3.26}$$

The NLL of the sample set $\{\boldsymbol{x}_i\}_{i=1}^n$ is then defined on the set of parameters $\theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\tau}) \in \mathbb{R}^p \times \mathcal{S}_p^{++} \times (\mathbb{R}_*^+)^n$ as (neglecting terms not depending on θ)

$$\mathcal{L}(\theta | \{\boldsymbol{x}_i\}_{i=1}^n) = \frac{1}{2}\sum_{i=1}^n \left[\log|\tau_i \boldsymbol{\Sigma}| + \frac{(\boldsymbol{x}_i-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu})}{\tau_i}\right]. \tag{3.27}$$

One can observe the presence of an ambiguity between the textures $\boldsymbol{\tau}$ and the scatter matrix $\boldsymbol{\Sigma}$. Indeed, $\forall \alpha \in \mathbb{R}_*^+$, we have

$$\mathcal{L}\left(\boldsymbol{\mu}, \alpha\boldsymbol{\Sigma}, \alpha^{-1}\boldsymbol{\tau} | \{\boldsymbol{x}_i\}_{i=1}^n\right) = \mathcal{L}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\tau} | \{\boldsymbol{x}_i\}_{i=1}^n\right). \tag{3.28}$$

Thus, to identify the textures and covariance matrix parameters, a constraint on $\boldsymbol{\tau}$ or $\boldsymbol{\Sigma}$ can be added. Here the choice is made to constrain the textures by fixing their product to be equal to one, *i.e.* $\prod_{i=1}^n \tau_i = 1$. We point out that most of the results in the rest of the chapter could be obtained by constraining the covariance matrix instead of the textures, with a unit determinant constraint, *i.e.* $|\boldsymbol{\Sigma}| = 1$ [27, 18]. The parameter space of interest is

$$\mathcal{M}_{p,n} = \mathbb{R}^p \times \mathcal{S}_p^{++} \times \mathcal{S}(\mathbb{R}_*^+)^n \tag{3.29}$$

where $\mathcal{S}(\mathbb{R}_*^+)^n$ is the set of textures with unit product,

$$\mathcal{S}(\mathbb{R}_*^+)^n = \left\{\boldsymbol{\tau} \in (\mathbb{R}_*^+)^n : \prod_{i=1}^n \tau_i = 1\right\}. \tag{3.30}$$

The choice of adding a constraint is motivated by two results additional to the identifiability:

- it reduces the dimension of the parameter space by removing the indeterminacy (3.28),

- the associated FIM (see Proposition 12) admits a simpler expression, which will be instrumental from Section 3.6 as it turns $\mathcal{M}_{p,n}$ into a Riemannian manifold. Its simple formula could not have been obtained without adding this constraint (either on $\boldsymbol{\tau}$ or its counterpart on $\boldsymbol{\Sigma}$).

In the rest of the chapter, the goal is to optimize several cost functions $h : \mathcal{M}_{p,n} \to \mathbb{R}$. Notably, two cost functions are studied in this chapter: a regularized NLL in Section 3.7, and a cost function to compute centers of mass of sets of points $\{\theta_i\} \subset \mathcal{M}_{p,n}$ in Section 3.8. To do so, $\mathcal{M}_{p,n}$ is turned into two different Riemannian manifolds. The first one, denoted $\mathcal{M}_{p,n}^{\text{Dec.}}$, is described in Section 3.5 and uses a "product Riemannian metric". This metric allows a simple derivation of geometric tools (exponential mapping, parallel transport, ...) since it leverages three well known Riemannian manifolds. However, the induced optimizers are slow as shown in the numerical experiments of Section 3.9. Thus, a second Riemannian manifold, denoted $\mathcal{M}_{p,n}^{\text{FIM}}$, is developed. This one uses the FIM of the NC-MSG. The geometric tools are harder to derive but the induced optimizers are faster than those of $\mathcal{M}_{p,n}^{\text{Dec.}}$ in the numerical experiments of Section 3.9. In Sections (3.5 and 3.6) $\mathcal{M}_{p,n}^{\text{Dec.}}$ and $\mathcal{M}_{p,n}^{\text{FIM}}$ are presented.

In Chapter 1 Section 1.3, we introduced NC-MSG dealing with non-Gaussian data. Indeed, the estimation of their scatter matrices reduce to the Tyler's M-estimator when the location is known [136]. This estimator is known for its many good properties such as its distribution-free property over the class of elliptically contoured distributions. However, the extension of the Tyler's M-estimator to non-centered sample sets, *i.e.* when the location is unknown, is not straightforward.

3.5 . $\mathcal{M}_{p,n}^{\text{Dec.}}$: parameter space $\mathcal{M}_{p,n}$ endowed with a product Riemannian metric

In this section $\mathcal{M}_{p,n}$ is turned into a Riemannian geometry using a "product Riemannian metric". To do so, we begin by defining the ambient space of the parameter space $\mathcal{M}_{p,n}$,

$$\mathcal{E}_{p,n} = \mathbb{R}^p \times \mathbb{R}^{p \times p} \times \mathbb{R}^n. \tag{3.31}$$

Therefore, the tangent space of $\mathcal{M}_{p,n}$ at θ is a subspace of the ambient space $\mathcal{E}_{p,n}$

$$T_\theta \mathcal{M}_{p,n} = \{\xi = (\boldsymbol{\xi}_{\boldsymbol{\mu}}, \boldsymbol{\xi}_{\boldsymbol{\Sigma}}, \boldsymbol{\xi}_{\boldsymbol{\tau}}) \in \mathbb{R}^p \times \mathcal{S}_p \times \mathbb{R}^n : \boldsymbol{\xi}_{\boldsymbol{\tau}}^T \boldsymbol{\tau}^{\odot -1} = 0\} \tag{3.32}$$

where $\mathcal{S}_p$ is the set of $p \times p$ symmetric matrices and $.^{\odot -1}$ is the elementwise inverse operator.

3.5.1 . Riemannian geometry

To turn $\mathcal{M}_{p,n}$ into a Riemannian geometry, we introduce a Riemannian metric in Definition 40.

Definition 40 (Product Riemannian metric)**.** *Let $\theta \in \mathcal{M}_{p,n}$ and $\xi, \eta \in \mathcal{E}_{p,n}$, an inner product on $\mathcal{M}_{p,n}$ is defined by*

$$\langle \xi, \eta \rangle_\theta^{\mathcal{M}_{p,n}^{Dec.}} = \boldsymbol{\xi}_{\boldsymbol{\mu}}^T \boldsymbol{\eta}_{\boldsymbol{\mu}} + \mathrm{Tr}(\boldsymbol{\Sigma}^{-1} \boldsymbol{\xi}_{\boldsymbol{\Sigma}}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\eta}_{\boldsymbol{\Sigma}}) + (\boldsymbol{\xi}_{\boldsymbol{\tau}} \odot \boldsymbol{\tau}^{\odot -1})^T (\boldsymbol{\eta}_{\boldsymbol{\tau}} \odot \boldsymbol{\tau}^{\odot -1})$$

where $\odot$ is the elementwise product operator. Restricted to elements $\xi, \eta \in T_\theta \mathcal{M}_{p,n}$, $(\xi, \eta) \mapsto \langle \xi, \eta \rangle_\theta^{\mathcal{M}_{p,n}^{Dec.}}$ defines a Riemannian metric on $\mathcal{M}_{p,n}$ which becomes a Riemannian manifold and is denoted $\mathcal{M}_{p,n}^{Dec.}$.

The Riemannian metric from Definition 40 is called a "product Riemannian metric" since it can be written as the sum of three independent Riemannian metrics of three Riemannian manifolds: $\mathbb{R}^p$, $\mathcal{S}_p^{++}$ and $\mathcal{S}(\mathbb{R}_*^+)^n$. Indeed, for all $\xi, \eta \in T_\theta \mathcal{M}_{p,n}$, it is rewritten as

$$\langle \xi, \eta \rangle_\theta^{\mathcal{M}_{p,n}^{\mathrm{Dec.}}} = \langle \boldsymbol{\xi}_{\boldsymbol{\mu}}, \boldsymbol{\eta}_{\boldsymbol{\mu}} \rangle_{\boldsymbol{\mu}}^{\mathbb{R}^p} + \langle \boldsymbol{\xi}_{\boldsymbol{\Sigma}}, \boldsymbol{\eta}_{\boldsymbol{\Sigma}} \rangle_{\boldsymbol{\Sigma}}^{\mathcal{S}_p^{++}} + \langle \boldsymbol{\xi}_{\boldsymbol{\tau}}, \boldsymbol{\eta}_{\boldsymbol{\tau}} \rangle_{\boldsymbol{\tau}}^{\mathcal{S}(\mathbb{R}_*^+)^n} \tag{3.33}$$

where

- $\langle \boldsymbol{\xi}_{\boldsymbol{\mu}}, \boldsymbol{\eta}_{\boldsymbol{\mu}} \rangle_{\boldsymbol{\mu}}^{\mathbb{R}^p} = \boldsymbol{\xi}_{\boldsymbol{\mu}}^T \boldsymbol{\eta}_{\boldsymbol{\mu}}$,
- $\langle \boldsymbol{\xi}_{\boldsymbol{\Sigma}}, \boldsymbol{\eta}_{\boldsymbol{\Sigma}} \rangle_{\boldsymbol{\Sigma}}^{\mathcal{S}_p^{++}} = \mathrm{Tr}(\boldsymbol{\Sigma}^{-1} \boldsymbol{\xi}_{\boldsymbol{\Sigma}} \boldsymbol{\Sigma}^{-1} \boldsymbol{\eta}_{\boldsymbol{\Sigma}})$,
- $\langle \boldsymbol{\xi}_{\boldsymbol{\tau}}, \boldsymbol{\eta}_{\boldsymbol{\tau}} \rangle_{\boldsymbol{\tau}}^{\mathcal{S}(\mathbb{R}_*^+)^n} = (\boldsymbol{\xi}_{\boldsymbol{\tau}} \odot \boldsymbol{\tau}^{\odot -1})^T (\boldsymbol{\eta}_{\boldsymbol{\tau}} \odot \boldsymbol{\tau}^{\odot -1})$.

Hence, $\mathcal{M}_{p,n}^{\mathrm{Dec.}}$ is a product Riemannian manifold of three well known Riemannian manifolds: $\mathbb{R}^p$, $\mathcal{S}_p^{++}$, and $\mathcal{S}(\mathbb{R}_*^+)^n$ [120, 113, 15, 122]. The Riemannian manifold $\mathcal{S}_p^{++}$ is presented in Chapter 2 Section 2.4 and the Riemannian manifold $\mathcal{S}(\mathbb{R}_*^+)^n$ is deduced from the one of $\mathcal{SS}_p^{++}$, Riemannian manifold of $p \times p$ symmetric positive definite matrices of unit determinant. The Riemannian manifold $\mathbb{R}^p$ is straightforward to derive. The proofs of the following results directly arise from properties of product manifolds [1]. We begin the description of $\mathcal{M}_{p,n}^{\mathrm{Dec.}}$ with the orthogonal projection from $\mathcal{E}_{p,n}$ onto $T_\theta \mathcal{M}_{p,n}$ which is

$$P_\theta^{\mathcal{M}_{p,n}^{\mathrm{Dec.}}}(\xi) = \left(\boldsymbol{\xi}_{\boldsymbol{\mu}}, \mathrm{sym}(\boldsymbol{\xi}_{\boldsymbol{\Sigma}}), \boldsymbol{\xi}_{\boldsymbol{\tau}} - \frac{\boldsymbol{\xi}_{\boldsymbol{\tau}}^T \boldsymbol{\tau}^{\odot -1}}{n} \boldsymbol{\tau} \right) \tag{3.34}$$

where $\mathrm{sym}(\boldsymbol{\xi}) = \frac{1}{2}\left(\boldsymbol{\xi} + \boldsymbol{\xi}^T\right)$. Then, the exponential mapping $\exp_\theta^{\mathcal{M}_{p,n}^{\mathrm{Dec.}}} : T_\theta \mathcal{M}_{p,n} \to \mathcal{M}_{p,n}^{\mathrm{Dec.}}$ is

$$\exp_\theta^{\mathcal{M}_{p,n}^{\mathrm{Dec.}}}(\xi) = \left(\exp_{\boldsymbol{\mu}}^{\mathbb{R}^p}(\boldsymbol{\xi}_{\boldsymbol{\mu}}), \exp_{\boldsymbol{\Sigma}}^{\mathcal{S}_p^{++}}(\boldsymbol{\xi}_{\boldsymbol{\Sigma}}), \exp_{\boldsymbol{\tau}}^{\mathcal{S}(\mathbb{R}_*^+)^n}(\boldsymbol{\xi}_{\boldsymbol{\tau}}) \right) \tag{3.35}$$

with

Algorithm 6: Riemannian gradient descent on $\mathcal{M}_{p,n}^{\text{Dec.}}$

Input: Initialization $\theta^{(0)} \in \mathcal{M}_{p,n}$
Output: Iterates $\theta^{(k)} \in \mathcal{M}_{p,n}$
for $k = 0$ ***to convergence*** **do**
  Compute a step size α (see [1, Ch. 4]) and set
  $\theta^{(k+1)} = R_{\theta^{(k)}}^{\mathcal{M}_{p,n}^{\text{Dec.}}}\left(-\alpha \operatorname{grad}_{\mathcal{M}_{p,n}^{\text{Dec.}}} h(\theta^{(k)})\right)$

- $\exp_{\boldsymbol{\mu}}^{\mathbb{R}^p}(\boldsymbol{\xi}_{\boldsymbol{\mu}}) = \boldsymbol{\mu} + \boldsymbol{\xi}_{\boldsymbol{\mu}}$,
- $\exp_{\boldsymbol{\Sigma}}^{\mathcal{S}_p^{++}}(\boldsymbol{\xi}_{\boldsymbol{\Sigma}}) = \boldsymbol{\Sigma}^{\frac{1}{2}} \exp\left(\boldsymbol{\Sigma}^{-\frac{1}{2}} \boldsymbol{\xi}_{\boldsymbol{\Sigma}} \boldsymbol{\Sigma}^{-\frac{1}{2}}\right) \boldsymbol{\Sigma}^{\frac{1}{2}}$,
- $\exp_{\boldsymbol{\tau}}^{\mathcal{S}(\mathbb{R}_*^+)^n}(\boldsymbol{\xi}_{\boldsymbol{\tau}}) = \boldsymbol{\tau} \odot \exp\left(\boldsymbol{\tau}^{\odot -1} \odot \boldsymbol{\xi}_{\boldsymbol{\tau}}\right)$.

Finally, the parallel transport between $\theta_1 \in \mathcal{M}_{p,n}$ and $\theta_2 \in \mathcal{M}_{p,n}$, denoted $\mathcal{T}_{\theta_1,\theta_2}^{\mathcal{M}_{p,n}^{\text{Dec.}}}$, moves vectors from the first tangent space $T_{\theta_1}\mathcal{M}_{p,n}$ onto the second one $T_{\theta_2}\mathcal{M}_{p,n}$ while preserving the Riemannian metric. For $\xi \in T_{\theta_1}\mathcal{M}_{p,n}$, it writes

$$\mathcal{T}_{\theta_1,\theta_2}^{\mathcal{M}_{p,n}^{\text{Dec.}}}(\xi) = \left(\mathcal{T}_{\boldsymbol{\mu}_1,\boldsymbol{\mu}_2}^{\mathbb{R}^p}(\boldsymbol{\xi}_{\boldsymbol{\mu}}), \mathcal{T}_{\boldsymbol{\Sigma}_1,\boldsymbol{\Sigma}_2}^{\mathcal{S}_p^{++}}(\boldsymbol{\xi}_{\boldsymbol{\Sigma}}), \mathcal{T}_{\boldsymbol{\tau}_1,\boldsymbol{\tau}_2}^{\mathcal{S}(\mathbb{R}_*^+)^n}(\boldsymbol{\xi}_{\boldsymbol{\tau}})\right), \tag{3.36}$$

with

- $\mathcal{T}_{\boldsymbol{\mu}_1,\boldsymbol{\mu}_2}^{\mathbb{R}^p}(\boldsymbol{\xi}_{\boldsymbol{\mu}}) = \boldsymbol{\xi}_{\boldsymbol{\mu}}$,
- $\mathcal{T}_{\boldsymbol{\Sigma}_1,\boldsymbol{\Sigma}_2}^{\mathcal{S}_p^{++}}(\boldsymbol{\xi}_{\boldsymbol{\Sigma}}) = \left(\boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_1^{-1}\right)^{\frac{1}{2}} \boldsymbol{\xi}_{\boldsymbol{\Sigma}} \left(\left(\boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_1^{-1}\right)^{\frac{1}{2}}\right)^T$,
- $\mathcal{T}_{\boldsymbol{\tau}_1,\boldsymbol{\tau}_2}^{\mathcal{S}(\mathbb{R}_*^+)^n}(\boldsymbol{\xi}_{\boldsymbol{\tau}}) = \boldsymbol{\tau}_2 \odot \boldsymbol{\tau}_1^{\odot -1} \odot \boldsymbol{\xi}_{\boldsymbol{\tau}}$.

3.5.2 . Riemannian optimization

To minimize a given smooth function $h : \mathcal{M}_{p,n} \to \mathbb{R}$ using gradient based Riemannian optimization algorithms, it remains to provide

- the Riemannian gradient of h associated with the product Riemannian metric from Definition 40,
- a retraction $R_\theta^{\mathcal{M}_{p,n}^{\text{Dec.}}} : T_\theta\mathcal{M}_{p,n} \to \mathcal{M}_{p,n}$.

These two tools are directly derived from the fact that $\mathcal{M}_{p,n}^{\text{Dec.}}$ is a Riemannian product manifold. Hence, the Riemannian gradient of h at θ is

$$\operatorname{grad}_{\mathcal{M}_{p,n}^{\text{Dec.}}} h(\theta) = P_\theta^{\mathcal{M}_{p,n}^{\text{Dec.}}}\left(\boldsymbol{G}_{\boldsymbol{\mu}}, \boldsymbol{\Sigma}\boldsymbol{G}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}, \boldsymbol{\tau}^{\odot 2} \odot \boldsymbol{G}_{\boldsymbol{\tau}}\right) \tag{3.37}$$

Algorithm 7: Riemannian conjugate gradient on $\mathcal{M}_{p,n}^{\text{Dec.}}$

Input: Initialization $\theta^{(0)} \in \mathcal{M}_{p,n}$
Output: Iterates $\theta^{(k)} \in \mathcal{M}_{p,n}$
$\xi^{(0)} := -\operatorname{grad}_{\mathcal{M}_{p,n}^{\text{Dec.}}} h(\theta^{(0)})$
for $k = 0$ ***to convergence*** **do**
  Compute a step size α (see [1, Ch 4]) and set
  $\theta^{(k+1)} = R_{\theta^{(k)}}^{\mathcal{M}_{p,n}^{\text{Dec.}}}(\alpha \xi^{(k)})$
  Compute β (see [1, Ch 8]) and set
  $\xi^{(k+1)} = -\operatorname{grad}_{\mathcal{M}_{p,n}^{\text{Dec.}}} h(\theta^{(k+1)}) + \beta \mathcal{T}_{\theta^{(k)},\theta^{(k+1)}}^{\mathcal{M}_{p,n}^{\text{Dec.}}}(\xi^{(k)})$

where $\operatorname{grad} h(\theta) = (\boldsymbol{G}_{\boldsymbol{\mu}}, \boldsymbol{G}_{\boldsymbol{\Sigma}}, \boldsymbol{G}_{\boldsymbol{\tau}})$ is the Euclidean gradient of h in $\mathbb{R}^p \times \mathbb{R}^{p\times p} \times \mathbb{R}^n$. A second order retraction on $\mathcal{M}_{p,n}^{\text{Dec.}}$ at θ is

$$R_\theta^{\mathcal{M}_{p,n}^{\text{Dec.}}}(\xi) = \left(R_{\boldsymbol{\mu}}^{\mathbb{R}^p}(\boldsymbol{\xi}_{\boldsymbol{\mu}}), R_{\boldsymbol{\Sigma}}^{\mathcal{S}_p^{++}}(\boldsymbol{\xi}_{\boldsymbol{\Sigma}}), R_{\boldsymbol{\tau}}^{\mathcal{S}(\mathbb{R}_*^+)^n}(\boldsymbol{\xi}_{\boldsymbol{\tau}})\right) \tag{3.38}$$

where

- $R_{\boldsymbol{\mu}}^{\mathbb{R}^p}(\boldsymbol{\xi}_{\boldsymbol{\mu}}) = \boldsymbol{\mu} + \boldsymbol{\xi}_{\boldsymbol{\mu}}$,
- $R_{\boldsymbol{\Sigma}}^{\mathcal{S}_p^{++}}(\boldsymbol{\xi}_{\boldsymbol{\Sigma}}) = \boldsymbol{\Sigma} + \boldsymbol{\xi}_{\boldsymbol{\Sigma}} + \frac{1}{2}\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\Sigma}}$,
- $R_{\boldsymbol{\tau}}^{\mathcal{S}(\mathbb{R}_*^+)^n}(\boldsymbol{\xi}_{\boldsymbol{\tau}}) = N\left(\boldsymbol{\tau} + \boldsymbol{\xi}_{\boldsymbol{\tau}} + \frac{1}{2}\left(\boldsymbol{\xi}_{\boldsymbol{\tau}}^{\odot 2} \odot \boldsymbol{\tau}^{\odot -1}\right)\right)$ with $\forall \boldsymbol{x} \in (\mathbb{R}_*^+)^n$, $N(\boldsymbol{x}) = \left(\prod_{i=1}^n x_i\right)^{-1/n} \boldsymbol{x}$.

With all the presented tools, two Riemannian optimization algorithms are derived: a Riemannian gradient descent (Algorithm 6) and a Riemannian conjugate gradient (Algorithm 7). These two algorithms are implementations of algorithms presented in Chapter 2.2. Unfortunately, they are quite slow in practice (see Section 3.9). Hence, the next section derives the information geometry of the NC-MSG to get faster optimization algorithms.

3.6 . $\mathcal{M}_{p,n}^{\text{FIM}}$: parameter space $\mathcal{M}_{p,n}$ endowed with the Fisher information metric

3.6.1 . Information geometry

The objective of this section is to present the information geometry of the NC-MSG (3.24); *i.e.* the Riemannian geometry of $\mathcal{M}_{p,n}$ with the FIM as a Riemannian metric [3]. It is expected to give faster optimization algorithms than those associated with the product Riemannian metric from Definition 40. This intuition is confirmed in Section 3.9 thanks numerical experiments. We begin by deriving the FIM of the statistical model (3.24).

Proposition 12 (Fisher information metric)**.** *Let $\theta \in \mathcal{M}_{p,n}$ and $\xi, \eta \in \mathcal{E}_{p,n}$, the FIM at θ associated with the NLL* (3.27) *is*

$$\begin{aligned}\langle \xi, \eta \rangle_\theta^{\mathcal{M}_{p,n}^{FIM}} = &\sum_{i=1}^{n} \left(\frac{1}{\tau_i}\right) \boldsymbol{\xi}_{\boldsymbol{\mu}}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\eta}_{\boldsymbol{\mu}} + \frac{n}{2} \operatorname{Tr}(\boldsymbol{\Sigma}^{-1} \boldsymbol{\xi}_{\boldsymbol{\Sigma}}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\eta}_{\boldsymbol{\Sigma}}) \\ &+ \frac{p}{2} (\boldsymbol{\xi}_{\boldsymbol{\tau}} \odot \boldsymbol{\tau}^{\odot -1})^T (\boldsymbol{\eta}_{\boldsymbol{\tau}} \odot \boldsymbol{\tau}^{\odot -1})\end{aligned}$$

where $\odot$ is the elementwise product operator. Restricted to elements of the tangent spaces $T_\theta \mathcal{M}_{p,n}$, the FIM defines a Riemannian metric on $\mathcal{M}_{p,n}$ which becomes a Riemannian manifold and is denoted $\mathcal{M}_{p,n}^{FIM}$.

Proof. See Appendix 3.A.3. □

Then, the orthogonal projection according to the FIM from $\mathcal{E}_{p,n}$ onto $T_\theta \mathcal{M}_{p,n}$ is given in Proposition 13.

Proposition 13 (Orthogonal projection)**.** *The orthogonal projection associated with the FIM of the Proposition 12 from $\mathcal{E}_{p,n}$ onto $T_\theta \mathcal{M}_{p,n}$ is*

$$P_\theta^{\mathcal{M}_{p,n}^{FIM}}(\xi) = \left(\boldsymbol{\xi}_{\boldsymbol{\mu}}, \operatorname{sym}(\boldsymbol{\xi}_{\boldsymbol{\Sigma}}), \boldsymbol{\xi}_{\boldsymbol{\tau}} - \frac{\boldsymbol{\xi}_{\boldsymbol{\tau}}^T \boldsymbol{\tau}^{\odot -1}}{n} \boldsymbol{\tau} \right).$$

Proof. See Appendix 3.A.4. □

The orthogonal projection proves useful to derive elements in tangent spaces such as the Riemannian gradient or the Levi-Civita connection. The latter is given for the manifold $\mathcal{M}_{p,n}^{\mathsf{FIM}}$ in the Proposition 14.

Proposition 14 (Levi-Civita connection)**.** *Let $\theta \in \mathcal{M}_{p,n}$ and $\xi, \eta \in T_\theta \mathcal{M}_{p,n}$, the Levi-Civita connection of $\mathcal{M}_{p,n}^{FIM}$ evaluated at θ is,*

$$\nabla_\xi^{\mathcal{M}_{p,n}^{FIM}} \eta = P_\theta^{\mathcal{M}_{p,n}^{FIM}} \left(\overline{\nabla}_\xi^{\mathcal{M}_{p,n}^{FIM}} \eta \right)$$

where

$$\begin{aligned}\overline{\nabla}_\xi^{\mathcal{M}_{p,n}^{FIM}} \eta = &\operatorname{D} \eta[\xi] + \\ &\left(-\frac{1}{2} \left[\left(\frac{\boldsymbol{\xi}_{\boldsymbol{\tau}}^T \boldsymbol{\tau}^{\odot -2}}{\sum_{i=1}^n \frac{1}{\tau_i}} \boldsymbol{I}_p + \boldsymbol{\xi}_{\boldsymbol{\Sigma}} \boldsymbol{\Sigma}^{-1} \right) \boldsymbol{\eta}_{\boldsymbol{\mu}} + \left(\frac{\boldsymbol{\eta}_{\boldsymbol{\tau}}^T \boldsymbol{\tau}^{\odot -2}}{\sum_{i=1}^n \frac{1}{\tau_i}} \boldsymbol{I}_p + \boldsymbol{\eta}_{\boldsymbol{\Sigma}} \boldsymbol{\Sigma}^{-1} \right) \boldsymbol{\xi}_{\boldsymbol{\mu}} \right], \right. \\ &\frac{1}{n} \sum_{i=1}^{n} \left(\frac{1}{\tau_i} \right) \boldsymbol{\eta}_{\boldsymbol{\mu}} \boldsymbol{\xi}_{\boldsymbol{\mu}}^T - \boldsymbol{\xi}_{\boldsymbol{\Sigma}} \boldsymbol{\Sigma}^{-1} \boldsymbol{\eta}_{\boldsymbol{\Sigma}}, \\ &\left. \frac{1}{p} \boldsymbol{\xi}_{\boldsymbol{\mu}}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\eta}_{\boldsymbol{\mu}} \mathbf{1} - \boldsymbol{\xi}_{\boldsymbol{\tau}} \odot \boldsymbol{\eta}_{\boldsymbol{\tau}} \odot \boldsymbol{\tau}^{\odot -1} \right).\end{aligned}$$

Algorithm 8: Riemannian gradient descent on $\mathcal{M}_{p,n}^{\text{FIM}}$

Input: Initialization $\theta^{(0)} \in \mathcal{M}_{p,n}$
Output: Iterates $\theta^{(k)} \in \mathcal{M}_{p,n}$
for $k = 0$ ***to convergence*** **do**
  Compute a step size α (see [1, Ch. 4]) and set
  $\theta^{(k+1)} = R_{\theta^{(k)}}^{\mathcal{M}_{p,n}^{\text{FIM}}}\left(-\alpha \operatorname{grad}_{\mathcal{M}_{p,n}^{\text{FIM}}} h(\theta^{(k)})\right)$

Proof. See Appendix 3.A.5. □

As detailed in Chapter 2, the Levi-Civita connection defines geodesics on a Riemannian manifold. Indeed, for I an open interval of $\mathbb{R}$, a geodesic $\gamma : I \to \mathcal{M}_{p,n}$ with initial position $\gamma(0) = \theta \in \mathcal{M}_{p,n}$ and initial velocity $\dot{\gamma}(0) = \xi \in T_\theta \mathcal{M}_{p,n}$ must respect

$$\nabla_{\dot{\gamma}(t)}^{\mathcal{M}_{p,n}^{\text{FIM}}} \dot{\gamma}(t) = 0, \quad \forall t \in I. \tag{3.39}$$

However an analytical solution of (3.39) remains unknown in this case. A retraction (approximation of the geodesic) can still be obtained (see Proposition 16) which allows us to optimize functions on $\mathcal{M}_{p,n}$. This implies that the geodesic between two points θ_1 and θ_2 is unknown. Thus, the geodesic distance is also unknown. This is not surprising since the geodesic and the Riemannian distance between two Gaussian distributions with different locations are unknown [120, 32, 129, 41]. To alleviate this problem, a divergence associated with the NC-MSG (3.24) is proposed in Section 3.8.

3.6.2 . Riemannian optimization

We propose tools to minimize smooth functions $h : \mathcal{M}_{p,n} \to \mathbb{R}$ with the Riemannian manifold $\mathcal{M}_{p,n}^{\text{FIM}}$. To do so, we consider a Riemannian steepest descent on $\mathcal{M}_{p,n}^{\text{FIM}}$. Only the tools required for this algorithm are derived here:

- the Riemannian gradient of h associated with the FIM from the Proposition 12,
- a retraction that maps tangent vectors from $T_\theta \mathcal{M}_{p,n}$ $\forall \theta \in \mathcal{M}_{p,n}$ onto $\mathcal{M}_{p,n}$.

We begin with the Riemannian gradient of h at θ which can be computed from the Euclidean gradient of h.

Proposition 15 (Riemannian gradient)**.** *Let $\theta \in \mathcal{M}_{p,n}$ and h be a real valued function defined on $\mathcal{M}_{p,n}$. The Riemannian gradient of h at θ is*

$$\operatorname{grad}_{\mathcal{M}_{p,n}^{FIM}} h(\theta) = P_\theta^{\mathcal{M}_{p,n}^{FIM}}\left(\left(\sum_{i=1}^{n}\frac{1}{\tau_i}\right)^{-1}\boldsymbol{\Sigma}\boldsymbol{G}_{\boldsymbol{\mu}}, \frac{2}{n}\boldsymbol{\Sigma}\boldsymbol{G}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}, \frac{2}{p}\boldsymbol{\tau}^{\odot 2}\odot \boldsymbol{G}_{\boldsymbol{\tau}}\right)$$

where $\operatorname{grad} h(\theta) = (\boldsymbol{G}_{\boldsymbol{\mu}}, \boldsymbol{G}_{\boldsymbol{\Sigma}}, \boldsymbol{G}_{\boldsymbol{\tau}})$ is the Euclidean gradient of h in $\mathbb{R}^p \times \mathbb{R}^{p\times p} \times \mathbb{R}^n$.

Proof. See Appendix 3.A.6. □

Then, it remains to define a retraction for every θ on $\mathcal{M}_{p,n}$. A retraction $R_\theta^{\mathcal{M}_{p,n}^{\mathsf{FIM}}}$ maps every $\xi \in T_\theta\mathcal{M}_{p,n}$ to a point $R_\theta^{\mathcal{M}_{p,n}^{\mathsf{FIM}}}(\xi) \in \mathcal{M}_{p,n}$ and is such that $R_\theta^{\mathcal{M}_{p,n}^{\mathsf{FIM}}}(\xi) = \theta + \xi + o(\|\xi\|)$. Several retractions could be obtained from this definition. Furthermore, it should be noted that a map respecting this definition is not necessarily related to the Riemannian metric of $\mathcal{M}_{p,n}^{\mathsf{FIM}}$. Thus, we choose to enforce an additional property: the desired retraction must be a second order retraction. This means that it must have a zero initial acceleration,

$$\nabla_{\dot r(t)}^{\mathcal{M}_{p,n}^{\mathsf{FIM}}}\dot r(t)\Big|_{t=0} = 0 \tag{3.40}$$

where $\dot r(t) = \frac{d}{dt}R_\theta^{\mathcal{M}_{p,n}^{\mathsf{FIM}}}(t\xi)$ and $\nabla^{\mathcal{M}_{p,n}^{\mathsf{FIM}}}$ is the Levi-Civita connection from the Proposition 14. Furthermore, the property of zero initial acceleration is linked to the definition of the geodesic. Indeed, a geodesic has a zero acceleration $\forall t$ along its path (see (3.39)) whereas here this condition is only needed at $t = 0$. By respecting this property, the retraction is associated with the Riemannian metric of the Proposition 12 since the Levi-Civita connection is itself derived from this Riemannian metric. Such a retraction is presented in the Proposition 16.

Proposition 16 (Second order retraction)**.** *Let $\theta \in \mathcal{M}_{p,n}$ and $\xi \in T_\theta\mathcal{M}_{p,n}$. There exists $t_{max} > 0$ (specified in the Appendix 3.A.7) such that $\forall t \in [0, t_{max}[$, a second order retraction on $\mathcal{M}_{p,n}^{FIM}$ at θ is*

$$\begin{aligned} R_\theta^{\mathcal{M}_{p,n}^{FIM}}(t\xi) = \Bigg(&\boldsymbol{\mu} + t\boldsymbol{\xi}_{\boldsymbol{\mu}} + \frac{t^2}{2}\left[\frac{\boldsymbol{\xi}_{\boldsymbol{\tau}}^T\boldsymbol{\tau}^{\odot -2}}{\sum_{i=1}^n \frac{1}{\tau_i}}\boldsymbol{I}_p + \boldsymbol{\xi}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\right]\boldsymbol{\xi}_{\boldsymbol{\mu}},\\ &\boldsymbol{\Sigma} + t\boldsymbol{\xi}_{\boldsymbol{\Sigma}} + \frac{t^2}{2}\left(\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\Sigma}} - \frac{1}{n}\sum_{i=1}^n\left(\frac{1}{\tau_i}\right)\boldsymbol{\xi}_{\boldsymbol{\mu}}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T\right),\\ &N\left(\boldsymbol{\tau} + t\boldsymbol{\xi}_{\boldsymbol{\tau}} + \frac{t^2}{2}\left(\boldsymbol{\xi}_{\boldsymbol{\tau}}^{\odot 2}\odot\boldsymbol{\tau}^{\odot -1} - \frac{1}{p}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\mu}}\mathbf{1}\right)\right)\Bigg) \end{aligned}$$

where $\forall \boldsymbol{x} \in (\mathbb{R}_*^+)^n$, N is defined as $N(\boldsymbol{x}) = \left(\prod_{i=1}^n x_i\right)^{-1/n}\boldsymbol{x}$.

Proof. See Appendix 3.A.7. □

With this retraction and the Riemannian gradient from Proposition 15, we have all the tools required to derive a Riemannian steepest descent. The latter is presented in Algorithm 8.

3.7 . Estimation of mixtures of scaled Gaussian distributions: existence and regularization

3.7.1 . A pathological example

In the two previous sections, tools to perform optimization on $\mathcal{M}_{p,n}$ have been developed. In this subsection, the objective is to leverage these tools to estimate parameters of the NC-MSG (3.23). In the following, we assume having $n \geq 1$ data points $\{\boldsymbol{x}_i\}_{i=1}^n \subset \mathbb{R}^p$. The estimation of the parameters of the statistical model (3.24) is performed by maximizing the associated likelihood on $\mathcal{M}_{p,n}$:

$$\underset{\theta \in \mathcal{M}_{p,n}}{\text{minimize}} \; \mathcal{L}\left(\theta | \{\boldsymbol{x}_i\}_{i=1}^n\right) \tag{3.41}$$

where $\mathcal{L}$ is the NLL (3.27). However, the existence of a solution to this problem is not guaranteed. To build an intuition, we present a short example of a problematic case where $\boldsymbol{\mu}$ gets attracted by one data point $\boldsymbol{x}_j$. Let k be the current iteration of a given optimizer of (3.41). For $k \to +\infty$, if $\boldsymbol{\mu}^{(k)} \to \boldsymbol{x}_j$ faster than $\tau_j^{(k)} \to 0$ and $\forall i \neq j$, $\tau_i^{(k)} \to +\infty$, then the quadratic form in $\mathcal{L}$ (3.27) tends to zero, which is its minimum,

$$\sum_{i=1}^n \frac{(\boldsymbol{x}_i - \boldsymbol{\mu}^{(k)})^T \left(\boldsymbol{\Sigma}^{(k)}\right)^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}^{(k)})}{\tau_i^{(k)}} \xrightarrow[k \to +\infty]{} 0. \tag{3.42}$$

Then, if an eigenvalue $\lambda^{(k)}$ of $\boldsymbol{\Sigma}^{(k)}$ tends to 0 slower than the respective limits of $\boldsymbol{\mu}^{(k)}$, $\boldsymbol{\tau}_i^{(k)}$ and $\boldsymbol{\tau}_j^{(k)}$ and since $\sum_{i=1}^n \log|\tau_i \boldsymbol{\Sigma}| = n \log(\boldsymbol{\Sigma})$, we obtain that

$$\mathcal{L}\left(\theta^{(k)} | \{\boldsymbol{x}_i\}_{i=1}^n\right) \xrightarrow[k \to +\infty]{} -\infty. \tag{3.43}$$

Hence, depending on the data points $\{\boldsymbol{x}_i\}_{i=1}^n$, a solution of the problem (3.41) does not necessarily exist.

3.7.2 . Regularization and existence

To overcome this issue, we present a regularized version of the NLL (3.27)

$$\mathcal{L}_{\mathcal{R}_\kappa}\left(\theta | \{\boldsymbol{x}_i\}_{i=1}^n\right) = \mathcal{L}\left(\theta | \{\boldsymbol{x}_i\}_{i=1}^n\right) + \beta \mathcal{R}_\kappa(\theta) \tag{3.44}$$

where $\beta \in \mathbb{R}_*^+$ and $\mathcal{R}_\kappa : \mathcal{M}_{p,n} \to \mathbb{R}$ is a regularization. Thus, the minimization problem (3.41) becomes

$$\underset{\theta \in \mathcal{M}_{p,n}}{\text{minimize}} \; \mathcal{L}_{\mathcal{R}_\kappa}\left(\theta | \{\boldsymbol{x}_i\}_{i=1}^n\right). \tag{3.45}$$

Name	$\mathcal{R}_\kappa(\theta)$	$r_\kappa(x)$
L1 penalty	$\left\|(\mathrm{diag}(\boldsymbol{\tau}) \otimes \boldsymbol{\Sigma})^{-1} - \kappa^{-1}\boldsymbol{I}_{n\times p}\right\|_1 = \sum_{i,j}\left\|(\tau_i\lambda_j)^{-1} - \kappa^{-1}\right\|$	$\|x^{-1} - \kappa^{-1}\|$
L2 penalty	$\left\|(\mathrm{diag}(\boldsymbol{\tau}) \otimes \boldsymbol{\Sigma})^{-1} - \kappa^{-1}\boldsymbol{I}_{n\times p}\right\|_2^2 = \sum_{i,j}\left((\tau_i\lambda_j)^{-1} - \kappa^{-1}\right)^2$	$(x^{-1} - \kappa^{-1})^2$
Bures-Wasserstein squared distance	$d^2_{\mathsf{BW}}\left((\mathrm{diag}(\boldsymbol{\tau}) \otimes \boldsymbol{\Sigma})^{-1}, \kappa^{-1}\boldsymbol{I}_{n\times p}\right) = \sum_{i,j}\left((\tau_i\lambda_j)^{-\frac{1}{2}} - \kappa^{-\frac{1}{2}}\right)^2$	$\left(x^{-\frac{1}{2}} - \kappa^{-\frac{1}{2}}\right)^2$
Gaussian KL divergence	$\delta_{\mathsf{KL}}(\kappa\boldsymbol{I}_{n\times p}, \mathrm{diag}(\boldsymbol{\tau}) \otimes \boldsymbol{\Sigma}) = \frac{1}{2}\left[\sum_{i,j}\left(\kappa\,(\tau_i\lambda_j)^{-1} + \log\left(\tau_i\lambda_j\right)\right) - np(1+\log(\kappa))\right]$	$\frac{1}{2}\left[\kappa x^{-1} + \log(x) - (1+\log(\kappa))\right]$

Table 3.1: Examples of regularizations $\mathcal{R}_\kappa$ respecting Assumptions 3, 4 and 5. $\forall q \in \mathbb{N}^*$, $\|.\|_q$ is the Schatten norm, *i.e.* $\forall \boldsymbol{A} \in \mathcal{S}_p\ \|\boldsymbol{A}\|_q^q = \sum_i |\lambda_i|^q$ where λ_i are the eigenvalues of $\boldsymbol{A}$. The diagonal matrix whose elements are those of $\boldsymbol{\tau}$ is denoted $\mathrm{diag}(\boldsymbol{\tau})$. The Kronecker product between matrices is denoted $\otimes$.

Though (3.45) is a generic formulation, we will focus on several proposals that ensure the existence of a solution. The proposed approach is to rewrite $\mathcal{R}_\kappa$ as a sum of regularizations r_κ on the eigenvalues of $\tau_i\boldsymbol{\Sigma}$. This rewriting is formalized in Assumption 3.

Assumption 3. *The regularization $\mathcal{R}_\kappa$ is a sum of regularizations on the eigenvalues of $\tau_i\boldsymbol{\Sigma}$*

$$\mathcal{R}_\kappa(\theta) = \sum_{i=1}^{n}\sum_{j=1}^{p} r_\kappa(\tau_i\lambda_j)$$

where $\lambda_j \in \mathbb{R}_*^+$ are the eigenvalues of $\boldsymbol{\Sigma}$ and $r_\kappa : \mathbb{R}_*^+ \to \mathbb{R}$ is a continuous function.

In the following, we assume that $\mathcal{R}_\kappa$ respects Assumption 3. To prevent the eigenvalues of $\tau_i\boldsymbol{\Sigma}$ to take values that are too large nor too small, a second assumption is added. Indeed, Assumption 4 states that the regularization r_κ goes to infinite when its argument goes to 0^+ or $+\infty$. This assumption is made so that if an eigenvalue of $\tau_i\boldsymbol{\Sigma}$ tends to 0^+ or $+\infty$ then $\mathcal{L}_{\mathcal{R}_\kappa} \to +\infty$.

Assumption 4. *The function r_κ admits the following limit $\forall \beta \in \mathbb{R}_*^+$*

$$\lim_{x \to \partial \mathbb{R}_*^+} \log(x) + \beta r_\kappa(x) = +\infty,$$

with $\partial \mathbb{R}_*^+$ is a border of $\mathbb{R}_*^+$, i.e. 0^+ *or* $+\infty$*.*

Assumptions 3 and 4 are sufficient to solve the problem of existence stated earlier. Indeed, when $\mathcal{R}_\kappa$ respects these assumptions, Proposition 17 states that the problem (3.45) has a solution, *i.e.* $\mathcal{L}_{\mathcal{R}_\kappa}$ admits a minimum in $\mathcal{M}_{p,n}$. Finally, Assumptions 3 and 4 are quite easy to meet in practice. Indeed, several regularizations respecting these assumptions are proposed in Table 3.1.

Proposition 17 (Existence)**.** *Under Assumptions 3 and 4, and $\forall \beta \in \mathbb{R}_*^+$, the regularized NLL*

$$\theta \mapsto \mathcal{L}_{\mathcal{R}_\kappa}\left(\theta | \{\boldsymbol{x}_i\}_{i=1}^n\right) = \mathcal{L}\left(\theta | \{\boldsymbol{x}_i\}_{i=1}^n\right) + \beta \mathcal{R}_\kappa(\theta),$$

with $\mathcal{L}$ being the NLL defined in (3.27)*, admits a minimum in $\mathcal{M}_{p,n}$.*

Proof. See Appendix 3.A.8. □

So far, the regularization has been chosen to guarantee the existence of a solution to the problem (3.45). However, this regularization shrinks the estimation towards an unknown parameter θ. In order to define this parameter, a third assumption is added. Indeed, Assumption 5 states that the regularization $\mathcal{R}_\kappa$ is a divergence (see Definition 41) on the set $\mathcal{S}_p^{++}$. This implies that the minima of $\mathcal{R}_\kappa$ are known and are derived in Proposition 18.

Definition 41 (Divergence)**.** *Given a set E, the function $\delta : E \times E \to \mathbb{R}$ is a divergence if it satisfies the following conditions for all $x, y \in E$*

1. $\delta(x, y) \geq 0$ *(positivity),*
2. $\delta(x, y) = 0$ *if and only if $x = y$ (separability).*

Assumption 5. *The regularization $\mathcal{R}_\kappa$ can be written as*

$$\mathcal{R}_\kappa(\theta) = \delta_{\mathcal{S}_p^{++}}\left(\operatorname{diag}(\boldsymbol{\tau}) \otimes \boldsymbol{\Sigma}, \kappa \boldsymbol{I}_{n \times p}\right)$$

where $\delta_{\mathcal{S}_p^{++}}$ is a divergence on the set $\mathcal{S}_p^{++}$ and $\kappa \in \mathbb{R}_*^+$.

Proposition 18 (Minima of $\mathcal{R}_\kappa$)**.** *Under Assumption 5, the set of minima in $\mathcal{M}_{p,n}$ of the regularization $\mathcal{R}_\kappa$ is*

$$\left\{\theta = (\boldsymbol{\mu}, \kappa \boldsymbol{I}_p, \mathbf{1}) : \boldsymbol{\mu} \in \mathbb{R}^p\right\}.$$

Proof. See Appendix 3.A.9. □

It should be noted that the regularizations from Table 3.1 respect Assumption 5. Thus, the minimum of (3.45) tends to $\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i, \kappa\boldsymbol{I}_p, \mathbf{1}\right)$ as $\beta \to +\infty$. This corresponds to a Gaussian distribution with a covariance matrix proportional to the identity. Thus, the β hyperparameter makes the trade-off between a NC-MSG (3.24) and a circular Gaussian distribution.

We finish this section with a remark on the estimation of the parameter θ when data undergo a rigid transformation. Given $\boldsymbol{Q} \in \mathcal{O}_p$ and $\boldsymbol{\mu}_0 \in \mathbb{R}^p$, the rigid transformation ψ of a set of data $\{\boldsymbol{x}_i\}_{i=1}^n$ is defined as

$$\psi\left(\{\boldsymbol{x}_i\}_{i=1}^n\right) = \left\{\boldsymbol{Q}^T\boldsymbol{x}_i + \boldsymbol{\mu}_0\right\}_{i=1}^n. \tag{3.46}$$

These rigid transformations define isometries on $\mathbb{R}^p$ since

$$\left\|\psi\left(\boldsymbol{x}_i\right) - \psi\left(\boldsymbol{x}_j\right)\right\|_2 = \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2 \tag{3.47}$$

$\forall \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^p$. These are important in machine learning problems since they transform data without changing distances. An important property of the regularized NLL (3.44) is that the estimated textures of the model are invariant under rigid transformations of the data; see Proposition 19. This is interesting since having parameters invariant to these transformations can improve performances when transformations happen between the training and the test sets for a given supervised problem. Numerical experiments in Section 3.9 leverage this property and show robust performances when data undergo a rigid transformation during the testing phase.

Proposition 19 (Minima of $\mathcal{L}_{\mathcal{R}_\kappa}$ and rigid transformations)**.** *Let $\mathcal{R}_\kappa$ be a regularization satisfying Assumption 3, and $\theta^\star = (\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\tau})$ be a minimum of the regularized NLL* (3.45) *computed on data $\{\boldsymbol{x}_i\}_{i=1}^n$,* i.e.

$$\theta^\star = \underset{\theta \in \mathcal{M}_{p,n}}{\arg\min}\ \mathcal{L}_{\mathcal{R}_\kappa}\left(\theta | \{\boldsymbol{x}_i\}_{i=1}^n\right),$$

then, given $\boldsymbol{Q} \in \mathcal{O}_p$ and $\boldsymbol{\mu}_0 \in \mathbb{R}^p$, a minimum of the regularized NLL computed on the transformed data $\psi\left(\{\boldsymbol{x}_i\}_{i=1}^n\right) = \left\{\boldsymbol{Q}^T\boldsymbol{x}_i + \boldsymbol{\mu}_0\right\}_{i=1}^n$ is $\phi(\theta^\star) = \left(\boldsymbol{Q}^T\boldsymbol{\mu} + \boldsymbol{\mu}_0, \boldsymbol{Q}^T\boldsymbol{\Sigma}\boldsymbol{Q}, \boldsymbol{\tau}\right)$, i.e.

$$\phi(\theta^\star) = \underset{\theta \in \mathcal{M}_{p,n}}{\arg\min}\ \mathcal{L}_{\mathcal{R}_\kappa}\left(\theta | \psi\left(\{\boldsymbol{x}_i\}_{i=1}^n\right)\right).$$

Proof. See Appendix 3.A.10. □

3.8 . Classification on $\mathcal{M}_{p,n}$

In the previous section, we proposed to optimize the regularized NLL (3.45) of the NC-MSG (3.24). Once these parameters are estimated, they can be used as features for Riemannian clustering-classification algorithms as presented in Chapter 1. To do this clustering-classification, two tools are presented in this section. Firstly, since no closed form formula of the Riemannian distance on $\mathcal{M}_{p,n}$ is known, a divergence between pairs of parameters is defined. The proposed one is the KL divergence between two NC-MSG (3.24). It benefits from a simple closed form formula that is presented in Subsection 3.8.1. Secondly, simple clustering-classification algorithms, such as *K-means++* or the *Nearest centroïd classifier*, rely on an algorithm to average parameters. Thus, an algorithm to compute centers of mass of estimated parameters θ must be defined. This center of mass is defined using the KL divergence and is presented in Subsection 3.8.2. Its computation is realized with Algorithm 8.

3.8.1 . Kullback-Leibler divergence

Clustering-classification algorithms, such as *K-means++* or the *Nearest centroïd classifier*, rely on a divergence between points. Thus, it remains to define a divergence on $\mathcal{M}_{p,n}$. The latter must be related to the NC-MSG (3.24) since the objective is to classify its parameters θ. In the context of measuring proximities between distributions admitting PDFs, a classical divergence is the KL one. The latter measures the similarity between two PDFs. Definition 42 gives the general formula of the KL divergence.

Definition 42 (KL divergence)**.** *Given two PDFs p and q defined on the sample space $\mathcal{X}$, the KL divergence is*

$$\delta_{KL}(p,q) = \int_{\mathcal{X}} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx.$$

Applied to NC-MSGs, the KL divergence is derived from the Gaussian one and is presented in Proposition 20. It benefits from a simple closed form formula and therefore is of practical interest.

Proposition 20 (KL divergence)**.** *Given the random variable $x = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ and two NC-MSGs of PDFs $p_{\theta_1}(x) = \prod_{i=1}^n f(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \tau_{1,i})$ and $p_{\theta_2}(x) = \prod_{i=1}^n f(\boldsymbol{x}_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2, \tau_{2,i})$ the KL divergence is*

$$\delta_{KL}(\theta_1, \theta_2) = \frac{1}{2}\left(\sum_{i=1}^n \frac{\tau_{1,i}}{\tau_{2,i}} \operatorname{Tr}\left(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1\right) + \sum_{i=1}^n \frac{1}{\tau_{2,i}} \Delta\boldsymbol{\mu}^T \boldsymbol{\Sigma}_2^{-1} \Delta\boldsymbol{\mu} + n \log\left(\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|}\right) - np \right)$$

with $\Delta\boldsymbol{\mu} = \boldsymbol{\mu}_2 - \boldsymbol{\mu}_1$.

Proof. See Appendix 3.A.11. □

Finally, this KL divergence is non-symmetrical. We rely on the classical symmetrization to define the proposed divergence $\delta_{\mathcal{M}_{p,n}} : \mathcal{M}_{p,n} \times \mathcal{M}_{p,n} \to \mathbb{R}$,

$$\delta_{\mathcal{M}_{p,n}}(\theta_1, \theta_2) = \frac{1}{2}\left(\delta_{\mathsf{KL}}(\theta_1, \theta_2) + \delta_{\mathsf{KL}}(\theta_2, \theta_1)\right). \tag{3.48}$$

3.8.2 . Estimation of centers of mass

To implement simple machine learning algorithms, it remains to define an averaging algorithm on $\mathcal{M}_{p,n}$. To do so, we leverage a classical definition of centers of mass which are minimizers of variances [75, 94]. Given a set of parameters $\{\theta_i\}_{i=1}^M$, its center of mass on $\mathcal{M}_{p,n}$ is defined as the solution of

$$\underset{\theta \in \mathcal{M}_{p,n}}{\text{minimize}} \ \frac{1}{M}\sum_{i=1}^{M} \delta_{\mathcal{M}_{p,n}}(\theta, \theta_i) \tag{3.49}$$

where $\delta_{\mathcal{M}_{p,n}}$ is the symmetrized KL divergence from the equation (3.48). To realize (3.49), Algorithm 8 can be employed.

3.9 . Numerical experiments

The objective of this section is to show the practical interests of the tools developed in the previous sections. More precisely, this section presents numerical experiments and is divided into two parts.

First, Subsection 3.9.1 studies the performance of Algorithms 6, 7 and 8, in terms of speed of convergence on the cost functions (3.45) and (3.49) through simulations. Algorithm 8 is shown to be fast. Indeed, it requires from 5 to 30 times less iterations to minimize costs functions (3.45) and (3.49) compared to other sophisticated optimization algorithms. This demonstrates the interest of the choice of the FIM to develop Riemannian optimization algorithms. Also, the estimation error on the cost function (3.41) realized by Algorithm 8 is studied on simulated data. This algorithm gives lower estimation errors than other classical estimators such as the Tyler joint mean-covariance one and the Gaussian ones.

Second, an application on the crop classification dataset *Breizhcrops* [118] is presented in Subsection 3.9.2. This dataset consists of $600\ 000$ time series to be classified into 9 classes. The application implements a *Nearest centroïd classsifier* on $\mathcal{M}_{p,n}$ using the divergence (3.48) and the Riemannian center of mass (3.49). Three results ensue. First, the proposed algorithms can be used on large scale datasets. Second, the proposed regularization in Section 3.7 plays an important role

in classification. Third, considering a NC-MSG (3.23) is interesting for time series especially when data undergo a rigid transformation (3.46).

Python code implementing the different experiments can be found at https://github.com/antoinecollas/optim_compound.

3.9.1 . Simulations

In this simulation setting we set the parameters $\theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\tau}) \in \mathcal{M}_{p,n}$ as follows. First, each component of $\boldsymbol{\mu}$ is sampled from a univariate Gaussian distribution $\mathcal{N}(0,1)$. Second, $\boldsymbol{\Sigma}$ is generated using its eigendecomposition $\boldsymbol{\Sigma} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$. $\boldsymbol{U} \in \mathcal{O}_p$ is drawn from the uniform distribution on $\mathcal{O}_p$ [89] using the module "scipy.stats" from the Scipy library [138]. Then, the elements on the diagonal of the diagonal matrix $\boldsymbol{\Lambda}$ are drawn from a χ^2_1 distribution. Third, the τ_i are drawn from a $\Gamma(\nu, 1/\nu)$ distribution with ν a parameter to be chosen. The smaller the ν, the greater the variance. In order to respect the constraint $\prod_{i=1}^n \tau_i = 1$, the vector $\boldsymbol{\tau}$ is normalized.

The speed of convergence of Algorithms 6, 7 and 8 is studied on two cost functions: the regularized NLL (3.45) and the cost function (3.49) to compute the center of mass associated to the KL divergence of Proposition 20.

We begin with the minimization of the regularized NLL (3.45). $n = 150$ data $\boldsymbol{x}_i \in \mathbb{R}^{10}$ are drawn from a NC-MSG, *i.e.* $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \tau_i \boldsymbol{\Sigma})$. The parameter $\theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\tau})$ of this distribution is generated as explained in the introduction of Subsection 3.9.1 with $\nu = 1$. Different parameters β in (3.45) are considered: $\beta \in \{0, 10^{-5}\}$. The chosen regularization is the L2 penalty from Table 3.1. When $\beta = 0$ the NLL is the plain one, *i.e.* it is not regularized. We point out that, in this setup, the optimization goes well although the existence of a solution to this problem is not proven. When $\beta > 0$ a solution to the minimization problem exists from Proposition 17. The results of this experiment are presented in Figure 3.3. We observe that Algorithm 8 is much faster than the two others regardless of the β parameter. Indeed, for $\beta \in \{0, 10^{-5}\}$, Algorithm 8 is at least 100 times faster than Algorithm 6 and 10 times faster than Algorithm 7.

Then, a similar experiment is performed with the cost function (3.49) to compute centers of mass. $M \in \{2, 100\}$ parameters θ are generated as described in the introduction of Subsection 3.9.1 with $\nu = 1$. The minimization is performed with the same optimization algorithms as previously. The results of this experiment are presented in Figure 3.4. We observe that Algorithm 8 is much faster than the two others regardless of M. Indeed, when $M = 2$ Algorithm 8 converges in 40 iterations whereas Algorithm 6 requires 300 iterations and Algorithm 7 still has not converged after 1000 iterations. When $M = 100$, Algorithm 8 converges in less than 60 iterations which is 4 times faster than Algorithm 7. It should be noted that Algorithm 6 has not converged after 1000 in the case $M = 100$.

Then, the estimation error made by Algorithm 8 applied on the NLL (3.27)

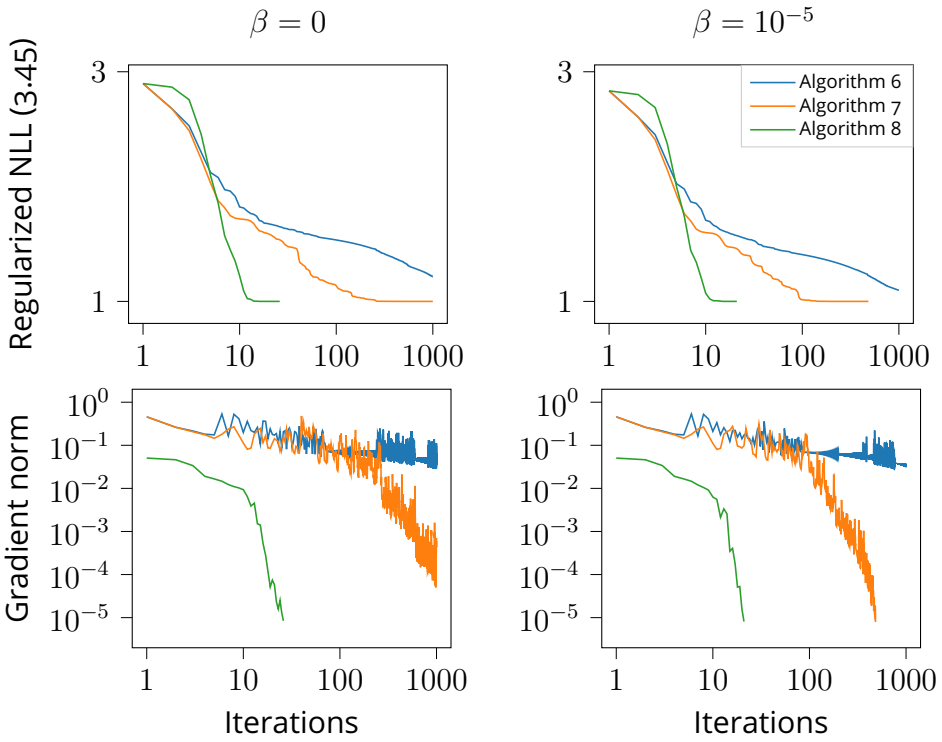


Figure 3.3: Regularized NLL (3.45) and its gradient norm versus the iterations of Algorithms 6, 7 and 8. The chosen regularization is the L2 penalty (see Table 3.1) and two different regularization intensities β are considered: 0 in the left column and 10^{-5} in the right one. Each estimation is performed on $n = 150$ samples in $\mathbb{R}^{10}$ sampled from a NC-MSG. The regularized NLL are normalized so that their minimum value is 1.

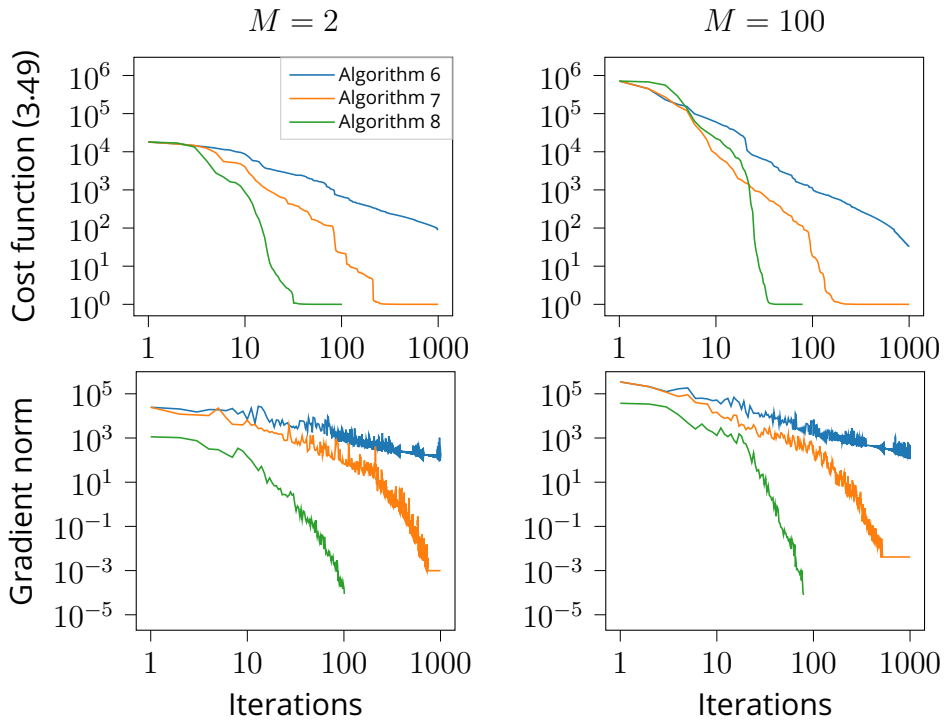


Figure 3.4: Cost function (3.49) and its gradient norm versus the iterations of Algorithms 6, 7 and 8. The dimensions of the parameter space are $p = 10$ and $n = 150$. Two different numbers of points M are considered: 2 in the left column and 100 in the right one. The cost functions are normalized so that their minimum value is 1.

($\beta = 0$) is studied with numerical experiments on simulated data. We do not measure the estimation errors made by Algorithms 6 and 7 since they minimize the same cost function as Algorithm 8 and return the same values of likelihood once the convergence reached. $n \in [\![20, 1000]\!]$ data $\boldsymbol{x}_i$ are sampled from the NC-MSG (3.24). The parameter $\theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\tau})$ of this distribution is generated as presented in the introduction of Subsection 3.9.1 with $\nu = 0.1$ in order to have heterogeneous textures τ_i. The considered estimators for this numerical experiment are the following:

- Gaussian estimators: the sample mean $\hat{\boldsymbol{\mu}}_{\text{SM}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i$ and the sample covariance matrix $\hat{\boldsymbol{\Sigma}}_{\text{SCM}} = \frac{1}{n}\sum_{i=1}^{n} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{\text{SM}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{\text{SM}})^T$.

- Tyler's joint location-covariance matrix estimator [136] denoted $\hat{\boldsymbol{\mu}}_{\text{Ty}}$ and $\hat{\boldsymbol{\Sigma}}_{\text{Ty}}$.

- Tyler's M-estimator with location known [136]. The sampled data $\boldsymbol{x}_i$ are centered with the true location $\boldsymbol{\mu}$ and then $\boldsymbol{\Sigma}$ is estimated. This estimator is denoted $\hat{\boldsymbol{\Sigma}}_{\text{Ty},\boldsymbol{\mu}}$.

- The proposed estimator denoted $\hat{\boldsymbol{\mu}}_{\text{IG}}$ and $\hat{\boldsymbol{\Sigma}}_{\text{IG}}$. Algorithm 8 minimizes the NLL (3.27).

The errors of estimation are measured with the Mean Squared Errors (MSE). These errors are computed as $\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_2^2$ and $\left\|\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}\right\|_2^2$ for the estimated location $\hat{\boldsymbol{\mu}}$ and the estimated covariance $\hat{\boldsymbol{\Sigma}}$ respectively. Then, they are averaged with 2000 Monte-Carlo on the samples $\boldsymbol{x}_i$. The MSE on the location and the covariance versus the number of samples $\boldsymbol{x}_i$ are plotted in Figure 3.5. First of all, we observe on both figures that the Gaussian estimators have a high MSE. This shows the interest of considering robust estimators such as the Tyler's joint location-covariance matrix estimator or the proposed one when the textures τ_i are heterogeneous. Then, the proposed estimators realize a much lower MSE than the Tyler's joint location-covariance estimator. We can note that when enough samples are provided, the MSE on the location realized by the proposed estimator reaches the machine precision and is therefore negligible. Finally, we compare the performance of the proposed estimator with the Tyler's M-estimator for the covariance estimation. Indeed, when the location is known, the Tyler's M-estimator is the MLE of the NC-MSG (3.24). We observe that when enough samples are provided, the proposed estimator matches the MSE of the Tyler's M-estimator. Overall, this experimental subsection illustrates the good performance of the proposed estimator when data are sampled from a NC-MSG (3.24).

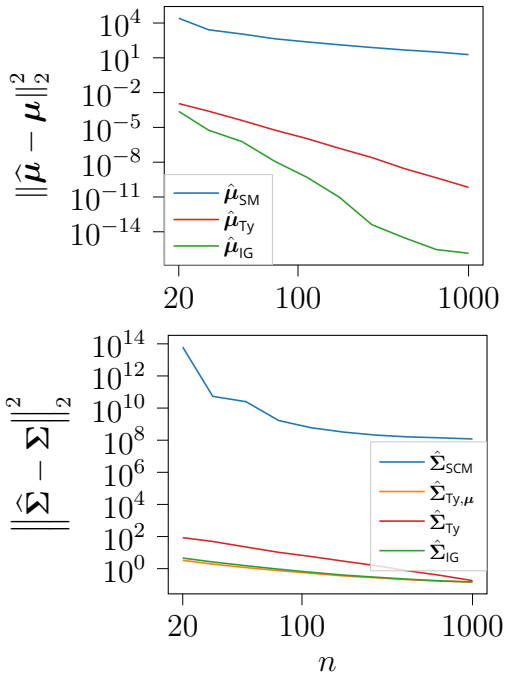


Figure 3.5: MSE over 2000 simulated sets $\{\boldsymbol{x}_i\}_{i=1}^n \subset \mathbb{R}^{10}$ versus the number samples $\boldsymbol{x}_i$ for the considered estimators $\hat{\boldsymbol{\mu}} \in \{\hat{\boldsymbol{\mu}}_{\mathsf{SM}}, \hat{\boldsymbol{\mu}}_{\mathsf{Ty}}, \hat{\boldsymbol{\mu}}_{\mathsf{IG}}\}$ and $\hat{\boldsymbol{\Sigma}} \in \{\hat{\boldsymbol{\Sigma}}_{\mathsf{SCM}}, \hat{\boldsymbol{\Sigma}}_{\mathsf{Ty},\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}_{\mathsf{Ty}}, \hat{\boldsymbol{\Sigma}}_{\mathsf{IG}}\}$. The proposed estimators $\hat{\boldsymbol{\mu}}_{\mathsf{IG}}$ and $\hat{\boldsymbol{\Sigma}}_{\mathsf{IG}}$ are computed as in (3.41) ($\beta = 0$) using Algorithm 8.

3.9.2 . Application

In the previous subsection, the different theoretical results derived in Sections from 3.4 to 3.8 showed several interests on synthetic data. We now focus on applying a *Nearest centroïd classifier* on $\mathcal{M}_{p,n}$ to real data using the estimation framework developed in Section 3.7, the divergence and the Riemannian center of mass from Section 3.8 as well as the optimization framework using the FIM from Section 3.6. This classifier is compared to several other *Nearest centroïd classifiers* associated with different estimators and divergences.

To do so, we consider the dataset *Breizhcrops* [118]: a large scale dataset of more than $600\,000$ crop time series from the Sentinel-2 satellite to classify. This dataset is presented in Chapter 1 Section 1.1. To classify these crops, we apply a *Nearest centroïd classifier* on descriptors. This classification algorithm works in three steps.

1. For each time series $\boldsymbol{X} \in \mathbb{R}^{n \times p}$, a descriptor is computed, *e.g.* a parameter $\theta \in \mathcal{M}_{p,n}$ from the minimization of the regularized NLL (3.44).

2. Then, on the training set, the center of mass of the descriptors of each

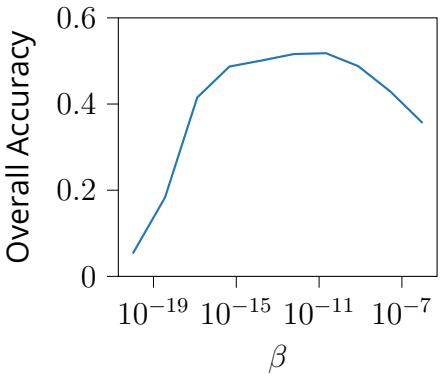


Figure 3.6: "Overall Accuracy" metric achieved by the proposed *Nearest centroïd classifier* on the *Breizhcrops* dataset versus the parameter of regularization β in (3.44). The chosen regularization is the L2 penalty from Table 3.1.

class is computed. This center of mass is always computed by minimizing the variance associated with a divergence between descriptors. For example, the center of mass on $\mathcal{M}_{p,n}$ is computed as in (3.49).

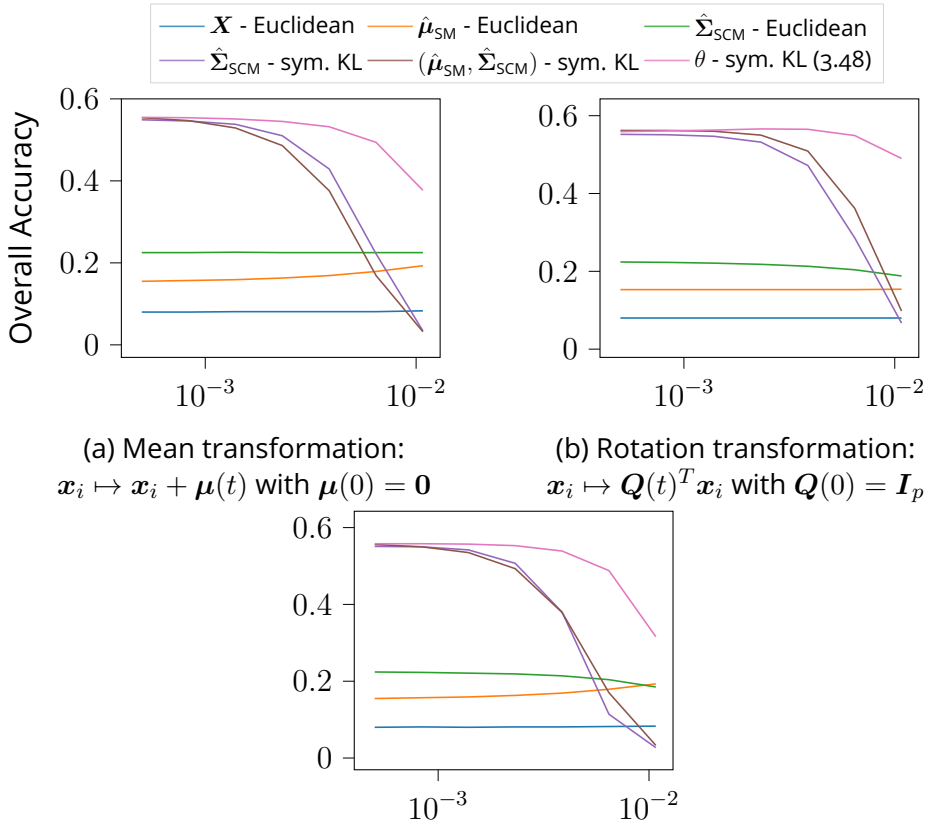
3. Finally, on the test set, each descriptor is labeled with the class of the nearest center of mass with respect to the chosen divergence.

Six *Nearest centroïd classifiers* are considered and they are grouped according to the divergence they use: the Euclidean distance, the symmetrized KL divergence between Gaussian distributions, or the symmetrized KL divergence (3.48) between NC-MSG. For each divergence, several *Nearest centroïd classifiers* are derived using several estimators. These estimators correspond to different assumptions on the data.

Three *Nearest centroïd classifiers* rely on the Euclidean distance between matrices (1.23). From this geometry, three *Nearest centroïd classifiers* are derived using three estimators: the batch itself $\boldsymbol{X}$, the sample mean $\hat{\boldsymbol{\mu}}_{\text{SM}}$ and $\hat{\boldsymbol{\Sigma}}_{\text{SCM}}^{\boldsymbol{\mu}=\mathbf{0}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^T$. The last two estimators correspond the assumption that data follow a Gaussian distribution (either with same covariance matrix for all batches or same location).

Two *Nearest centroïd classifiers* rely on the symmetrized KL divergence between Gaussian distributions. Let $\mathcal{M}_p = \mathbb{R}^p \times \mathcal{S}_p^{++}$. Given two pairs of parameters $\upsilon_1 = (\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \in \mathcal{M}_p$ and $\upsilon_2 = (\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \in \mathcal{M}_p$, this divergence is given by

$$\delta_{\mathcal{M}_p}(\upsilon_1, \upsilon_2) = \frac{1}{2}(\delta_{\text{KL}}(\upsilon_1, \upsilon_2) + \delta_{\text{KL}}(\upsilon_2, \upsilon_1)) \tag{3.50}$$



(a) Mean transformation:
$\boldsymbol{x}_i \mapsto \boldsymbol{x}_i + \boldsymbol{\mu}(t)$ with $\boldsymbol{\mu}(0) = \mathbf{0}$

(b) Rotation transformation:
$\boldsymbol{x}_i \mapsto \boldsymbol{Q}(t)^T \boldsymbol{x}_i$ with $\boldsymbol{Q}(0) = \boldsymbol{I}_p$

(c) Joint mean and rotation transformation:
$\boldsymbol{x}_i \mapsto \boldsymbol{Q}(t)^T \boldsymbol{x}_i + \boldsymbol{\mu}(t)$ with $\boldsymbol{\mu}(0) = \mathbf{0}$ and $\boldsymbol{Q}(0) = \boldsymbol{I}_p$

Figure 3.7: “Overall Accuracy" metric versus the parameter t associated with three transformations applied to the test set of the *Breizhcrops* dataset. The different *Nearest centroïd classifiers* estimate the centers of mass on the training data without transformations. Then, the classification is performed on the test set with three different transformations. For $t = 0$, the test set is not transformed, and the larger t is, the more the test set is transformed. Six different *Nearest centroïd classifiers* are compared: each one is a combination of an estimator, a divergence and its associated center of mass computation. The proposed one is denoted "θ - sym. KL". The latter uses Equations (3.45), (3.48) and (3.49) for the estimation, the divergence and the center of mass computation respectively. The regularization parameter β is fixed at 10^{-11} and the regularization is the L2 penalty from Table 3.1.

where

$$\delta_{\mathsf{KL}}(\upsilon_1, \upsilon_2) = \frac{1}{2}\left(\mathrm{Tr}\left(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1\right) + \Delta\boldsymbol{\mu}^T\boldsymbol{\Sigma}_2^{-1}\Delta\boldsymbol{\mu} + \log\left(\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|}\right) - p\right). \tag{3.51}$$

The center of mass of $\{\upsilon_i\}_{i=1}^M$ is the solution of

$$\underset{\upsilon \in \mathcal{M}_p}{\text{minimize}} \ \frac{1}{M}\sum_{i=1}^{M} \delta_{\mathcal{M}_p}(\upsilon, \upsilon_i). \tag{3.52}$$

Then, two *Nearest centroïd classifiers* are derived using two estimators: $\hat{\boldsymbol{\Sigma}}_{\mathsf{SCM}}^{\boldsymbol{\mu}=\mathbf{0}}$ (and thus $\boldsymbol{\mu}$ is assumed to be zero) and the MLE of the Gaussian distribution $(\hat{\boldsymbol{\mu}}_{\mathsf{SM}}, \hat{\boldsymbol{\Sigma}}_{\mathsf{SCM}})$.

Finally, the proposed *Nearest centroïd classifier* on $\mathcal{M}_{p,n}$ relies on the symmetrized KL divergence (3.48) between NC-MSGs. The center of mass is computed as explained in Section 3.8 and the estimation is described in Section 3.7 with the L2 penalty for the regularization.

The data are divided into two sets: a training set and a test set with $485\ 649$ and $122\ 614$ batches respectively. Among the six *Nearest centroïd classifiers*, only the one on $\mathcal{M}_{p,n}$ has a hyperparameter which the parameter β of the regularized NLL (3.44). Several values of β are tested on a small training set and a small validation set that both are subsets of the original training set. The performance is measured with the "Overall Accuracy" metric used in [118] and is plotted in Figure 3.6. The value of β with the highest "Overall Accuracy" metric is 10^{-11}. Hence, we use this value in the rest of the section. Then, we propose an experiment to illustrate Proposition 19 on the invariance of the estimation of textures under rigid transformations. Indeed, we train the six *Nearest centroïd classifiers* on a subset of the original training set and apply them on the full test set with a rigid transformation. Thus, the more a *Nearest centroïd classifier* is robust to these rigid transformations, the better the "Overall Accuracy" metric. Given $t \in [0, 1]$, three different rigid transformations are performed: transformation of the mean $\boldsymbol{x}_i \mapsto \boldsymbol{x}_i + \boldsymbol{\mu}(t)$ with $\boldsymbol{\mu}(t) = t\boldsymbol{a}$ for a given $\boldsymbol{a} \in \mathbb{R}^p$, rotation transformation $\boldsymbol{x}_i \mapsto \boldsymbol{Q}(t)^T\boldsymbol{x}_i$ with $\boldsymbol{Q}(t) = \exp(t\boldsymbol{\xi})$ for a given skew-symmetric $\boldsymbol{\xi} \in \mathbb{R}^{p\times p}$ (hence $\boldsymbol{Q}(t) \in \mathcal{O}_p$), and the joint mean and rotation transformation $\boldsymbol{x}_i \mapsto \boldsymbol{Q}(t)^T\boldsymbol{x}_i + \boldsymbol{\mu}(t)$. It should be noted that at $t = 0$, the data are left unchanged. The results are presented in Figure 3.7.

The conclusions of these experiments are fourfold. First, the proposed *Nearest centroïd classifier* is applicable to large scale datasets such as the *Breizhcrops* dataset. Second, the regularization proposed in Section 3.7 is important to get good classification performance. Indeed, we observe from Figure 3.6 that if β is too small then the "Overall Accuracy" metric becomes very low. Also, if β is too large then the "Overall Accuracy" metric becomes

also very low. Third, using KL divergences and their associated centers of mass to classify estimators give much better performance compared to the classical Euclidean distance. Indeed, even when data do not undergo rigid transformations, *Nearest centroïd classifiers* based on KL divergence outperform Euclidean *Nearest centroïd classifiers* in Figure 3.7. Fourth, considering NC-MSGs, as well as its KL divergence, instead of the Gaussian distribution is interesting to classify time series especially when rigid transformations are applied on the data. Indeed, in Figure 3.7, we observe a large improvement of performance when data are considered distributed from a NC-MSG and undergo rigid transformations.

3.10 . Conclusions

This chapter proposed novel statistical methods to handle non-centered data that are potentially non-Gaussian. We began with the information geometry of the non-centered multivariate Gaussian distribution and proposed two divergences. The latter that can be used in place of the Riemannian distance whose expression has no known closed form formula. An optimization algorithm has been developed to compute centers of mass, associated with the proposed divergences, of pairs location-covariance matrix. These divergences along with their centers of mass enabled us to implement a *Nearest centroïd classifier*. The latter has been applied on the *Breizhcrops* dataset and proved to be more robust than classifiers that rely only on the covariance matrix.

Then, we studied the statistical model of the NC-MSG. This model is well known when its location is assumed to be known but little work has been done when the location is unknown. In this study, we tackle the problem of the joint estimation of the location, the scatter matrix and the textures as well as their classification. To do so, two Riemannian manifolds and Riemannian optimization algorithms have been developed. The existence of a solution to the estimation problem is proven when a regularization is added to the NLL. Thus, this regularized NLL can be minimized using one the proposed Riemannian optimization algorithms to estimate the parameters. Once estimated, these parameters are classified with a *Nearest centroïd classifier* based on a KL divergence and its associated center of mass. The latter is also computed using one of the proposed Riemannian optimization algorithms. In particular, one of these algorithms is shown to be fast on both cost functions (the regularized NLL and the center of mass computation cost function). This allowed us to apply the proposed *Nearest centroïd classifier* on the dataset *Breizhcrops*. The proposed classifier is shown to be more robust than classifiers that rely on a Gaussian assumption.

3.A . Appendix

3.A.1 . Proof of Proposition 10: Riemannian gradient on $\mathcal{M}_p$

Using the definition of the gradient associated with the Euclidean metric [1, Ch. 3], we get $\forall \xi \in T_\upsilon \mathcal{M}_p$

$$\mathrm{D}\, h(\upsilon)[\xi] = \boldsymbol{G}_{\boldsymbol{\mu}}^T \boldsymbol{\xi}_{\boldsymbol{\mu}} + \mathrm{Tr}\left(\boldsymbol{G}_{\boldsymbol{\Sigma}}^T \boldsymbol{\xi}_{\boldsymbol{\Sigma}}\right) \tag{3.53}$$

$$= \left(\boldsymbol{\Sigma}\boldsymbol{G}_{\boldsymbol{\mu}}\right)^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\mu}} + \frac{1}{2}\,\mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1}\left(2\boldsymbol{\Sigma}\boldsymbol{G}_{\boldsymbol{\Sigma}}^T\boldsymbol{\Sigma}\right)\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\right) \tag{3.54}$$

$$= \left(\boldsymbol{\Sigma}\boldsymbol{G}_{\boldsymbol{\mu}}\right)^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\mu}} + \frac{1}{2}\,\mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1}\,\mathrm{sym}(2\boldsymbol{\Sigma}\boldsymbol{G}_{\boldsymbol{\Sigma}}^T\boldsymbol{\Sigma})\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\right) \tag{3.55}$$

$$= \langle P_\upsilon^{\mathcal{M}_p}\left(\boldsymbol{\Sigma}\boldsymbol{G}_{\boldsymbol{\mu}}, 2\boldsymbol{\Sigma}\boldsymbol{G}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}\right), \xi\rangle_\upsilon^{\mathcal{M}_p}. \tag{3.56}$$

Using the definition of the Riemannian gradient [1, Ch. 3] $\mathrm{D}\, h(\upsilon)[\xi] = \langle \mathrm{grad}_{\mathcal{M}_p} h(\upsilon), \xi\rangle_\upsilon^{\mathcal{M}_p}$, we get Proposition 10.

3.A.2 . Proof of Proposition 11: Second order retraction on $\mathcal{M}_p$

$\forall \upsilon \in \mathcal{M}_p$, $R_\upsilon^{\mathcal{M}_p}$ is a smooth mapping from $T_\upsilon\mathcal{M}_p$ onto $\mathcal{M}_p$. To be a second order retraction, it remains to check the three following properties [1, Ch. 4 and 5]: $\forall \xi \in T_\upsilon \mathcal{M}_p$

$$R_\upsilon^{\mathcal{M}_p}(0) = \upsilon, \mathrm{D}\, R_\upsilon^{\mathcal{M}_p}(0_\upsilon)[\xi] = \xi, \frac{\mathrm{D}^2}{dt^2} R_\upsilon^{\mathcal{M}_p}(t\xi)\Big|_{t=0} = 0 \tag{3.57}$$

where 0_υ denotes the zero element of $T_\upsilon\mathcal{M}_p$ and $\frac{\mathrm{D}^2}{dt^2}\gamma$ denotes the acceleration of the curve $t \mapsto \gamma(t)$ on $\mathcal{M}_p$ (see [1, Ch. 5]). The first two properties are easily verified. By denoting $R_\upsilon^{\mathcal{M}_p}(t\xi) = (\boldsymbol{\mu}(t), \boldsymbol{\Sigma}(t))$, and using Equation (3.10), the third property is equivalent to

$$\begin{cases} \ddot{\boldsymbol{\mu}}(0) - \dot{\boldsymbol{\Sigma}}(0)\boldsymbol{\Sigma}(0)^{-1}\dot{\boldsymbol{\mu}}(0) = \mathbf{0} \\ \ddot{\boldsymbol{\Sigma}}(0) + \dot{\boldsymbol{\mu}}(0)\dot{\boldsymbol{\mu}}(0)^T - \dot{\boldsymbol{\Sigma}}(0)\boldsymbol{\Sigma}(0)^{-1}\dot{\boldsymbol{\Sigma}}(0) = \mathbf{0}, \end{cases} \tag{3.58}$$

which is also verified.

3.A.3 . Proof of Proposition 12: Fisher information metric

First, we recall the definition of the FIM. See Chapter 2 for a more in-depth presentation. Let $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$ be n data points. Assuming that the underlying distribution admits a PDF, the corresponding NLL is denoted $\mathcal{L}$ and maps parameters θ, belonging to the parameter space $\mathcal{M}$, onto $\mathbb{R}$. By denoting $T_\theta\mathcal{M}$ the tangent space of $\mathcal{M}$ at $\theta \in \mathcal{M}$, and under conditions of regularity of $\mathcal{L}$, the FIM is defined $\forall \xi, \eta \in T_\theta\mathcal{M}$ as

$$\langle \xi, \eta\rangle_\theta^{\mathcal{M}} = \mathbb{E}[\mathrm{D}\,\mathcal{L}(\theta)[\xi]\,\mathrm{D}\,\mathcal{L}(\theta)[\eta]] = \mathbb{E}[\mathrm{D}^2\,\mathcal{L}(\theta)[\xi, \eta]]. \tag{3.59}$$

To derive the FIM of the NC-MSG given in Proposition 12, we recall classical formulas for the Gaussian distribution. To do so, we denote the set of its parameters (*i.e.* the set of locations and covariance matrices) as

$$\mathcal{M}_p = \mathbb{R}^p \times \mathcal{S}_p^{++}. \tag{3.60}$$

The NLL at $\upsilon = (\boldsymbol{\mu}, \boldsymbol{\Sigma}) \in \mathcal{M}_p$ and associated to one data point $\boldsymbol{x}$ is (neglecting terms not depending on υ)

$$\mathcal{L}^g_{\boldsymbol{x}}(\upsilon) = \frac{1}{2}\left[\log|\boldsymbol{\Sigma}| + (\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right] \tag{3.61}$$

Since $\mathcal{M}_p$ is an open set in the vector space $\mathbb{R}^p \times \mathcal{S}_p$, the tangent space of $\mathcal{M}_p$ at υ is

$$T_\upsilon\mathcal{M}_p = \mathbb{R}^p \times \mathcal{S}_p. \tag{3.62}$$

Finally, $\forall \xi = (\boldsymbol{\xi}_{\boldsymbol{\mu}}, \boldsymbol{\xi}_{\boldsymbol{\Sigma}}), \eta = (\boldsymbol{\eta}_{\boldsymbol{\mu}}, \boldsymbol{\eta}_{\boldsymbol{\Sigma}}) \in T_\upsilon\mathcal{M}_p$, the FIM of the Gaussian distribution associated to the NLL (3.61) is (see [120] for a derivation)

$$\langle \xi, \eta \rangle_\upsilon^{\mathcal{M}_p} = \boldsymbol{\xi}_{\boldsymbol{\mu}}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_{\boldsymbol{\mu}} + \frac{1}{2}\operatorname{Tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_{\boldsymbol{\Sigma}}). \tag{3.63}$$

Then, we derive the FIM associated to the NLL of the NC-MSG (3.27). We begin by writing (3.27) as a sum of Gaussian NLL (3.61). Indeed, $\forall \theta \in \mathcal{M}_{p,n}$, we have

$$\mathcal{L}(\theta|\{\boldsymbol{x}_i\}_{i=1}^n) = \sum_{i=1}^n (\mathcal{L}^g_{\boldsymbol{x}_i} \circ \varphi_i)(\theta), \tag{3.64}$$

where $\varphi_i(\theta) = (\boldsymbol{\mu}, \tau_i\boldsymbol{\Sigma})$. Thus, $\forall \theta \in \mathcal{M}_{p,n}$, $\forall \xi, \eta \in T_\theta\mathcal{M}_{p,n}$, and following the reasoning of [17, Proposition 6] and [18, Proposition 3.1], the FIM of the mixture of scaled Gaussian is expressed as a sum of FIM of the Gaussian distribution (3.63)

$$\begin{aligned}
\langle \xi, \eta \rangle_\theta^{\mathcal{M}_{p,n}^{\mathsf{FIM}}} &= \mathbb{E}\left[\mathrm{D}^2\,\mathcal{L}(\theta|\{\boldsymbol{x}_i\}_{i=1}^n)[\xi,\eta]\right] && (3.65)\\
&= \sum_{i=1}^n \mathbb{E}\left[\mathrm{D}^2(\mathcal{L}^g_{\boldsymbol{x}_i} \circ \varphi_i)(\theta)[\xi,\eta]\right] && (3.66)\\
&= \sum_{i=1}^n \mathbb{E}\left[\mathrm{D}(\mathcal{L}^g_{\boldsymbol{x}_i} \circ \varphi_i)(\theta)[\xi]\,\mathrm{D}(\mathcal{L}^g_{\boldsymbol{x}_i} \circ \varphi_i)(\theta)[\eta]\right] && (3.67)\\
&= \sum_{i=1}^n \mathbb{E}\left[\mathrm{D}(\mathcal{L}^g_{\boldsymbol{x}_i}(\varphi_i(\theta)))[\mathrm{D}\,\varphi_i(\theta)[\xi]]\,\mathrm{D}(\mathcal{L}^g_{\boldsymbol{x}_i}(\varphi_i(\theta)))[\mathrm{D}\,\varphi_i(\theta)[\eta]]\right] && (3.68)\\
&= \sum_{i=1}^n \langle \mathrm{D}\,\varphi_i(\theta)[\xi], \mathrm{D}\,\varphi_i(\theta)[\eta] \rangle_{\varphi_i(\theta)}^{\mathcal{M}_p}. && (3.69)
\end{aligned}$$

In the following, the i-th components of $\boldsymbol{\xi}_{\boldsymbol{\tau}}$ and $\boldsymbol{\eta}_{\boldsymbol{\tau}}$ are denoted ξ_i and η_i respectively. Therefore, the directional derivative of the function φ_i is

$$\mathrm{D}\,\varphi_i(\theta)[\xi] = (\boldsymbol{\xi}_{\boldsymbol{\mu}}, \xi_i\boldsymbol{\Sigma} + \tau_i\boldsymbol{\xi}_{\boldsymbol{\Sigma}}). \tag{3.70}$$

Thus, we get

$$\begin{aligned}
\langle \xi, \eta \rangle_\theta^{\mathcal{M}_{p,n}^{\mathsf{FIM}}} &= \sum_{i=1}^{n} \Bigg[\boldsymbol{\xi}_{\boldsymbol{\mu}}^T (\tau_i\boldsymbol{\Sigma})^{-1} \boldsymbol{\eta}_{\boldsymbol{\mu}} && (3.71)\\
&\qquad + \frac{1}{2} \operatorname{Tr}\left((\tau_i\boldsymbol{\Sigma})^{-1} (\xi_i\boldsymbol{\Sigma} + \tau_i\boldsymbol{\xi}_{\boldsymbol{\Sigma}}) (\tau_i\boldsymbol{\Sigma})^{-1} (\eta_i\boldsymbol{\Sigma} + \tau_i\boldsymbol{\eta}_{\boldsymbol{\Sigma}}) \right) \Bigg] \\
&= \sum_{i=1}^{n} \Bigg[\frac{1}{\tau_i} \boldsymbol{\xi}_{\boldsymbol{\mu}}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\eta}_{\boldsymbol{\mu}} + \frac{1}{2} p \frac{\xi_i \eta_i}{\tau_i^2} + \frac{1}{2} \frac{\xi_i}{\tau_i} \operatorname{Tr}(\boldsymbol{\Sigma}^{-1} \boldsymbol{\eta}_{\boldsymbol{\Sigma}}) && (3.72)\\
&\qquad + \frac{1}{2} \frac{\eta_i}{\tau_i} \operatorname{Tr}(\boldsymbol{\Sigma}^{-1} \boldsymbol{\xi}_{\boldsymbol{\Sigma}}) + \frac{1}{2} \operatorname{Tr}(\boldsymbol{\Sigma}^{-1} \boldsymbol{\xi}_{\boldsymbol{\Sigma}} \boldsymbol{\Sigma}^{-1} \boldsymbol{\eta}_{\boldsymbol{\Sigma}}) \Bigg] \\
&= \sum_{i=1}^{n} \left(\frac{1}{\tau_i} \right) \boldsymbol{\xi}_{\boldsymbol{\mu}}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\eta}_{\boldsymbol{\mu}} + \frac{n}{2} \operatorname{Tr}(\boldsymbol{\Sigma}^{-1} \boldsymbol{\xi}_{\boldsymbol{\Sigma}} \boldsymbol{\Sigma}^{-1} \boldsymbol{\eta}_{\boldsymbol{\Sigma}}) && (3.73)\\
&\qquad + \frac{p}{2} \left(\boldsymbol{\xi}_{\boldsymbol{\tau}} \odot \boldsymbol{\tau}^{-1} \right)^T \left(\boldsymbol{\eta}_{\boldsymbol{\tau}} \odot \boldsymbol{\tau}^{-1} \right) \\
&\qquad + \frac{1}{2} \boldsymbol{\xi}_{\boldsymbol{\tau}}^T \boldsymbol{\tau}^{\odot -1} \operatorname{Tr}(\boldsymbol{\Sigma}^{-1} \boldsymbol{\eta}_{\boldsymbol{\Sigma}}) + \frac{1}{2} \boldsymbol{\eta}_{\boldsymbol{\tau}}^T \boldsymbol{\tau}^{\odot -1} \operatorname{Tr}(\boldsymbol{\Sigma}^{-1} \boldsymbol{\xi}_{\boldsymbol{\Sigma}})
\end{aligned}$$

Since $\boldsymbol{\xi}_{\boldsymbol{\tau}}, \boldsymbol{\eta}_{\boldsymbol{\tau}} \in T_{\boldsymbol{\tau}}\mathcal{S}(\mathbb{R}_*^+)^n$, we have $\boldsymbol{\xi}_{\boldsymbol{\tau}}^T \boldsymbol{\tau}^{\odot -1} = \boldsymbol{\eta}_{\boldsymbol{\tau}}^T \boldsymbol{\tau}^{\odot -1} = 0$. Thus, the last two terms of (3.74) cancel and the expression of the FIM from Proposition 12 is obtained

$$\begin{aligned}
\langle \xi, \eta \rangle_\theta^{\mathcal{M}_{p,n}^{\mathsf{FIM}}} = \sum_{i=1}^{n} \left(\frac{1}{\tau_i} \right) \boldsymbol{\xi}_{\boldsymbol{\mu}}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\eta}_{\boldsymbol{\mu}} + \frac{n}{2} \operatorname{Tr}(\boldsymbol{\Sigma}^{-1} \boldsymbol{\xi}_{\boldsymbol{\Sigma}} \boldsymbol{\Sigma}^{-1} \boldsymbol{\eta}_{\boldsymbol{\Sigma}}) \\
+ \frac{p}{2} \left(\boldsymbol{\xi}_{\boldsymbol{\tau}} \odot \boldsymbol{\tau}^{-1} \right)^T \left(\boldsymbol{\eta}_{\boldsymbol{\tau}} \odot \boldsymbol{\tau}^{-1} \right). \qquad (3.74)
\end{aligned}$$

It should be noted that this formula defines an inner product on $\mathcal{E}_{p,n}$ if a transpose is added to $\boldsymbol{\xi}_{\boldsymbol{\Sigma}}$. Thus, $\langle ., . \rangle_{\cdot}^{\mathcal{M}_{p,n}^{\mathsf{FIM}}}$ is extended $\forall \xi, \eta \in \mathcal{E}_{p,n}$ as presented in Proposition 12.

3.A.4 . Proof of Proposition 13: Orthogonal projection on $\mathcal{M}_{p,n}^{\mathsf{FIM}}$

First of all, $\forall \theta \in \mathcal{M}_{p,n}$ the ambient space $\mathcal{E}_{p,n}$ defined in (3.31) is decomposed into two complementary subspaces

$$\mathcal{E}_{p,n} = T_\theta\mathcal{M}_{p,n} + T_\theta^\perp\mathcal{M}_{p,n} \tag{3.75}$$

where $T_\theta\mathcal{M}_{p,n}$ is the tangent space at θ defined in (3.32) and $T_\theta^\perp\mathcal{M}_{p,n}$ is the orthogonal complement

$$T_\theta^\perp\mathcal{M}_{p,n} = \left\{ \xi \in \mathcal{E}_{p,n} : \langle \xi, \eta \rangle_\theta^{\mathcal{M}_{p,n}^{\mathsf{FIM}}} = 0 \quad \forall \eta \in T_\theta\mathcal{M}_{p,n} \right\}. \tag{3.76}$$

It can be checked that this orthogonal complement is

$$T_\theta^\perp \mathcal{M}_{p,n} = \{\mathbf{0}\} \times \mathcal{A}_p \times \left\{ \alpha\boldsymbol{\tau} : \alpha \in \mathbb{R}_*^+ \right\} \tag{3.77}$$

where $\mathcal{A}_p$ is the set of $p \times p$ skew-symmetric matrices. Indeed, the elements of (3.77) verify Definition (3.76) and $\dim(\mathcal{E}_{p,n}) = \dim(T_\theta \mathcal{M}_{p,n}) + \dim(T_\theta^\perp \mathcal{M}_{p,n})$. Using the equations (3.75) and (3.77), the orthogonal projection of $\xi = (\boldsymbol{\xi_\mu}, \boldsymbol{\xi_\Sigma}, \boldsymbol{\xi_\tau}) \in \mathcal{E}_{p,n}$ onto $T_\theta \mathcal{M}_{p,n}$ is

$$P_\theta^{\mathcal{M}_{p,n}^{\mathsf{FIM}}}(\xi) = \left(\boldsymbol{\xi_\mu}, \boldsymbol{\xi_\Sigma} - \boldsymbol{A}, \boldsymbol{\xi_\tau} - \alpha\boldsymbol{\tau}\right) \tag{3.78}$$

where $\boldsymbol{A} \in \mathcal{A}_p$ and $\alpha \in \mathbb{R}_*^+$ have to be determined. Furthermore, $\forall \eta = (\mathbf{0}, \boldsymbol{\eta_\Sigma}, \beta\boldsymbol{\tau}) \in T_\theta^\perp \mathcal{M}_{p,n}$ with $\beta \in \mathbb{R}_*^+$, we must have

$$\langle P_\theta^{\mathcal{M}_{p,n}^{\mathsf{FIM}}}(\xi), \eta \rangle_\theta^{\mathcal{M}_{p,n}^{\mathsf{FIM}}} = 0. \tag{3.79}$$

This induces that

$$\begin{cases} \boldsymbol{\xi_\Sigma} - \boldsymbol{A} = \mathrm{sym}(\boldsymbol{\xi_\Sigma}) \\ \alpha = \frac{\boldsymbol{\xi_\tau}^T \boldsymbol{\tau}^{\odot -1}}{n} \end{cases} \tag{3.80}$$

where $\mathrm{sym}(\boldsymbol{\xi_\Sigma}) = \frac{1}{2}(\boldsymbol{\xi_\Sigma} + \boldsymbol{\xi_\Sigma}^T)$. Thus the orthogonal projection from $\mathcal{E}_{p,n}$ onto $T_\theta \mathcal{M}_{p,n}$ is

$$P_\theta^{\mathcal{M}_{p,n}^{\mathsf{FIM}}}(\xi) = \left(\boldsymbol{\xi_\mu}, \mathrm{sym}(\boldsymbol{\xi_\Sigma}), \boldsymbol{\xi_\tau} - \frac{\boldsymbol{\xi_\tau}^T \boldsymbol{\tau}^{\odot -1}}{n} \boldsymbol{\tau}\right). \tag{3.81}$$

3.A.5 . Proof of Proposition 14: Levi-Civita connection on $\mathcal{M}_{p,n}^{\mathsf{FIM}}$

First of all, the FIM defined in Proposition 12 is rewritten with a function g. Indeed, let $\theta \in \mathcal{M}_{p,n}$ and $\xi, \eta \in T_\theta \mathcal{M}$, the function g is defined as

$$g_\theta(\xi, \eta) = \langle \xi, \eta \rangle_\theta^{\mathcal{M}_{p,n}^{\mathsf{FIM}}}. \tag{3.82}$$

This function g is of primary importance for the development of the Levi Civita connection.

We briefly introduce the Levi-Civita connection. The general theory of it can be found in [1, Ch. 5]. The Levi-Civita connection, simply denoted $\nabla^{\mathcal{M}_{p,n}^{\mathsf{FIM}}} : (\xi, \eta) \mapsto \nabla_\xi^{\mathcal{M}_{p,n}^{\mathsf{FIM}}} \eta$, is characterized by the Koszul formula. Let $\nu \in T_\theta \mathcal{M}_{p,n}$, in our case the Koszul formula writes

$$\begin{aligned} g_\theta(\nabla_\xi^{\mathcal{M}_{p,n}^{\mathsf{FIM}}} \eta, \nu) - g_\theta(\mathrm{D}\,\eta[\xi], \nu) = \\ \frac{1}{2}\left(\mathrm{D}\,g_\theta[\xi](\eta, \nu) + \mathrm{D}\,g_\theta[\eta](\xi, \nu) - \mathrm{D}\,g_\theta[\nu](\xi, \eta)\right) \end{aligned} \tag{3.83}$$

where $\mathrm{D}\, g_\theta[\nu](\xi,\eta)$ is the directional derivative of the function $g_\cdot(\xi,\eta) : \theta \mapsto g_\theta(\xi,\eta)$. We begin by computing $\mathrm{D}\, g_\theta[\nu](\xi,\eta)$:

$$
\begin{aligned}
-\mathrm{D}\, g_\theta[\nu](\xi,\eta) = &\sum_{i=1}^{n}\left(\frac{\nu_i}{\tau_i^2}\right)\boldsymbol{\xi}_{\boldsymbol{\mu}}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_{\boldsymbol{\mu}} + \sum_{i=1}^{n}\left(\frac{1}{\tau_i}\right)\boldsymbol{\xi}_{\boldsymbol{\mu}}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\nu}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_{\boldsymbol{\mu}} \\
&+ n\,\mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1}\,\mathrm{sym}(\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_{\boldsymbol{\Sigma}})\boldsymbol{\Sigma}^{-1}\boldsymbol{\nu}_{\boldsymbol{\Sigma}}\right) \\
&+ p\left(\boldsymbol{\xi}_{\boldsymbol{\tau}}\odot\boldsymbol{\eta}_{\boldsymbol{\tau}}\odot\boldsymbol{\tau}^{\odot-2}\right)^T\left(\boldsymbol{\nu}_{\boldsymbol{\tau}}\odot\boldsymbol{\tau}^{\odot-1}\right).
\end{aligned}
\tag{3.84}
$$

Since the objective is to identify $\nabla_\xi^{\mathcal{M}_{p,n}^{\mathsf{FIM}}}\eta$ using (3.83) and the FIM from Proposition 12, (3.84) needs to be rewritten. To do so, the following two terms are rewritten

$$
\sum_{i=1}^{n}\left(\frac{\nu_i}{\tau_i^2}\right)\boldsymbol{\xi}_{\boldsymbol{\mu}}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_{\boldsymbol{\mu}} = p\left(\frac{1}{p}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_{\boldsymbol{\mu}}\mathbf{1}_n\odot\boldsymbol{\tau}^{\odot-1}\right)^T\left(\boldsymbol{\nu}\odot\boldsymbol{\tau}^{\odot-1}\right), \tag{3.85}
$$

and, since $\boldsymbol{\nu}_{\boldsymbol{\Sigma}}\in\mathcal{S}_p$

$$
\sum_{i=1}^{n}\left(\frac{1}{\tau_i}\right)\boldsymbol{\xi}_{\boldsymbol{\mu}}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\nu}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_{\boldsymbol{\mu}} = \sum_{i=1}^{n}\left(\frac{1}{\tau_i}\right)\mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1}\,\mathrm{sym}(\boldsymbol{\eta}_{\boldsymbol{\mu}}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T)\boldsymbol{\Sigma}^{-1}\boldsymbol{\nu}_{\boldsymbol{\Sigma}}\right). \tag{3.86}
$$

Hence, we get that

$$
\begin{aligned}
-\mathrm{D}\, g_\theta[\nu](\xi,\eta) = &\sum_{i=1}^{n}\left(\frac{1}{\tau_i}\right)\mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1}\,\mathrm{sym}(\boldsymbol{\eta}_{\boldsymbol{\mu}}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T)\boldsymbol{\Sigma}^{-1}\boldsymbol{\nu}_{\boldsymbol{\Sigma}}\right) \qquad (3.87)\\
&+ n\,\mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\,\mathrm{sym}(\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_{\boldsymbol{\Sigma}})\boldsymbol{\Sigma}^{-1}\boldsymbol{\nu}_{\boldsymbol{\Sigma}}) \\
&+ p\left(\frac{1}{p}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_{\boldsymbol{\mu}}\mathbf{1}_n\odot\boldsymbol{\tau}^{\odot-1}\right)^T\left(\boldsymbol{\nu}_{\boldsymbol{\tau}}\odot\boldsymbol{\tau}^{\odot-1}\right) \\
&+ p\left(\boldsymbol{\xi}_{\boldsymbol{\tau}}\odot\boldsymbol{\eta}_{\boldsymbol{\tau}}\odot\boldsymbol{\tau}^{\odot-2}\right)^T\left(\boldsymbol{\nu}_{\boldsymbol{\tau}}\odot\boldsymbol{\tau}^{\odot-1}\right) \\
= &\, n\,\mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1}\left[\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{\tau_i}\right)\mathrm{sym}(\boldsymbol{\eta}_{\boldsymbol{\mu}}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T)\right.\right. \qquad (3.88)\\
&\qquad\left.\left.+\,\mathrm{sym}(\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_{\boldsymbol{\Sigma}})\right]\boldsymbol{\Sigma}^{-1}\boldsymbol{\nu}_{\boldsymbol{\Sigma}}\right) \\
&+ p\left(\left[\frac{1}{p}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_{\boldsymbol{\mu}}\mathbf{1}_n\right.\right. \\
&\qquad\left.\left.+\,\boldsymbol{\xi}_{\boldsymbol{\tau}}\odot\boldsymbol{\eta}_{\boldsymbol{\tau}}\odot\boldsymbol{\tau}^{\odot-1}\right]\odot\boldsymbol{\tau}^{\odot-1}\right)^T\left(\boldsymbol{\nu}_{\boldsymbol{\tau}}\odot\boldsymbol{\tau}^{\odot-1}\right).
\end{aligned}
$$

We then compute $\mathrm{D}\, g_\theta[\xi](\eta,\nu)$:

$$\begin{aligned}\mathrm{D}\, g_\theta[\xi](\eta,\nu) = &-\sum_{i=1}^n \left(\frac{\xi_i}{\tau_i^2}\right)\boldsymbol{\eta}_{\boldsymbol{\mu}}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\nu}_{\boldsymbol{\mu}} - \sum_{i=1}^n \left(\frac{1}{\tau_i}\right)\boldsymbol{\eta}_{\boldsymbol{\mu}}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\nu}_{\boldsymbol{\mu}} \\ &- n\,\mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\nu}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\Sigma}}) \\ &- p\left(\boldsymbol{\eta}_{\boldsymbol{\tau}}\odot\boldsymbol{\nu}_{\boldsymbol{\tau}}\odot\boldsymbol{\tau}^{\odot-2}\right)^T\left(\boldsymbol{\xi}_{\boldsymbol{\tau}}\odot\boldsymbol{\tau}^{\odot-1}\right)\end{aligned} \tag{3.89}$$

$$\begin{aligned}= &-\sum_{i=1}^n \left(\frac{1}{\tau_i}\right)\boldsymbol{\eta}_{\boldsymbol{\mu}}^T\left(\frac{\boldsymbol{\xi}_{\boldsymbol{\tau}}^T\boldsymbol{\tau}^{\odot-2}}{\sum_{i=1}^n \frac{1}{\tau_i}}\boldsymbol{I}_p + \boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\right)\boldsymbol{\Sigma}^{-1}\boldsymbol{\nu}_{\boldsymbol{\mu}} \\ &- n\,\mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\,\mathrm{sym}(\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_{\boldsymbol{\Sigma}})\boldsymbol{\Sigma}^{-1}\boldsymbol{\nu}_{\boldsymbol{\Sigma}}) \\ &- p\left(\boldsymbol{\xi}_{\boldsymbol{\tau}}\odot\boldsymbol{\eta}_{\boldsymbol{\tau}}\odot\boldsymbol{\tau}^{\odot-2}\right)^T\left(\boldsymbol{\nu}_{\boldsymbol{\tau}}\odot\boldsymbol{\tau}^{\odot-1}\right).\end{aligned} \tag{3.90}$$

Using (3.84) and (3.89), we can calculate the right-hand side of the Koszul formula (3.83),

$$\begin{aligned}&\frac{1}{2}\left(\mathrm{D}\, g_\theta[\xi](\eta,\nu) + \mathrm{D}\, g_\theta[\eta](\xi,\nu) - \mathrm{D}\, g_\theta[\nu](\xi,\eta)\right) \\ &= \sum_{i=1}^n \left(\frac{1}{\tau_i}\right)\Bigg[-\frac{1}{2}\Bigg[\boldsymbol{\eta}_{\boldsymbol{\mu}}^T\left(\frac{\boldsymbol{\xi}_{\boldsymbol{\tau}}^T\boldsymbol{\tau}^{\odot-2}}{\sum_{i=1}^n \frac{1}{\tau_i}}\boldsymbol{I}_p + \boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\right) \\ &+ \boldsymbol{\xi}_{\boldsymbol{\mu}}^T\left(\frac{\boldsymbol{\eta}_{\boldsymbol{\tau}}^T\boldsymbol{\tau}^{\odot-2}}{\sum_{i=1}^n \frac{1}{\tau_i}}\boldsymbol{I}_p + \boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_{\boldsymbol{\Sigma}}\right)\Bigg]\Bigg]\boldsymbol{\Sigma}^{-1}\boldsymbol{\nu}_{\boldsymbol{\mu}} \\ &+ \frac{n}{2}\,\mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1}\left[\frac{1}{n}\sum_{i=1}^n\left(\frac{1}{\tau_i}\right)\mathrm{sym}(\boldsymbol{\eta}_{\boldsymbol{\mu}}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T) - \mathrm{sym}(\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_{\boldsymbol{\Sigma}})\right]\boldsymbol{\Sigma}^{-1}\boldsymbol{\nu}_{\boldsymbol{\Sigma}}\right) \\ &+ \frac{p}{2}\left(\left[\frac{1}{p}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_{\boldsymbol{\mu}}\mathbf{1}_n - \boldsymbol{\xi}_{\boldsymbol{\tau}}\odot\boldsymbol{\eta}_{\boldsymbol{\tau}}\odot\boldsymbol{\tau}^{\odot-1}\right]\odot\boldsymbol{\tau}^{\odot-1}\right)^T\left(\boldsymbol{\nu}_{\boldsymbol{\tau}}\odot\boldsymbol{\tau}^{\odot-1}\right).\end{aligned} \tag{3.91}$$

Using formulas (3.83) and (3.91) and by identification, we get that the Levi Civita connection is

$$\nabla_\xi^{\mathcal{M}_{p,n}^{\mathsf{FIM}}}\eta = P_\theta^{\mathcal{M}_{p,n}^{\mathsf{FIM}}}(\overline{\nabla}_\xi^{\mathcal{M}_{p,n}^{\mathsf{FIM}}}\eta) \tag{3.92}$$

where

$$\overline{\nabla}_{\xi}^{\mathcal{M}_{p,n}^{\text{FIM}}}\eta = \mathrm{D}\,\eta[\xi] + \Bigg(-\frac{1}{2}\Bigg[\left(\frac{\boldsymbol{\xi}_{\boldsymbol{\tau}}^T\boldsymbol{\tau}^{\odot -2}}{\sum_{i=1}^n \frac{1}{\tau_i}}\boldsymbol{I}_p + \boldsymbol{\xi}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\right)\boldsymbol{\eta}_{\boldsymbol{\mu}} + \left(\frac{\boldsymbol{\eta}_{\boldsymbol{\tau}}^T\boldsymbol{\tau}^{\odot -2}}{\sum_{i=1}^n \frac{1}{\tau_i}}\boldsymbol{I}_p + \boldsymbol{\eta}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\right)\boldsymbol{\xi}_{\boldsymbol{\mu}}\Bigg], \frac{1}{n}\sum_{i=1}^n \left(\frac{1}{\tau_i}\right)\boldsymbol{\eta}_{\boldsymbol{\mu}}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T - \boldsymbol{\xi}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_{\boldsymbol{\Sigma}}, \frac{1}{p}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_{\boldsymbol{\mu}}\mathbf{1}_n - \boldsymbol{\xi}_{\boldsymbol{\tau}} \odot \boldsymbol{\eta}_{\boldsymbol{\tau}} \odot \boldsymbol{\tau}^{\odot -1}\Bigg).$$

3.A.6 . Proof of Proposition 15: Riemannian gradient on $\mathcal{M}_{p,n}^{\text{FIM}}$

Let $h : \mathcal{M}_{p,n} \to \mathbb{R}$ be a smooth function and θ be a point in $\mathcal{M}_{p,n}$. We present the correspondence between the Euclidean gradient of h (which can be computed using automatic differentiation libraries such as Autograd [79] and JAX [25]) and the Riemannian gradient associated with the FIM defined in Proposition 12. The Euclidean gradient $\operatorname{grad} h(\theta) = (\boldsymbol{G}_{\boldsymbol{\mu}}, \boldsymbol{G}_{\boldsymbol{\Sigma}}, \boldsymbol{G}_{\boldsymbol{\tau}})$ of h at $\theta \in \mathcal{M}_{p,n}$ is defined as the unique element in $\mathbb{R}^p \times \mathbb{R}^{p\times p} \times \mathbb{R}^n$ such that $\forall \xi \in \mathbb{R}^p \times \mathbb{R}^{p\times p} \times \mathbb{R}^n$

$$\mathrm{D}\,h(\theta)[\xi] = \langle \operatorname{grad} h(\theta), \xi\rangle_\theta = \boldsymbol{G}_{\boldsymbol{\mu}}^T\boldsymbol{\xi}_{\boldsymbol{\mu}} + \operatorname{Tr}\left(\boldsymbol{G}_{\boldsymbol{\Sigma}}^T\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\right) + \boldsymbol{G}_{\boldsymbol{\tau}}^T\boldsymbol{\xi}_{\boldsymbol{\tau}}. \tag{3.93}$$

Then, the Riemannian gradient $\operatorname{grad}_{\mathcal{M}_{p,n}^{\text{FIM}}} h(\theta) = (\boldsymbol{G}_{\boldsymbol{\mu}}^{\mathcal{M}_{p,n}^{\text{FIM}}}, \boldsymbol{G}_{\boldsymbol{\Sigma}}^{\mathcal{M}_{p,n}^{\text{FIM}}}, \boldsymbol{G}_{\boldsymbol{\tau}}^{\mathcal{M}_{p,n}^{\text{FIM}}})$ is defined as the unique element in $T_\theta\mathcal{M}_{p,n}$ such that $\forall \xi \in T_\theta\mathcal{M}_{p,n}$

$$\mathrm{D}\,h(\theta)[\xi] = \langle \operatorname{grad}_{\mathcal{M}_{p,n}^{\text{FIM}}} h(\theta), \xi\rangle_\theta^{\mathcal{M}_{p,n}^{\text{FIM}}}. \tag{3.94}$$

Hence, $\forall \xi \in T_\theta \mathcal{M}_{p,n}$, we get that

$$\mathrm{D}\, h(\theta)[\xi] = \boldsymbol{G}_{\boldsymbol{\mu}}^T \boldsymbol{\xi}_{\boldsymbol{\mu}} + \operatorname{Tr}\left(\boldsymbol{G}_{\boldsymbol{\Sigma}}^T \boldsymbol{\xi}_{\boldsymbol{\Sigma}}\right) + \boldsymbol{G}_{\boldsymbol{\tau}}^T \boldsymbol{\xi}_{\boldsymbol{\tau}} \tag{3.95}$$

$$= \left(\sum_{i=1}^n \frac{1}{\tau_i}\right) \left(\left(\sum_{i=1}^n \frac{1}{\tau_i}\right)^{-1} \boldsymbol{\Sigma} \boldsymbol{G}_{\boldsymbol{\mu}}\right)^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\xi}_{\boldsymbol{\mu}} \tag{3.96}$$

$$+ \frac{n}{2} \operatorname{Tr}\left(\boldsymbol{\Sigma}^{-1} \left(\frac{2}{n} \boldsymbol{\Sigma} \boldsymbol{G}_{\boldsymbol{\Sigma}} \boldsymbol{\Sigma}\right)^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\xi}_{\boldsymbol{\Sigma}}\right)$$

$$+ \frac{p}{2} \left(\boldsymbol{\tau}^{\odot -1} \odot \left(\frac{2}{p} \boldsymbol{\tau}^{\odot 2} \odot \boldsymbol{G}_{\boldsymbol{\tau}}\right)\right)^T \left(\boldsymbol{\tau}^{\odot -1} \odot \boldsymbol{\xi}_{\boldsymbol{\tau}}\right)$$

$$= \left(\sum_{i=1}^n \frac{1}{\tau_i}\right) \boldsymbol{\eta}_{\boldsymbol{\mu}}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\xi}_{\boldsymbol{\mu}} + \frac{n}{2} \operatorname{Tr}\left(\boldsymbol{\Sigma}^{-1} \boldsymbol{\eta}_{\boldsymbol{\Sigma}}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\xi}_{\boldsymbol{\Sigma}}\right) \tag{3.97}$$

$$+ \frac{p}{2} \left(\boldsymbol{\tau}^{\odot -1} \odot \boldsymbol{\eta}_{\boldsymbol{\tau}}\right)^T \left(\boldsymbol{\tau}^{\odot -1} \odot \boldsymbol{\xi}_{\boldsymbol{\tau}}\right)$$

where $\eta = \left(\boldsymbol{\eta}_{\boldsymbol{\mu}}, \boldsymbol{\eta}_{\boldsymbol{\Sigma}}, \boldsymbol{\eta}_{\boldsymbol{\tau}}\right) = \left(\left(\sum_{i=1}^n \frac{1}{\tau_i}\right)^{-1} \boldsymbol{\Sigma} \boldsymbol{G}_{\boldsymbol{\mu}}, \frac{2}{n} \boldsymbol{\Sigma} \boldsymbol{G}_{\boldsymbol{\Sigma}} \boldsymbol{\Sigma}, \frac{2}{p} \boldsymbol{\tau}^{\odot 2} \odot \boldsymbol{G}_{\boldsymbol{\tau}}\right)$. To get the Riemannian gradient, it remains to project η into the tangent space $T_\theta \mathcal{M}_{p,n}$ using the orthogonal projection $P_\theta^{\mathcal{M}_{p,n}^{\mathsf{FIM}}}$. Thus, we get that

$$\operatorname{grad}_{\mathcal{M}_{p,n}^{\mathsf{FIM}}} h(\theta) = P_\theta^{\mathcal{M}_{p,n}^{\mathsf{FIM}}}(\eta), \tag{3.98}$$

which is exactly the Riemannian gradient defined in Proposition 15.

3.A.7 . Proof of Proposition 16: Second order retraction on $\mathcal{M}_{p,n}^{\mathsf{FIM}}$

Let $\theta \in \mathcal{M}_{p,n}$, $\xi \in T_\theta \mathcal{M}_{p,n}$ and $t \in [0, t_{max}[$ where t_{max} is to be defined. We denote $r(t) = R(t\xi)$ where R is defined in Proposition 16, *i.e.*

$$r(t) = \left(\boldsymbol{\mu} + t\boldsymbol{\xi}_{\boldsymbol{\mu}} + \frac{t^2}{2} \left[\frac{\boldsymbol{\xi}_{\boldsymbol{\tau}}^T \boldsymbol{\tau}^{\odot -2}}{\sum_{i=1}^n \frac{1}{\tau_i}} \boldsymbol{I}_p + \boldsymbol{\xi}_{\boldsymbol{\Sigma}} \boldsymbol{\Sigma}^{-1}\right] \boldsymbol{\xi}_{\boldsymbol{\mu}}, \right. \tag{3.99}$$

$$\boldsymbol{\Sigma} + t\boldsymbol{\xi}_{\boldsymbol{\Sigma}} + \frac{t^2}{2} \left(\boldsymbol{\xi}_{\boldsymbol{\Sigma}} \boldsymbol{\Sigma}^{-1} \boldsymbol{\xi}_{\boldsymbol{\Sigma}} - \frac{1}{n} \sum_{i=1}^n \left(\frac{1}{\tau_i}\right) \boldsymbol{\xi}_{\boldsymbol{\mu}} \boldsymbol{\xi}_{\boldsymbol{\mu}}^T\right), \tag{3.100}$$

$$\left. N\left(\boldsymbol{\tau} + t\boldsymbol{\xi}_{\boldsymbol{\tau}} + \frac{t^2}{2} \left(\boldsymbol{\xi}_{\boldsymbol{\tau}}^{\odot 2} \odot \boldsymbol{\tau}^{\odot -1} - \frac{1}{p} \boldsymbol{\xi}_{\boldsymbol{\mu}}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\xi}_{\boldsymbol{\mu}} \mathbf{1}_n\right)\right)\right), \tag{3.101}$$

where $\forall \boldsymbol{x} \in (\mathbb{R}_*^+)^n$, N is defined as $N(\boldsymbol{x}) = \left(\prod_{i=1}^n x_i\right)^{-1/n} \boldsymbol{x}$.

The objective is to prove that r is a second order retraction on $\mathcal{M}_{p,n}^{\mathsf{FIM}}$. The different properties of the definition of a second order retraction are verified in the following; see [1, Ch. 4 and 5] for a complete definition.

First of all, we define t_{max} such that r is a valid retraction. Indeed, r must respect some constraints of positivity,

$$\boldsymbol{\Sigma} + t\boldsymbol{\xi}_{\boldsymbol{\Sigma}} + \frac{t^2}{2}\left(\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\Sigma}} - \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{\tau_i}\right)\boldsymbol{\xi}_{\boldsymbol{\mu}}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T\right) \succ \mathbf{0}, \tag{3.102}$$

$$\boldsymbol{\tau} + t\boldsymbol{\xi}_{\boldsymbol{\tau}} + \frac{t^2}{2}\left(\boldsymbol{\xi}_{\boldsymbol{\tau}}^{\odot 2} \odot \boldsymbol{\tau}^{\odot -1} - \frac{1}{p}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\mu}}\mathbf{1}_n\right) > \mathbf{0}, \tag{3.103}$$

where for $\boldsymbol{A} \in \mathcal{S}_p$, $\boldsymbol{A} \succ \mathbf{0}$ means $\boldsymbol{A}$ is positive definite and for $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{x} > \mathbf{0}$ means the components of $\boldsymbol{x}$ are strictly positive. Of course, (3.102) and (3.103) are not necessarily respected depending on the value of t. To define the value of t_{max} such that (3.102) and (3.103) are respected, we begin by studying the eigenvalues of the left side of (3.102). To do so, let $\lambda^-(\boldsymbol{A})$ be the smallest eigenvalue of $\boldsymbol{A}$ and $\boldsymbol{\Sigma}(t)$ be the left side of (3.102). Thus, we get that

$$\lambda^-\left(\boldsymbol{\Sigma}(t)\right) \geq \lambda^-\left(\boldsymbol{\Sigma}\right) + t\lambda^-\left(\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\right) + \frac{t^2}{2}\left[\lambda^-\left(\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\right) - \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{\tau_i}\right)\left\|\boldsymbol{\xi}_{\boldsymbol{\mu}}\right\|_2^2\right] \tag{3.104}$$

$$\geq \lambda^-\left(\boldsymbol{\Sigma}\right) + t\lambda^-\left(\boldsymbol{\xi}_{\boldsymbol{\Sigma}}\right) - \frac{t^2}{2n}\sum_{i=1}^{n}\left(\frac{1}{\tau_i}\right)\left\|\boldsymbol{\xi}_{\boldsymbol{\mu}}\right\|_2^2. \tag{3.105}$$

A sufficient condition to satisfy (3.102) is that the right side of (3.105) is strictly positive. This is achieved whenever t is in $[0, t_1[$ where t_1 is defined as followed

- if $\boldsymbol{\xi}_{\boldsymbol{\mu}} \neq \mathbf{0}$, $t_1 = \frac{\sqrt{\Delta_1} - \lambda^-(\boldsymbol{\xi}_{\boldsymbol{\Sigma}})}{2\lambda^-(\boldsymbol{\Sigma})}$ and $\Delta_1 = \lambda^-(\boldsymbol{\xi}_{\boldsymbol{\Sigma}})^2 + \frac{2}{n}\lambda^-(\boldsymbol{\Sigma})\sum_{i=1}^{n}\left(\frac{1}{\tau_i}\right)\left\|\boldsymbol{\xi}_{\boldsymbol{\mu}}\right\|_2^2$,
- if $\boldsymbol{\xi}_{\boldsymbol{\mu}} = \mathbf{0}$, $t_1 = \frac{\lambda^-(\boldsymbol{\Sigma})}{|\lambda^-(\boldsymbol{\xi}_{\boldsymbol{\Sigma}})|}$ for $\lambda^-(\boldsymbol{\xi}_{\boldsymbol{\Sigma}}) < 0$, $t_1 = +\infty$ otherwise.

Lets denote the minimum value of $\boldsymbol{x} \in \mathbb{R}^n$ by $(\boldsymbol{x})_{min}$. Using the same reasoning as before, one can show that whenever t is in $[0, t_2[$, where t_2 is defined in the following, (3.103) is satisfied.

- If $\boldsymbol{\xi}_{\boldsymbol{\mu}} \neq \mathbf{0}$, $t_2 = \frac{\sqrt{\Delta_2} - (\boldsymbol{\xi}_{\boldsymbol{\tau}})_{min}}{2(\boldsymbol{\tau})_{min}}$ and $\Delta_2 = (\boldsymbol{\xi}_{\boldsymbol{\tau}})_{min}^2 + \frac{2}{p}(\boldsymbol{\tau})_{min}\left\|\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\xi}_{\boldsymbol{\mu}}\right\|_2^2$.
- If $\boldsymbol{\xi}_{\boldsymbol{\mu}} = \mathbf{0}$, $t_2 = \frac{(\boldsymbol{\tau})_{min}}{|(\boldsymbol{\xi}_{\boldsymbol{\tau}})_{min}|}$ for $(\boldsymbol{\xi}_{\boldsymbol{\tau}})_{min} < 0$, $t_2 = +\infty$ otherwise.

Hence, we get $t_{max} = \min\{t_1, t_2\} > 0$ such that $\forall t \in [0, t_{max}[$, $r(t) \in \mathcal{M}_{p,n}$.

Then, to be a second order retraction, it remains to check that the three following properties are respected,

$$\begin{cases} r(0) = \theta, \\ \dot{r}(0) = \xi, \\ \nabla_{\dot{r}}^{\mathcal{M}_{p,n}^{\mathsf{FIM}}} \dot{r}\Big|_{t=0} = 0, \end{cases} \tag{3.106}$$

where $\dot{r}(t) = \frac{d}{dt}r(t)$ and $\nabla^{\mathcal{M}_{p,n}^{\mathsf{FIM}}}$ is the Levi-Civita connection defined in Proposition 14. The first property is easily verified. In the rest of the proof, the following notations are used: $r(t) = (\boldsymbol{\mu}(t), \boldsymbol{\Sigma}(t), \boldsymbol{\tau}(t))$, $\dot{r}(t) = (\dot{\boldsymbol{\mu}}(t), \dot{\boldsymbol{\Sigma}}(t), \dot{\boldsymbol{\tau}}(t))$ and $\ddot{r}(t) = (\ddot{\boldsymbol{\mu}}(t), \ddot{\boldsymbol{\Sigma}}(t), \ddot{\boldsymbol{\tau}}(t))$.

We verify the second property of (3.106) which is $\dot{r}(0) = \xi$. It is readily check that $\dot{\boldsymbol{\mu}}(0) = \boldsymbol{\xi}_{\boldsymbol{\mu}}$ and $\dot{\boldsymbol{\Sigma}}(0) = \boldsymbol{\xi}_{\boldsymbol{\Sigma}}$. It remains to verify that $\dot{\boldsymbol{\tau}}(0) = \boldsymbol{\xi}_{\boldsymbol{\tau}}$. Computing the derivative of N (defined in Proposition 14) at a point $\boldsymbol{x}(t) \in (\mathbb{R}_*^+)^n$, we get that

$$\frac{d}{dt}(N \circ \boldsymbol{x})(t) = \left(\prod_{i=1}^n x_i(t)\right)^{-1/n} \left[\dot{\boldsymbol{x}}(t) - \frac{\dot{\boldsymbol{x}}(t)^T \boldsymbol{x}(t)^{\odot -1}}{n} \boldsymbol{x}(t)\right], \tag{3.107}$$

where $\dot{\boldsymbol{x}}(t) = \frac{d}{dt}\boldsymbol{x}(t)$ (and simply denoted $\dot{\boldsymbol{x}}(t)$). Using this derivative and the constraints $\prod_{i=1}^n \tau_i = 1$ and $\boldsymbol{\xi}_{\boldsymbol{\tau}}^T \boldsymbol{\tau}^{\odot -1} = 0$, the desired property is derived

$$\dot{\boldsymbol{\tau}}(0) = \frac{d}{dt}\left(N \circ \left(\boldsymbol{\tau} + t\boldsymbol{\xi}_{\boldsymbol{\tau}} + \frac{t^2}{2}\left(\boldsymbol{\xi}_{\boldsymbol{\tau}}^{\odot 2} \odot \boldsymbol{\tau}^{\odot -1} - \frac{1}{p}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\xi}_{\boldsymbol{\mu}} \mathbf{1}_n\right)\right)\right)\Big|_{t=0} \tag{3.108}$$

$$= \boldsymbol{\xi}_{\boldsymbol{\tau}}. \tag{3.109}$$

It remains to verify the third condition of (3.106). Using the first two conditions of (3.106), we find that $\nabla_{\dot{r}}^{\mathcal{M}_{p,n}^{\mathsf{FIM}}} \dot{r}\Big|_{t=0} = 0$ if and only if

$$\begin{cases} \ddot{\boldsymbol{\mu}}(0) = \left[\frac{\boldsymbol{\xi}_{\boldsymbol{\tau}}^T \boldsymbol{\tau}^{\odot -2}}{\sum_{i=1}^n \frac{1}{\tau_i}} \boldsymbol{I}_p + \boldsymbol{\xi}_{\boldsymbol{\Sigma}} \boldsymbol{\Sigma}^{-1}\right] \boldsymbol{\xi}_{\boldsymbol{\mu}}, \\ \ddot{\boldsymbol{\Sigma}}(0) = \boldsymbol{\xi}_{\boldsymbol{\Sigma}} \boldsymbol{\Sigma}^{-1} \boldsymbol{\xi}_{\boldsymbol{\Sigma}} - \frac{1}{n}\sum_{i=1}^n \left(\frac{1}{\tau_i}\right) \boldsymbol{\xi}_{\boldsymbol{\mu}} \boldsymbol{\xi}_{\boldsymbol{\mu}}^H, \\ P_{\boldsymbol{\tau}}^{\mathcal{S}(\mathbb{R}_*^+)^n}\left(\ddot{\boldsymbol{\tau}}(0)\right) = P_{\boldsymbol{\tau}}^{\mathcal{S}(\mathbb{R}_*^+)^n}\left(\boldsymbol{\xi}_{\boldsymbol{\tau}}^{\odot 2} \odot \boldsymbol{\tau}^{\odot -1} - \frac{1}{p}\boldsymbol{\xi}_{\boldsymbol{\mu}}^H \boldsymbol{\Sigma}^{-1} \boldsymbol{\xi}_{\boldsymbol{\mu}} \mathbf{1}_n\right), \end{cases} \tag{3.110}$$

where, $\forall \boldsymbol{\xi} \in \mathbb{R}^n$, $P_{\boldsymbol{\tau}}^{\mathcal{S}(\mathbb{R}_*^+)^n}(\boldsymbol{\xi}) = \boldsymbol{\xi} - \frac{\boldsymbol{\xi}^T \boldsymbol{\tau}^{\odot -1}}{n}\boldsymbol{\tau}$. It is readily checked that the first two conditions of (3.110) are met. Thus, only the third condition remains to be verified. To do so, we differentiate (3.107) to get the second

derivative of N,

$$\frac{d^2}{dt^2}(N \circ \boldsymbol{x})(t) = -\frac{1}{n}\left(\prod_{i=1}^{n} x_i(t)\right)\left(\dot{\boldsymbol{x}}(t)^T \boldsymbol{x}(t)^{\odot -1}\right)\left(\prod_{i=1}^{n} x_i(t)\right)^{-\frac{1}{n}-1} \tag{3.111}$$

$$\times \left[\dot{\boldsymbol{x}}(t) - \frac{1}{n}\left(\dot{\boldsymbol{x}}(t)^T \boldsymbol{x}(t)^{\odot -1}\right)\boldsymbol{x}(t)\right]$$

$$+ \left(\prod_{i=1}^{n} x_i(t)\right)^{-\frac{1}{n}}\left[\ddot{\boldsymbol{x}}(t) - \frac{1}{n}\left(\ddot{\boldsymbol{x}}(t)^T \boldsymbol{x}(t)^{\odot -1}\right)\boldsymbol{x}(t)\right.$$

$$\left. + \frac{1}{n}\left[\left(\dot{\boldsymbol{x}}(t)^{\odot 2}\right)^T \boldsymbol{x}(t)^{\odot -2}\right]\boldsymbol{x}(t) - \frac{1}{n}\left(\dot{\boldsymbol{x}}(t)^T \boldsymbol{x}(t)^{\odot -1}\right)\dot{\boldsymbol{x}}(t)\right]$$

$$= \frac{1}{n}\left(\prod_{i=1}^{n} x_i(t)\right)^{-\frac{1}{n}}\left[n\ddot{\boldsymbol{x}}(t)\right. \tag{3.112}$$

$$+ \left(\left(\dot{\boldsymbol{x}}(t)^{\odot 2}\right)^T \boldsymbol{x}(t)^{\odot -2} - \ddot{\boldsymbol{x}}(t)^T \boldsymbol{x}(t)^{\odot -1}\right)\boldsymbol{x}(t)$$

$$\left. - 2\left(\dot{\boldsymbol{x}}(t)^T \boldsymbol{x}(t)^{\odot -1}\right)\dot{\boldsymbol{x}}(t) + \frac{1}{n}\left(\dot{\boldsymbol{x}}(t)^T \boldsymbol{x}(t)^{\odot -1}\right)^2 \boldsymbol{x}(t)\right].$$

where $\ddot{\boldsymbol{x}}(t) = \frac{d^2}{dt^2}\boldsymbol{x}(t)$. Using this derivative and the constraints $\prod_{i=1}^{n} \tau_i = 1$ and $\boldsymbol{\xi}_{\boldsymbol{\tau}}^T \boldsymbol{\tau}^{\odot -1} = 0$, the following expression of $\ddot{\boldsymbol{\tau}}(0)$ is derived

$$\ddot{\boldsymbol{\tau}}(0) = \frac{d^2}{dt^2}\left(N \circ \left(\boldsymbol{\tau} + t\boldsymbol{\xi}_{\boldsymbol{\tau}} + \frac{t^2}{2}\left(\boldsymbol{\xi}_{\boldsymbol{\tau}}^{\odot 2} \odot \boldsymbol{\tau}^{\odot -1} - \frac{1}{p}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\mu}}\mathbf{1}_n\right)\right)\right)\Big|_{t=0} \tag{3.113}$$

$$= \boldsymbol{\xi}_{\boldsymbol{\tau}}^{\odot 2} \odot \boldsymbol{\tau}^{\odot -1} - \frac{1}{p}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\mu}}\mathbf{1}_n \tag{3.114}$$

$$+ \frac{1}{n}\left[\left(\boldsymbol{\xi}_{\boldsymbol{\tau}}^{\odot 2}\right)^T \boldsymbol{\tau}^{\odot -2} - \left(\boldsymbol{\xi}_{\boldsymbol{\tau}}^{\odot 2} \odot \boldsymbol{\tau}^{\odot -1} - \frac{1}{p}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\mu}}\mathbf{1}_n\right)^T \boldsymbol{\tau}^{\odot -1}\right]\boldsymbol{\tau}$$

$$= \boldsymbol{\xi}_{\boldsymbol{\tau}}^{\odot 2} \odot \boldsymbol{\tau}^{\odot -1} - \frac{1}{p}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\mu}}\mathbf{1}_n + \frac{1}{np}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\mu}}\left(\mathbf{1}_n^T \boldsymbol{\tau}^{\odot -1}\right)\boldsymbol{\tau}. \tag{3.115}$$

Using the linearity of the projection $P_{\boldsymbol{\tau}}^{\mathcal{S}(\mathbb{R}_*^+)^n}$, (3.114) implies that

$$P_{\boldsymbol{\tau}}^{\mathcal{S}(\mathbb{R}_*^+)^n}(\ddot{\boldsymbol{\tau}}(0)) = P_{\boldsymbol{\tau}}^{\mathcal{S}(\mathbb{R}_*^+)^n}\left(\boldsymbol{\xi}_{\boldsymbol{\tau}}^{\odot 2} \odot \boldsymbol{\tau}^{\odot -1} - \frac{1}{p}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\mu}}\mathbf{1}_n\right) \tag{3.116}$$

$$+ P_{\boldsymbol{\tau}}^{\mathcal{S}(\mathbb{R}_*^+)^n}\left(\frac{1}{np}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\mu}}\left(\mathbf{1}_n^T \boldsymbol{\tau}^{\odot -1}\right)\boldsymbol{\tau}\right).$$

Finally, one can check that $\forall \alpha \in \mathbb{R}$, $P_{\boldsymbol{\tau}}^{\mathcal{S}(\mathbb{R}_*^+)^n}(\alpha\boldsymbol{\tau}) = 0$. Hence, we get the

desired expression

$$P_{\boldsymbol{\tau}}^{\mathcal{S}(\mathbb{R}_*^+)^n}(\ddot{\boldsymbol{\tau}}(0)) = P_{\boldsymbol{\tau}}^{\mathcal{S}(\mathbb{R}_*^+)^n}\left(\boldsymbol{\xi}_{\boldsymbol{\tau}}^{\odot 2} \odot \boldsymbol{\tau}^{\odot -1} - \frac{1}{p}\boldsymbol{\xi}_{\boldsymbol{\mu}}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}_{\boldsymbol{\mu}} \mathbf{1}_n\right), \tag{3.117}$$

which completes the proof.

3.A.8 . Proof of Proposition 17: Existence of a regularized MLE in $\mathcal{M}_{p,n}$

$\mathcal{L}_{\mathcal{R}_\kappa}$ is a continuous function on $\mathcal{M}_{p,n}$. Hence, to prove the existence of a solution of the minimization problem (3.45), it is enough to show that

$$\lim_{\theta \to \partial\theta} \mathcal{L}_{\mathcal{R}_\kappa}(\theta | \{\boldsymbol{x}_i\}_{i=1}^n) = +\infty \tag{3.118}$$

where $\partial\theta$ is the boundary of $\mathcal{M}_{p,n}$.

First, it is easily checked that, for $\boldsymbol{\Sigma}$ and $\boldsymbol{\tau}$ not tending to the boundaries $\partial\mathcal{S}_p^{++}$ and $\partial\mathcal{S}(\mathbb{R}_*^+)^n$ of $\mathcal{S}_p^{++}$ and $\mathcal{S}(\mathbb{R}_*^+)^n$ respectively, we have

$$\lim_{\|\boldsymbol{\mu}\| \to +\infty} \mathcal{L}_{\mathcal{R}_\kappa}(\theta | \{\boldsymbol{x}_i\}_{i=1}^n) = +\infty. \tag{3.119}$$

Second, we handle the cases where $\boldsymbol{\Sigma} \to \partial\mathcal{S}_p^{++}$ and/or $\boldsymbol{\tau} \to \partial\mathcal{S}(\mathbb{R}_*^+)^n$. This means that, at least, one $\lambda_j \to \partial\mathbb{R}_*^+$ and/or one $\tau_i \to \partial\mathbb{R}_*^+$, with $\partial\mathbb{R}_*^+$ being the boundary of $\mathbb{R}_*^+$, *i.e.* 0^+ or $+\infty$. Using the positivity of the quadratic form in the NLL (3.27), we get the following inequality

$$\mathcal{L}(\theta | \{\boldsymbol{x}_i\}_{i=1}^n) \geq \sum_{i=1}^n \log |\tau_i \boldsymbol{\Sigma}| \,. \tag{3.120}$$

Hence, we get the resulting inequality on the regularized cost function

$$\mathcal{L}_{\mathcal{R}_\kappa}(\theta | \{\boldsymbol{x}_i\}_{i=1}^n) \geq \sum_{i=1}^n \sum_{j=1}^p \left[\log(\tau_i \lambda_j) + \beta r_\kappa(\tau_i \lambda_j)\right]. \tag{3.121}$$

We can remark that the lower bound (3.121) does not depend on $\boldsymbol{\mu}$. Hence, in the rest of the proof, we consider $\boldsymbol{\mu}$ to be either such that $\|\boldsymbol{\mu}\| < +\infty$ or such that $\|\boldsymbol{\mu}\| \to +\infty$. Then, we give a sufficient condition to prove (3.118) when $\boldsymbol{\Sigma} \to \partial\mathcal{S}_p^{++}$ and/or $\boldsymbol{\tau} \to \partial\mathcal{S}(\mathbb{R}_*^+)^n$. To give this sufficient condition, we first recall Assumption 3, $\forall \beta \in \mathbb{R}_*^+$

$$\lim_{x \to \partial\mathbb{R}_*^+} \log(x) + \beta r(x) = +\infty. \tag{3.122}$$

Thus, to prove (3.118), a sufficient condition, when $\boldsymbol{\Sigma} \to \partial\mathcal{S}_p^{++}$ and/or $\boldsymbol{\tau} \to \partial\mathcal{S}(\mathbb{R}_*^+)^n$, is that there exists at least one term $\tau_i \lambda_j$ such that

$$\tau_i \lambda_j \to \partial\mathbb{R}_*^+. \tag{3.123}$$

Since $\boldsymbol{\Sigma} \to \partial\mathcal{S}_p^{++}$ and/or $\boldsymbol{\tau} \to \partial\mathcal{S}(\mathbb{R}_*^+)^n$, there exists at least one $\lambda_j \to \partial\mathbb{R}_*^+$ and/or one $\tau_i \to \partial\mathbb{R}_*^+$. The condition (3.123) is of course met in the four following cases

$$\lambda_j \to 0^+ \text{ and/or } \tau_i \to 0^+, \tag{3.124}$$

$$\lambda_j \to +\infty \text{ and/or } \tau_i \to +\infty, \tag{3.125}$$

$$\lambda_j \to 0^+ \text{ and } \tau_i \to +\infty \text{ such that } \tau_i\lambda_j \to \partial\mathbb{R}_*^+, \tag{3.126}$$

$$\lambda_j \to +\infty \text{ and } \tau_i \to 0^+ \text{ such that } \tau_i\lambda_j \to \partial\mathbb{R}_*^+. \tag{3.127}$$

Finally, we treat the case where $\forall l \in \{1, \cdots, n\}$, $\lambda_l \to \partial\mathbb{R}_*^+$ and $\tau_i \to \partial\mathbb{R}_*^+$ such that the limit of $\tau_i\lambda_l$ is not $\partial\mathbb{R}_*^+$ (*i.e.* $\tau_i\lambda_l \nrightarrow \partial\mathbb{R}_*^+$). Since $\prod_{i=1}^n \tau_i = 1$, there exists at least one τ_q, with $q \neq i$, such that $\tau_q\lambda_j \to \partial\mathbb{R}_*^+$. Hence, the condition (3.123) is met, which completes the proof.

3.A.9 . Proof of Proposition 18: Minima of $\mathcal{R}_\kappa$

The objective of this proof is to solve

$$\underset{\theta \in \mathcal{M}_{p,n}}{\text{minimize}}\ \mathcal{R}_\kappa(\theta). \tag{3.128}$$

Using Assumption 5, we know that $\mathcal{R}_\kappa(\theta) \geq 0$ and $\mathcal{R}_\kappa(\theta) = 0 \iff \operatorname{diag}(\boldsymbol{\tau}) \otimes \boldsymbol{\Sigma} = \kappa \boldsymbol{I}_{n\times p}$. Thus, the minimum of $\mathcal{R}_\kappa$ is 0 and is reached at $\operatorname{diag}(\boldsymbol{\tau}) \otimes \boldsymbol{\Sigma} = \kappa \boldsymbol{I}_{n\times p}$, $\forall \boldsymbol{\mu} \in \mathbb{R}^p$. This implies that the minimum satisfies the following system of equations

$$\tau_i\lambda_j = \kappa \quad \forall i, j. \tag{3.129}$$

Hence, we deduce that $\tau_1 = \cdots = \tau_n$. Using the constraint $\prod_{i=1}^n \tau_i = 1$, we get that $\tau_1 = \cdots = \tau_n = 1$. Thus, $\lambda_1 = \cdots = \lambda_p = \kappa$. This means that

$$\{(\boldsymbol{\mu}, \kappa \boldsymbol{I}_{n\times p}, \mathbf{1}_n) : \boldsymbol{\mu} \in \mathbb{R}^p\} = \underset{\theta \in \mathcal{M}_{p,n}}{\arg\min}\ \mathcal{R}_\kappa(\theta) \tag{3.130}$$

which is Proposition 18.

3.A.10 . Proof of Proposition 19: Minima of $\mathcal{L}_{\mathcal{R}_\kappa}$ and rigid transformations

First of all, given $\boldsymbol{Q} \in \mathcal{O}_p$ and $\boldsymbol{\mu}_0 \in \mathbb{R}^p$, one can check that

$$\mathcal{L}\left(\widetilde{\theta} | \left\{\boldsymbol{Q}^T\boldsymbol{x}_i + \boldsymbol{\mu}_0\right\}_{i=1}^n\right) = \mathcal{L}\left(\theta | \{\boldsymbol{x}_i\}_{i=1}^n\right) \tag{3.131}$$

where $\mathcal{L}$ is the NLL defined in (3.27), $\theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\tau})$ and $\widetilde{\theta} = \left(\boldsymbol{Q}^T\boldsymbol{\mu} + \boldsymbol{\mu}_0, \boldsymbol{Q}^T\boldsymbol{\Sigma}\boldsymbol{Q}, \boldsymbol{\tau}\right)$. Then, $\mathcal{R}_\kappa$ satisfies Assumption 3 and thus only depends on the eigenvalues of the matrices $\tau_i\boldsymbol{\Sigma}$. This implies that $\mathcal{R}_\kappa(\widetilde{\theta}) = \mathcal{R}_\kappa(\theta)$ and hence we get that

$$\mathcal{L}_{\mathcal{R}_\kappa}\left(\widetilde{\theta} | \left\{\boldsymbol{Q}^T\boldsymbol{x}_i + \boldsymbol{\mu}_0\right\}_{i=1}^n\right) = \mathcal{L}_{\mathcal{R}_\kappa}\left(\theta | \{\boldsymbol{x}_i\}_{i=1}^n\right). \tag{3.132}$$

The equation (3.132) implies that, if

$$\theta = \underset{\theta \in \mathcal{M}_{p,n}}{\arg\min}\ \mathcal{L}_{\mathcal{R}_\kappa}\left(\theta | \{\boldsymbol{x}_i\}_{i=1}^n\right), \tag{3.133}$$

then

$$\widetilde{\theta} = \underset{\theta \in \mathcal{M}_{p,n}}{\arg\min}\ \mathcal{L}_{\mathcal{R}_\kappa}\left(\theta | \left\{\boldsymbol{Q}^T\boldsymbol{x}_i + \boldsymbol{\mu}_0\right\}_{i=1}^n\right) \tag{3.134}$$

which is exactly Proposition 19.

3.A.11 . Proof of Proposition 20: Kullback-Leibler divergence

In the following, we show that the KL divergence between two NC-MSGs is equal to the sum of KL divergences between Gaussian distributions with specific mean and covariance matrices:

$$\delta_{\mathsf{KL}}(\theta_1, \theta_2) = \int p_{\theta_1}(x) \log\left(\frac{p_{\theta_1}(x)}{p_{\theta_2}(x)}\right) dx \tag{3.135}$$

$$= \int \prod_{i=1}^n f(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \tau_{1,i}) \log\left(\prod_{i=1}^n \frac{f(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \tau_{1,i})}{f(\boldsymbol{x}_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2, \tau_{2,i})}\right) d\boldsymbol{x}_1 \cdots d\boldsymbol{x}_n \tag{3.136}$$

$$= \sum_{i=1}^n \int \prod_{j=1}^n f(\boldsymbol{x}_j; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \tau_{1,j}) \log\left(\frac{f(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \tau_{1,i})}{f(\boldsymbol{x}_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2, \tau_{2,i})}\right) d\boldsymbol{x}_1 \cdots d\boldsymbol{x}_n \tag{3.137}$$

$$= \sum_{i=1}^n \int f(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \tau_{1,i}) \log\left(\frac{f(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \tau_{1,i})}{f(\boldsymbol{x}_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2, \tau_{2,i})}\right) d\boldsymbol{x}_i \tag{3.138}$$

$$= \sum_{i=1}^n \int f_{\mathsf{G}}(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \tau_{1,i}\boldsymbol{\Sigma}_1) \log\left(\frac{f_{\mathsf{G}}(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \tau_{1,i}\boldsymbol{\Sigma}_1)}{f_{\mathsf{G}}(\boldsymbol{x}_i; \boldsymbol{\mu}_2, \tau_{2,i}\boldsymbol{\Sigma}_2)}\right) d\boldsymbol{x}_i. \tag{3.139}$$

Using the KL divergence between Gaussian distributions and the constraint $\prod_{i=1}^n \tau_{1,i} = \prod_{i=1}^n \tau_{2,i} = 1$, we get the desired formula

$$\delta_{\mathsf{KL}}(\theta_1, \theta_2) = \frac{1}{2}\Bigg(\sum_{i=1}^n \frac{\tau_{1,i}}{\tau_{2,i}} \operatorname{Tr}\left(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1\right) + \sum_{i=1}^n \frac{1}{\tau_{2,i}} \Delta\boldsymbol{\mu}^T \boldsymbol{\Sigma}_2^{-1} \Delta\boldsymbol{\mu} + n \log\left(\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|}\right) - np \Bigg). \tag{3.140}$$

4 - Probabilistic PCA from heteroscedastic signals

Principal Component Analysis (PCA) [131, 72] is a standard tool used in signal processing and machine learning literature for dimensional reduction and statistical interpretation. In this scope, Probabilistic PCA (PPCA) refers to a reformulation of PCA as a parametric estimation problem; see Chapter 1 Section 1.3 for a detailed presentation of PPCA. This approach was proposed in [131], which considered a model of White Gaussian Noise (WGN) plus a linear mapping of a low-dimensional centered Gaussian latent space with unit variance (the signal contribution). Leveraging the statistical formulation of PPCA allows going beyond Gaussian models. For example, the two independent contributions (either signal or noise) can be generalized to the distribution of compound Gaussian. The latter represents a family of elliptical distributions (cf. review in [104]) that encompasses numerous standard heavy-tailed models, such as the multivariate t-distribution. Its stochastic representation involves a Gaussian vector multiplied by an independent random power factor referred to as *texture*. In order to be robust to various underlying distributions, this parameter is often assumed to be unknown deterministic. This assumption yields the so-called *mixture of scaled Gaussian distributions* (MSG) [141], also referred to as *heteroscedastic* [65], and presented in Chapter 1 Section 1.3. In this scope [26, 13, 128] considered MSGs for the signal component to perform robust PCA for non-Gaussian signals. Conversely, [65] considered Gaussian signals embedded in white MSG noise to model data where some samples are noisier than others. Alternatively, [36] uses a t-distribution to model both of the contributions. Finally, [30] considered a mixture of three components to account for potential outliers (the thirds contribution being orthogonal to the signal subspace).

In the following, we will focus on MSG plus WGN model [26, 13, 128] which is interpreted as impulsive signals (power variation across samples) plus thermal noise due to electronics. A common relaxation of this model is to assume that eigenvalues of the (low-rank) signal covariance matrix are identical as in [115, 28]. Indeed, this hypothesis is relevant since we still estimate the power variations which contain, the information of the eigenvalues. Moreover, [26, 30, 10] showed that neglecting the differences between eigenvalues does not harm the accuracy of subspace estimation while allowing for a more meaningful statistical interpretation [115].

Yet, the previous studies still left some unanswered issues: first, the algorithms in [115, 28] are dedicated bloc-coordinate descent type. Thus, they can be limited in practice, as they offer no generalization to on-line (or

stochastic) settings. It would then be relevant for the estimation problem to be cast in a more generic optimization framework that can account for the parameter structure (*e.g.*, subspaces, vectors with strictly positive values). Second, the MLE of the considered model is the solution of a nonconvex problem with no guarantee for global optimality. Thus, it would be interesting to derive performance bound in order to assess for various algorithms performance. Such bound is not trivial for these models because structured parameters require accounting for specific constraints, as well as for the use of relevant distances as error measure (*e.g.* to ensure for some invariance). Finally, one can inquire if the features of such statistical model can be meaningfully leveraged in machine learning tasks such as clustering.

Therefore, this chapter conducts a study of the MSG plus WGN model [115, 28] through the prism of Riemannian geometry, as this theoretical framework allows us to propose a unified view to tackle the aforementioned questions. The contributions concern the following directions:

1. Riemannian optimization framework for model features: MSG plus WGN model involves parameters that are textures (power factors) and a low-rank subspace. Endowing this parameter space with a Riemannian metric yields a Riemannian manifold, which can be leveraged in an optimization framework [1] as presented in Chapter 2. In this context, we consider the model's Fisher information metric (FIM). We then obtain several essential tools (tangent space, Riemannian gradient, retraction) from established results on the Grassman manifold [52] presented in Chapter 2 Section 2.4. These tools are then used to propose algorithms in order to compute the MLE, as well as the Riemannian means used in clustering algorithms (cf. next points). We notably propose a Riemannian stochastic gradient descent algorithm [147] suited to large datasets (or online settings [152]).

2. Performance bounds: We show that the FIM of the considered model (and its corresponding Riemannian distance) permits to derive closed forms and product intrinsic Cramér-Rao lower bound (ICRB) for the model's parameters. These lower bounds represent partial extensions of [13] (Euclidean CRB in the case of colored signals) to the ICRB framework of [121], introduced Chapter 2 Section 2.5. Interestingly, the proposed approach offers a new interpretable result regarding problem dimensions and signal-to-noise ratio (SNR). Then, we assess the performance of different estimation algorithms numerically. We show that both the proposed estimation algorithm and the previously established block-coordinate algorithm [28] are statistically efficient for the signal subspace estimation. In a low SNR scenario, they also both outperform subspace estimated by Singular Value Decomposition (SVD)

in terms of MSE.

3. Applications to clustering: we propose a Riemannian clustering algorithm for data following the MSG plus WGN model. Here, we extend the methodology presented Chapter 1 Section 1.5 to the considered statistical model using a *K-means++* [7]. Replacing the Euclidean distance by a Riemannian one allows for this clustering algorithm to takes into account the geometrical constraints of the parameter space (invariance properties of subspaces and positivity of powers), which is shown to improve the clustering performance on the hyperspectral image Indian Pines benchmark [9].

This chapter is organized as follows. Section 4.1 presents the statistical model and the parameter space as a manifold. Section 4.2 presents a Riemannian geometry for this manifold, and essential tools driven from two possible metrics. Section 4.3 presents results related to parameter estimation (MLE based on Riemannian optimization and ICRBs). Section 4.4 presents a clustering algorithm (Riemannian *K-means++*) adapted to the considered parameter manifold. Numerical results are presented in Section 4.5. Appendix 4.A contains the technical proofs.

In the rest of the chapter, the model, ICRBs and clustering algorithm are derived for complex valued data and all the calculus are realised with complex numbers. This is opposed to all other chapters of this manuscript that have been written with real numbers. However, it should be noted that the calculus, formulas and algorithms of this chapter can easily be adapted to real valued data and to the model of real valued MSG plus real valued WGN noise. Moreover, the clustering pipeline is applied to the *Indian pines* dataset which are real valued data which is coherent with Chapter 1.

4.1 . Heteroschedastic signal model and its parameter space

4.1.1 . Statistical model

Let $\{\boldsymbol{x}_i\}_{i=1}^n$ be a dataset of p-dimensional complex vectors. We consider a k-dimensional linear signal representation embedded in white Gaussian noise, *i.e.* the model:

$$\boldsymbol{x} \overset{d}{=} \boldsymbol{U}\,\boldsymbol{g} + \boldsymbol{n}, \tag{4.1}$$

where $\boldsymbol{g} \in \mathbb{C}^k$ is the signal of interest, $\boldsymbol{n} \sim \mathbb{C}\mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_p)$ is a white Gaussian noise, and $\boldsymbol{U} \in \mathsf{St}_{p,k}$ is an orthonormal basis of the signal subspace, where

$$\mathsf{St}_{p,k} = \left\{\boldsymbol{U} \in \mathbb{C}^{p\times k} : \; \boldsymbol{U}^H\boldsymbol{U} = \boldsymbol{I}_k\right\}, \tag{4.2}$$

denotes the complex Stiefel manifold. In array-processing literature it is classically assumed that $\boldsymbol{g} \sim \mathbb{C}\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$, which yields a low-rank structured

Gaussian model, also referred to as the (Gaussian) Probabilistic PCA (PPCA) model in [131]. Note that these models often rely on the unconstrained identification $\boldsymbol{x} \overset{d}{=} \boldsymbol{W}\,\tilde{\boldsymbol{g}} + \boldsymbol{n}$, with $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Sigma}^{1/2}$ and $\tilde{\boldsymbol{g}} \sim \mathbb{C}\mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_k)$. However, using $\boldsymbol{U} \in \mathrm{St}_{p,k}$ is here more coherent with later developments.

In order to model heavy-tailed signals (e.g., outliers or power discrepancies), several works [115, 26, 13, 128] considered generalizing the Gaussian PPCA to compound Gaussian distributions [104]. Such signal model yields

$$\boldsymbol{x}_i|\tau_i \overset{d}{=} \sqrt{\tau_i}\,\boldsymbol{U}\,\boldsymbol{g} + \boldsymbol{n}, \tag{4.3}$$

where $\boldsymbol{g} \sim \mathbb{C}\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ and $\tau_i \in \mathbb{R}_*^+$ is a random power factor referred to as texture, which is statistically independent of $\boldsymbol{g}$. Starting from this representation, we make the following additional assumptions:

• *Known noise floor*: The variance σ^2 is considered known. If σ^2 is unknown in practice, it can be accurately pre-estimated by averaging lowest eigenvalues of the SCM [131]. The hypothesis of known σ^2 simplifies the exposition and does not change significantly the performance in practice when compared to a joint estimation scheme (see *e.g.* [90]). Without loss of generality, such assumption allows us to set $\sigma^2 = 1$.

• *Unknown deterministic textures*: In order to provide a model that is robust to any underlying compound Gaussian distribution, it is often assumed that the textures $\{\tau_i\}_{i=1}^n$ are unknown deterministic rather than assigning it a predetermined probability density function [26, 13, 128]. Such distribution is then referred to as MSG. • *Isotropic signal*: We consider the relaxation from [115, 28], assuming that the eigenvalues of the signal covariance matrix are identical, *i.e.*, $\boldsymbol{g} \sim \mathbb{C}\mathcal{N}(\boldsymbol{0}, \sigma_s \boldsymbol{I}_k)$. This relaxation greatly simplifies the study of the statistical model as well as the Riemannian geometry of its parameter space. Indeed, considering non equal eigenvalues forces to develop more complicated Riemannian quotient manifolds than the one presented in Section 4.2; see for example [17]. In conjunction with the unknown deterministic textures assumption, considering identical eigenvalues allows the change of variable $\tilde{\tau}_i = \sigma_s \tau_i$, and thus setting $\sigma_s = 1$ without loss of generality. While apparently not realistic, this hypothesis is still representative since the average signal power information is accounted for by the texture parameters. Moreover, [26, 30, 10] showed that neglecting the differences between eigenvalues does not harm the accuracy of subspace estimation while allowing for a more meaningful statistical interpretation [115].

Finally, we have the data $\{\boldsymbol{x}_i\}_{i=1}^n$ distributed as in (4.3) where $\boldsymbol{g} \sim \mathbb{C}\mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_k)$ and $\boldsymbol{n} \sim \mathbb{C}\mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_p)$. The unknown model parameters are the textures $\{\tau_i\}_{i=1}^p$ (denoted by the vector $\boldsymbol{\tau} \in (\mathbb{R}_*^+)^n$) and the signal subspace, represented by a basis $\boldsymbol{U} \in \mathrm{St}_{p,k}$. The following section will recast this parameter space as a manifold. This reformulation will then allow us to leverage tools from the Riemannian geometry in order to derive distances,

intrinsic Cramér-Rao Bounds and optimization methods with a unified view.

4.1.2 . Manifold approach to the parameter space

Due to their specific geometrical structure, the parameters $(\boldsymbol{U}, \boldsymbol{\tau})$ of model (4.3) can be embedded into the product manifold $\overline{\mathcal{M}}_{p,k,n} = \mathsf{St}_{p,k} \times (\mathbb{R}_*^+)^n$. With this model, from $\overline{\mathcal{M}}_{p,k,n}$, the scaled covariance matrix in $\mathcal{H}_p^{++}$ of sample $\boldsymbol{x}_i$ is obtained through the function

$$\begin{array}{rccl} \bar{\psi}_i : & \overline{\mathcal{M}}_{p,k,n} & \to & \mathcal{H}_p^{++} \\ & (\boldsymbol{U}, \boldsymbol{\tau}) & \mapsto & \boldsymbol{I}_p + \tau_i \boldsymbol{U}\boldsymbol{U}^H. \end{array} \tag{4.4}$$

It follows that the negative log-likelihood corresponding to model (4.3) is given, for all $\bar{\theta} = (\boldsymbol{U}, \boldsymbol{\tau}) \in \overline{\mathcal{M}}_{p,k,n}$, by

$$\bar{\mathcal{L}}(\bar{\theta}) = \sum_i \log \left|\bar{\psi}_i(\bar{\theta})\right| + \boldsymbol{x}_i^H \left(\bar{\psi}_i(\bar{\theta})\right)^{-1} \boldsymbol{x}_i. \tag{4.5}$$

The model (4.3) is ambiguous since the representation by the basis $\boldsymbol{U}$ is invariant by rotation: for all $\boldsymbol{O} \in \mathcal{U}_k$ (where $\mathcal{U}_k$ is the unitary group of degree k), $(\boldsymbol{U}\boldsymbol{O}, \boldsymbol{\tau})$ is equivalent to $(\boldsymbol{U}, \boldsymbol{\tau})$, *i.e.*, it yields the same scaled covariance matrices in $\mathcal{H}_p^{++}$. The consequence is that the manifold $\overline{\mathcal{M}}_{p,k,n}$ is not optimal with respect to the model of interest. In terms of optimization, for instance for maximum likelihood estimation, it is possible to exploit $\overline{\mathcal{M}}_{p,k,n}$ directly but it is advantageous to take into account the invariance. Moreover, to measure estimation errors or perform geometrical classification and clustering, employing a distance function onto $\overline{\mathcal{M}}_{p,k,n}$ is not ideal: the distance between two equivalent points is not equal to zero. It thus appears very attractive to take this invariance into account.

Fortunately, it is possible to naturally handle this rotation invariance from a geometrical perspective. It is achieved by considering the Grassmann manifold $\mathsf{Gr}_{p,k}$ (set of all k-dimensional subspaces of $\mathbb{C}^p$) presented in Chapter 2 Section 2.4. The Grassmann manifold can be identified to the quotient manifold [52, 2, 1]

$$\mathsf{Gr}_{p,k} = \{\{\boldsymbol{U}\boldsymbol{O} : \boldsymbol{O} \in \mathcal{U}_k\} : \boldsymbol{U} \in \mathsf{St}_{p,k}\}. \tag{4.6}$$

From there, to optimally embed the parameters of model (4.3), we construct the manifold $\mathcal{M}_{p,k,n} = \mathsf{Gr}_{p,k} \times (\mathbb{R}_*^+)^n$. This manifold can be viewed as a quotient manifold of $\overline{\mathcal{M}}_{p,k,n}$ (see Chapter 2 Section 2.3 for an introduction to quotient manifolds). Indeed, it can be defined as

$$\mathcal{M}_{p,k,n} = \{\pi(\bar{\theta}) : \bar{\theta} \in \overline{\mathcal{M}}_{p,k,n}\}, \tag{4.7}$$

where, for all $\bar{\theta} = (\boldsymbol{U}, \boldsymbol{\tau}) \in \overline{\mathcal{M}}_{p,k,n}$, the equivalence class is defined as

$$\pi(\bar{\theta}) = \{(\boldsymbol{U}\boldsymbol{O}, \boldsymbol{\tau}) : \boldsymbol{O} \in \mathcal{U}_k\}. \tag{4.8}$$

Functions $\overline{\psi}_i$ defined onto $\overline{\mathcal{M}}_{p,k,n}$ induce functions ψ_i onto $\mathcal{M}_{p,k,n}$, *i.e.* $\overline{\psi}_i(\bar{\theta}) = \psi_i(\pi(\bar{\theta}))$. Thus, $\boldsymbol{x}_i$ in (4.3) is drawn as $\boldsymbol{x}_i \sim \mathbb{C}\mathcal{N}(\boldsymbol{0}, \psi_i(\theta))$. It follows that the log-likelihood $\overline{\mathcal{L}}$ in (4.5) defined onto $\overline{\mathcal{M}}_{p,k,n}$ can also be defined onto $\mathcal{M}_{p,k,n}$ by using functions ψ_i instead of $\overline{\psi}_i$. This log-likelihood function is denoted $\mathcal{L}$ in the following.

Besides acknowledging the model invariances, considering $\mathcal{M}_{p,k,n}$ as a manifold allows for advantageously exploiting Riemannian geometry, *i.e.*, the geometries of $\mathcal{M}_{p,k,n}$ induced by Riemannian metrics. In particular for signal processing applications, it can be leveraged for:

1. Estimation: the Riemannian optimization framework can be employed to compute maximum likelihood estimators (Section 4.3.1) and Riemannian means (Section 4.4) in various practical scenarios.

2. Performance measuring: the Riemannian distance naturally defines an error measure, which can then be bounded using the framework of intrinsic Cramér-Rao bound [121]. This point will be detailed in Section 4.3.2.

3. Machine learning: the Riemannian distance can also be exploited to cluster and classify various data which follow model (4.3), which will be further discussed in Section 4.4.

In order to achieve these, different geometrical objects are needed. Section 4.2 will introduce these tools conditionally to the choice of the Riemannian metric.

4.2 . Riemannian manifolds of interest

Various choices of Riemannian geometries are available for $\mathcal{M}_{p,k,n}$, entirely depending on the choice of the Riemannian metric. Among different possibilities, one is optimal with respect to the considered statistical model: the Fisher information metric [3]. Indeed, it is derived from the log-likelihood function of the distribution at hand and thus perfectly captures the particularities of the model. However, the geometry induced by the Fisher information metric is often hard to fully leverage. One has therefore to compromise and define an alternate geometry (induced by a metric as close as possible to the Fisher one) in order to obtain tractable expressions for the needed geometrical tools.

		Tools for Riemmanian optimization	
Metric	Horizontal space $\mathcal{H}_{\bar{\theta}}$	Riemannian gradient	Retraction
Fisher information metric (4.13)	(4.14)	Prop. 22 for $\mathcal{L}$	(4.16)
Product metric (4.17)	(4.14)	$\sim$	$\sim$

Table 4.1: Summary of the geometric tools of $\mathcal{M}_{p,k,n}$ for optimization. Symbol $\sim$ means that it is not provided in this Chapter but that it could be easily derived.

		Tools for Riemannian distances			
Metric	Horizontal space $\mathcal{H}_{\bar{\theta}}$	Orthonormal basis of $\mathcal{H}_{\bar{\theta}}$	Distance	Exp.	Log.
Fisher information metric (4.13)	(4.14)	$\sim$	x	x	x
Product metric (4.17)	(4.14)	Prop. 25	(4.18)-(4.19)	(4.21)	(4.20)

Table 4.2: Summary of the geometric tools of $\mathcal{M}_{p,k,n}$ for distances. Symbol $\sim$ means that it is not provided in this Chapter but that it could be easily derived; and symbol x means that it is complicated to find and remains unknown.

In this section, we first provide an introduction on $\mathcal{M}_{p,k,n}$ viewed as a Riemannian quotient manifold in Section 4.2.1. We then study the Fisher information metric of likelihood (4.5) and derive the geometrical objects needed for Riemannian optimization in Section 4.2.2. However, required objects related to Riemannian distances cannot be obtained in closed-form. An alternate geometry using a product metric (close to the Fisher one) is thus proposed in order to achieve these in Section 4.2.3. The obtained results are summarized in Tables 4.1 and 4.2.

4.2.1 . $\mathcal{M}_{p,k,n}$ as a Riemannian quotient manifold

Since $\mathrm{Gr}_{p,k}$ is a quotient manifold of $\mathrm{St}_{p,k}$ with respect to the action of $\mathcal{U}_k$ [52], $\mathcal{M}_{p,k,n} = \mathrm{Gr}_{p,k} \times (\mathbb{R}_*^+)^n$ is a quotient of $\overline{\mathcal{M}}_{p,k,n} = \mathrm{St}_{p,k} \times (\mathbb{R}_*^+)^n$. To handle elements of $\mathcal{M}_{p,k,n}$, which are equivalence classes $\{(\boldsymbol{UO}, \boldsymbol{\tau}) : \boldsymbol{O} \in \mathcal{U}_k\}$, one usually exploits the canonical projection $\pi : \overline{\mathcal{M}}_{p,k,n} \to \mathcal{M}_{p,k,n}$ in (4.8). Equivalence classes are obtained through π as $\{(\boldsymbol{UO}, \boldsymbol{\tau}) : \boldsymbol{O} \in \mathcal{U}_k\} = \pi^{-1}(\pi(\boldsymbol{U}, \boldsymbol{\tau}))$ and each element $\theta \in \mathcal{M}_{p,k,n}$ can be represented by any $\bar{\theta} = (\boldsymbol{U}, \boldsymbol{\tau}) \in \overline{\mathcal{M}}_{p,k,n}$ such that $\theta = \pi(\bar{\theta})$. In general, geometrical

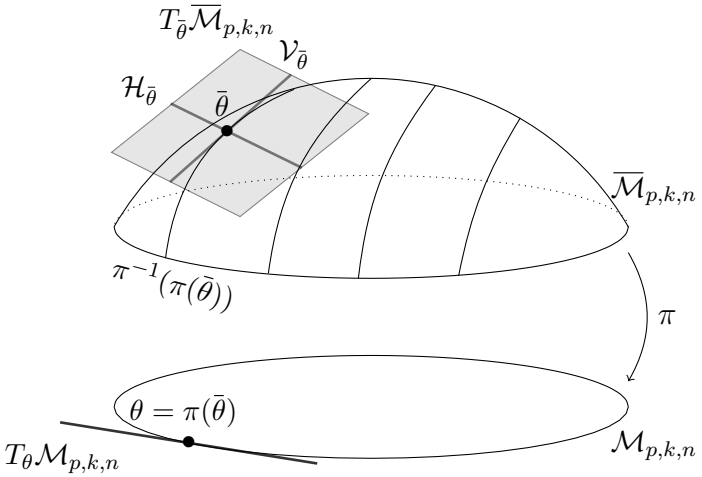


Figure 4.1: Illustration of the quotient $\mathcal{M}_{p,k,n}$ represented by elements of $\overline{\mathcal{M}}_{p,k,n}$. The set of all representations of $\theta = \pi(\bar{\theta}) \in \mathcal{M}_{p,k,n}$ is the equivalence class $\pi^{-1}(\pi(\bar{\theta})) \subset \overline{\mathcal{M}}_{p,k,n}$. The tangent space $T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$ can be decomposed into the vertical space $\mathcal{V}_{\bar{\theta}} = T_{\boldsymbol{U}}\pi^{-1}(\pi(\bar{\theta}))$ and its orthogonal complement, the horizontal space $\mathcal{H}_{\bar{\theta}}$, which provides proper representatives for tangent vectors in $T_\theta\mathcal{M}_{p,k,n}$. See Chapter 2 Section 2.3 for an introduction to Riemannian quotient manifolds.

objects on $\mathcal{M}_{p,k,n}$ can be represented by objects on $\overline{\mathcal{M}}_{p,k,n}$. A schematic illustration of the quotient manifold is provided in Figure 4.1.

The tangent space $T_\theta\mathcal{M}_{p,k,n}$ of $\theta = \pi(\bar{\theta}) \in \mathcal{M}_{p,k,n}$ can be represented by a subspace of the tangent space $T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$. First, we note that

$$T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n} = T_{\boldsymbol{U}}\mathsf{St}_{p,k} \times T_{\boldsymbol{\tau}}(\mathbb{R}_*^+)^n \tag{4.9}$$

$$= \{(\boldsymbol{\xi}_{\boldsymbol{U}}, \boldsymbol{\xi}_{\boldsymbol{\tau}}) \in \mathbb{C}^{p\times k} \times \mathbb{R}^n : \boldsymbol{U}^H\boldsymbol{\xi}_{\boldsymbol{U}} + \boldsymbol{\xi}_{\boldsymbol{U}}^H\boldsymbol{U} = \boldsymbol{0}\}. \tag{4.10}$$

thanks to $T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$ being a product manifold, and standard results on $\mathsf{St}_{p,k}$ and $(\mathbb{R}_*^+)^n$ respectively. The tangent space $T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$ can now be decomposed into two complementary subspaces: the vertical and horizontal subspaces [1]. The vertical space is defined as the tangent space $T_{\bar{\theta}}\pi^{-1}(\pi(\bar{\theta}))$ of the equivalence class $\pi^{-1}(\pi(\bar{\theta}))$ at $\bar{\theta}$. In the case of $\mathcal{M}_{p,k,n}$, the vertical space at $\bar{\theta}$ is

$$\mathcal{V}_{\bar{\theta}} = \{(\boldsymbol{U}\boldsymbol{A}, \boldsymbol{0}) : \boldsymbol{A} \in \mathcal{H}_k^\perp\}, \tag{4.11}$$

where $\mathcal{H}_k^\perp = \{\boldsymbol{A} \in \mathbb{C}^{k\times k} : \boldsymbol{A}^H = -\boldsymbol{A}\}$ is the set of $k \times k$ skew-Hermitian matrices. The orthogonal complement of the vertical space $\mathcal{V}_{\bar{\theta}}$ is the horizontal space $\mathcal{H}_{\bar{\theta}}$, which provides proper representations of the tangent vectors in $T_\theta\mathcal{M}_{p,k,n}$ called horizontal lifts. Indeed, there is a one-to-one correspondence between elements of $T_\theta\mathcal{M}_{p,k,n}$ and those of $\mathcal{H}_{\bar{\theta}}$, *i.e.* each element

$\xi \in T_\theta \mathcal{M}_{p,k,n}$ is represented by its unique horizontal lift, denoted $\mathrm{lift}_{\bar{\theta}}(\xi)$, in $\mathcal{H}_{\bar{\theta}}$. Note that the notion of orthogonal complement is conditioned by the choice of an inner product $\langle \cdot, \cdot \rangle_{\bar{\theta}}$ defined on $T_{\bar{\theta}} \overline{\mathcal{M}}_{p,k,n}$, which will also turn $\mathcal{M}_{p,k,n}$ into a Riemannian manifold.

Indeed, a Riemannian manifold is a manifold endowed with a Riemmanian metric (inner product defined for every tangent space). In the case of a Riemannian quotient manifold, such metric can be represented by a metric on $\overline{\mathcal{M}}_{p,k,n}$, *i.e.*, an inner product $\langle \cdot, \cdot \rangle_{\bar{\theta}}$ defined for $T_{\bar{\theta}} \overline{\mathcal{M}}_{p,k,n}$ at each point $\bar{\theta}$. Still, for $\mathcal{M}_{p,k,n}$ to be properly defined as a Riemannian quotient manifold, this metric on $\overline{\mathcal{M}}_{p,k,n}$ has to be invariant along each equivalence class. In our case, for all $\boldsymbol{O} \in \mathcal{U}_k$, $\bar{\theta} = (\boldsymbol{U}, \boldsymbol{\tau}) \in \overline{\mathcal{M}}_{p,k,n}$, $\bar{\xi} = (\boldsymbol{\xi}_{\boldsymbol{U}}, \boldsymbol{\xi}_{\boldsymbol{\tau}})$ and $\bar{\eta} = (\boldsymbol{\eta}_{\boldsymbol{U}}, \boldsymbol{\eta}_{\boldsymbol{\tau}})$ in $T_{\bar{\theta}} \overline{\mathcal{M}}_{p,k,n}$, we must have

$$\langle \bar{\xi}, \bar{\eta} \rangle_{\bar{\theta}} = \langle (\boldsymbol{\xi}_{\boldsymbol{U}} \boldsymbol{O}, \boldsymbol{\xi}_{\boldsymbol{\tau}}), (\boldsymbol{\eta}_{\boldsymbol{U}} \boldsymbol{O}, \boldsymbol{\eta}_{\boldsymbol{\tau}}) \rangle_{(\boldsymbol{U}\boldsymbol{O}, \boldsymbol{\tau})}. \tag{4.12}$$

The choice of such Riemannian metric on $\overline{\mathcal{M}}_{p,k,n}$ will then induce a specific geometry (and corresponding theoretical tools) for this space.

4.2.2 . Fisher information metric: geometry for optimization

First, we consider the geometry resulting from the Fisher information metric of corresponding to likelihood (4.5) on $\overline{\mathcal{M}}_{p,k,n}$. Since the statistical model is invariant along equivalence classes, the corresponding Fisher metric satisfies (4.12). It thus induces a Riemannian metric onto $\mathcal{M}_{p,k,n}$. To do so, we first derive this metric in Proposition 21.

Proposition 21 (Fisher information metric)**.** *The Fisher information metric at $\bar{\theta}$ corresponding to the negative likelihood* (4.5) *is, for all $\bar{\xi}$, $\bar{\eta} \in T_{\bar{\theta}} \overline{\mathcal{M}}_{p,k,n}$,*

$$\begin{aligned} \langle \bar{\xi}, \bar{\eta} \rangle_{\bar{\theta}}^{\mathsf{FIM}} = 2\, n\, c_{\boldsymbol{\tau}}\, \mathfrak{Re}\left(\mathrm{Tr}\left(\boldsymbol{\xi}_{\boldsymbol{U}}^H \boldsymbol{\eta}_{\boldsymbol{U}}\right)\right) \\ + k \left(\boldsymbol{\xi}_{\boldsymbol{\tau}} \odot (\mathbf{1} + \boldsymbol{\tau})^{\odot -1}\right)^T \left(\boldsymbol{\eta}_{\boldsymbol{\tau}} \odot (\mathbf{1} + \boldsymbol{\tau})^{\odot -1}\right), \end{aligned} \tag{4.13}$$

where $c_{\boldsymbol{\tau}} = \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} \frac{\tau_i^2}{1 + \tau_i}$.

Proof. See Appendix 4.A.1. □

The part of the Fisher metric in the above proposition which is related to $\boldsymbol{U}$, *i.e.*, the part that depends on components $\boldsymbol{\xi}_{\boldsymbol{U}}$ and $\boldsymbol{\eta}_{\boldsymbol{U}}$, is equal to the classical metric on Grassmann [2, 1, 52], up to the factor $2nc_{\boldsymbol{\tau}}$. We can also note that this factor does not affect the classical definition of the horizontal

space of the Grassmann manifold. This directly yields that the horizontal space $\mathcal{H}_{\bar{\theta}}$ in $T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$ associated with the metric of Proposition 21 is

$$\mathcal{H}_{\bar{\theta}} = \{(\boldsymbol{\xi}_{\boldsymbol{U}}, \boldsymbol{\xi}_{\boldsymbol{\tau}}) \in \mathbb{C}^{p\times k} \times \mathbb{R}^n : \boldsymbol{U}^H \boldsymbol{\xi}_{\boldsymbol{U}} = \mathbf{0}\}. \tag{4.14}$$

Unfortunately, the geometry of $\mathcal{M}_{p,k,n}$ associated with the Fisher information metric of Proposition 21 is complicated to fully characterize. In particular, finding the geodesics of $\mathcal{M}_{p,k,n}$ (curves of minimal length between two points in $\mathcal{M}_{p,k,n}$) is very hard because of the factor $c_{\boldsymbol{\tau}}$ in the metric. In this part, we will focus on the use of the Fisher information metric in the framework of Riemannian optimization [1]. Alternate tractable geometric tools regarding geodesics and distance measurements (Riemannian exponential and logarithm mapping, Riemannian distance), will be obtained from a product metric in Section 4.2.3.

We will consider optimization problems of the form

$$\underset{\theta\in\mathcal{M}_{p,k,n}}{\text{minimize}}\ h(\theta) \tag{4.15}$$

for a cost function $h : \mathcal{M}_{p,k,n} \to \mathbb{R}$, induced by $\bar{h} : \overline{\mathcal{M}}_{p,k,n} \to \mathbb{R}$ invariant along equivalence classes (*i.e.*, $\bar{h} = h \circ \pi$). In order to perform first order Riemannian optimization algorithms, we need a retraction (operator transforming tangent vectors into points onto the manifold) [1].

To obtain a point on $\mathcal{M}_{p,k,n}$ from a descent direction (vector in $\mathcal{H}_{\bar{\theta}}$) one needs a retraction, *i.e.*, an operator $R_\theta^{\mathcal{M}_{p,k,n}} : T_\theta\mathcal{M}_{p,k,n} \to \mathcal{M}_{p,k,n}$ which maps tangent vectors onto the manifold. Chapter 2 Section 2.2 presents the notion of retractions as well as their use in optimization algorithms. In the same chapter, Section 2.3 introduces the required properties by a retraction to be valid on a quotient manifold. These properties are briefly recalled in the following. Such retraction on $\mathcal{M}_{p,k,n}$ can be obtained by a retraction on $\overline{\mathcal{M}}_{p,k,n}$ (denoted $\overline{R}_{\bar{\theta}}^{\overline{\mathcal{M}}_{p,k,n}} : T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n} \to \overline{\mathcal{M}}_{p,k,n}$) using the relation $R_\theta^{\mathcal{M}_{p,k,n}}(\xi) = \pi(\overline{R}_{\bar{\theta}}^{\overline{\mathcal{M}}_{p,k,n}}(\bar{\xi}))$. This requires two conditions

1. $\overline{R}_{\bar{\theta}}^{\overline{\mathcal{M}}_{p,k,n}}$ is a proper retraction on $\overline{\mathcal{M}}_{p,k,n}$: $\forall\ \bar{\theta} \in \overline{\mathcal{M}}_{p,k,n}$ and $\bar{\xi} \in T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$, $\overline{R}_{\bar{\theta}}^{\overline{\mathcal{M}}_{p,k,n}}(0) = \bar{\theta}$ and $\mathrm{D}\,\overline{R}_{\bar{\theta}}^{\overline{\mathcal{M}}_{p,k,n}}(0)[\bar{\xi}] = \bar{\xi}$.
2. The induced retraction on $\mathcal{M}_{p,k,n}$ invariant along the equivalence classes: in our case, this translates into $\pi(\overline{R}_{\bar{\theta}}^{\overline{\mathcal{M}}_{p,k,n}}(\bar{\xi})) = \pi(\overline{R}_{(\boldsymbol{U}\boldsymbol{O},\boldsymbol{\tau})}^{\overline{\mathcal{M}}_{p,k,n}}((\boldsymbol{\xi}_{\boldsymbol{U}}\boldsymbol{O}, \boldsymbol{\xi}_{\boldsymbol{\tau}}))$, for all $\boldsymbol{O} \in \mathcal{U}_k$, $\bar{\theta} = (\boldsymbol{U}, \boldsymbol{\tau}) \in \overline{\mathcal{M}}_{p,k,n}$ and $\bar{\xi} = (\boldsymbol{\xi}_{\boldsymbol{U}}, \boldsymbol{\xi}_{\boldsymbol{\tau}}) \in T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$.

Notice that the notion of retraction does not depend on the choice of the metric, so several options are generally available. In this Chapter, we consider

the following retraction from classical results on $\mathrm{St}_{p,k}$ [85] and $(\mathbb{R}_*^+)^n$. This retraction defined on $\overline{\mathcal{M}}_{p,k,n}$ for all $\bar{\theta} = (\boldsymbol{U}, \boldsymbol{\tau}) \in \overline{\mathcal{M}}_{p,k,n}$ and $\bar{\xi} = (\boldsymbol{\xi}_{\boldsymbol{U}}, \boldsymbol{\xi}_{\boldsymbol{\tau}}) \in \mathcal{H}_{\bar{\theta}}$ as

$$\overline{R}_{\bar{\theta}}^{\overline{\mathcal{M}}_{p,k,n}}(\bar{\xi}) = \left(\boldsymbol{X}\boldsymbol{Y}^H, \boldsymbol{\tau} + \boldsymbol{\xi}_{\boldsymbol{\tau}} + \frac{1}{2}\boldsymbol{\tau}^{\odot -1}\boldsymbol{\xi}_{\boldsymbol{\tau}}^{\odot 2} \right), \tag{4.16}$$

where $\boldsymbol{U} + \boldsymbol{\xi}_{\boldsymbol{U}} = \boldsymbol{X}\boldsymbol{\Sigma}\boldsymbol{Y}^H$ is the thin SVD. Notice that for the part that concerns $\boldsymbol{\tau}$, we have a second degree polynom in $\boldsymbol{\xi}_{\boldsymbol{\tau}}$ with a negative discriminant, thus the resulting vector contains strictly positive numbers. It can be checked that the two conditions are satisfied, and this option was chosen for its numerical stability.

4.2.3 . Product metric: geometry for distances

Riemannian distances can be used either for performance assessment, or in machine learning algorithms (*e.g.* for clustering). Their interest can notably be their natural invariances with respect to the manifold and/or metric of interest. These distances are obtained by measuring the length of geodesics, which generalize straight lines onto manifolds while taking into account the curvature induced by the metric and geometric constraints. Unfortunately the Riemannian distance induced by the Fisher information metric of Proposition 21 cannot be obtained in closed-form. To overcome this difficulty, we propose to use a product metric from the following definition.

Definition 43 (Product metric)**.** *The Riemannian metric* $\langle \cdot, \cdot \rangle^{\overline{\mathcal{M}}_{p,k,n}}$ *is defined, for all* $\bar{\theta} = (\boldsymbol{U}, \boldsymbol{\tau}) \in \overline{\mathcal{M}}_{p,k,n}$, $\bar{\xi} = (\boldsymbol{\xi}_{\boldsymbol{U}}, \boldsymbol{\xi}_{\boldsymbol{\tau}})$ *and* $\bar{\eta} = (\boldsymbol{\eta}_{\boldsymbol{U}}, \boldsymbol{\eta}_{\boldsymbol{\tau}}) \in T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$, *as*

$$\langle \bar{\xi}, \bar{\eta} \rangle_{\bar{\theta}}^{\overline{\mathcal{M}}_{p,k,n}} = \alpha \mathfrak{Re}\left(\mathrm{Tr}\left(\boldsymbol{\xi}_{\boldsymbol{U}}^H \boldsymbol{\eta}_{\boldsymbol{U}} \right) \right) + \beta \left(\boldsymbol{\xi}_{\boldsymbol{\tau}} \odot \boldsymbol{\tau}^{\odot -1} \right)^T \left(\boldsymbol{\eta}_{\boldsymbol{\tau}} \odot \boldsymbol{\tau}^{\odot -1} \right), \tag{4.17}$$

where $\alpha, \beta > 0$.

Notice that the product metric has a structure similar to the Fisher information metric in Proposition 21: it consists in a scaled combination of standard metrics on $\mathrm{Gr}_{p,k}$ [2, 1, 52] and $(\mathbb{R}_*^+)^n$ [18]. The main difference is that the weights α and β remain constant in the product metric, which will yield a geometry from well-known results. Another particular interest is that the flexibility regarding this factors allows emphasizing a parameter (subspace spanned by $\boldsymbol{U}$ or textures $\boldsymbol{\tau}$) in the considered geometry. This is notably interesting for clustering applications (see Section 4.4) where we want to control the importance of each feature.

First, one can check that the horizontal space at $\bar{\theta}$ in $\overline{\mathcal{M}}_{p,k,n}$ for the Riemannian metric in Definition 43 is the same as the one given in (4.14) corresponding to the Fisher information metric of Proposition 21. It is thus also denoted $\mathcal{H}_{\bar{\theta}}$ in the following.

Second, we can deduce several geometric tools from classical results about $\mathsf{Gr}_{p,k}$ [2, 1, 52] and $(\mathbb{R}_*^+)^n$ [18] presented in Chapter 2 Section 2.4. The squared Riemannian distance between $\theta_1 = \pi(\bar{\theta}_1)$ and $\theta_2 = \pi(\bar{\theta}_2)$ in $\mathcal{M}_{p,k,n}$ is given by

$$d^2_{\mathcal{M}_{p,k,n}}(\theta_1, \theta_2) = \alpha d^2_{\mathsf{Gr}_{p,k}}(\boldsymbol{U}_1, \boldsymbol{U}_2) + \beta d^2_{(\mathbb{R}_*^+)^n}(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2), \tag{4.18}$$

where $d^2_{\mathsf{Gr}_{p,k}}$ and $d^2_{(\mathbb{R}_*^+)^n}$ are the squared Riemannian distances of $\mathsf{Gr}_{p,k}$ and $(\mathbb{R}_*^+)^n$, respectively. They are defined as

$$\begin{aligned} d^2_{\mathsf{Gr}_{p,k}}(\boldsymbol{U}_1, \boldsymbol{U}_2) &= \|\boldsymbol{\Theta}\|_2^2, \\ d^2_{(\mathbb{R}_*^+)^n}(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2) &= \|\log(\boldsymbol{\tau}_1) - \log(\boldsymbol{\tau}_2)\|_2^2, \end{aligned} \tag{4.19}$$

where $\boldsymbol{\Theta}$ is obtained from the SVD $\boldsymbol{U}_1^H \boldsymbol{U}_2 = \boldsymbol{O}_1 \cos(\boldsymbol{\Theta}) \boldsymbol{O}_2^H$. An additional tool linked to the Riemannian distance is the Riemannian logarithm mapping. Given a reference point $\theta_1 = \pi(\bar{\theta}_1)$ and a second point $\theta_2 = \pi(\bar{\theta}_2)$ both in $\mathcal{M}_{p,k,n}$, the Riemannian logarithm mapping is an operator that provides a vector of $T_{\theta_1}\mathcal{M}_{p,k,n}$ that points towards θ_2 and whose squared norm with respect to the metric from Definition 43 is $d^2_{\mathcal{M}_{p,k,n}}(\theta_1, \theta_2)$ (as defined in (4.18)). Here, the representation in $\mathcal{H}_{\bar{\theta}_1}$ of the Riemannian logarithm mapping $\log_{\theta_1}^{\mathcal{M}_{p,k,n}}(\theta_2)$ on $\mathcal{M}_{p,k,n}$ is

$$\begin{aligned} &\log_{\bar{\theta}_1}(\bar{\theta}_2) = \left(\log_{\boldsymbol{U}_1}^{\mathsf{Gr}_{p,k}}(\boldsymbol{U}_2),\ \log_{\boldsymbol{\tau}_1}^{(\mathbb{R}_*^+)^n}(\boldsymbol{\tau}_2) \right), \\ &\log_{\boldsymbol{U}_1}^{\mathsf{Gr}_{p,k}}(\boldsymbol{U}_2) = \boldsymbol{X} \boldsymbol{\Theta} \boldsymbol{Y}^H, \\ &\log_{\boldsymbol{\tau}_1}^{(\mathbb{R}_*^+)^n}(\boldsymbol{\tau}_2) = \boldsymbol{\tau}_1 \odot \log(\boldsymbol{\tau}_1^{\odot -1} \odot \boldsymbol{\tau}_2), \end{aligned} \tag{4.20}$$

where $\boldsymbol{X} \boldsymbol{\Theta} \boldsymbol{Y}^H$ is defined through the SVD $(\boldsymbol{I}_p - \boldsymbol{U}_1 \boldsymbol{U}_1^H) \boldsymbol{U}_2 (\boldsymbol{U}_1^H \boldsymbol{U}_2)^{-1} = \boldsymbol{X} \tan(\boldsymbol{\Theta}) \boldsymbol{Y}^H$. Conversely, the inverse of this application is the Riemannian exponential mapping $\exp_{\theta}^{\mathcal{M}_{p,k,n}}(\xi)$ on $\mathcal{M}_{p,k,n}$, whose representation in $\overline{\mathcal{M}}_{p,k,n}$ is given by

$$\begin{aligned} &\exp_{\bar{\theta}}(\bar{\xi}) = \left(\exp_{\boldsymbol{U}}^{\mathsf{Gr}_{p,k}}(\boldsymbol{\xi}_{\boldsymbol{U}}),\ \exp_{\boldsymbol{\tau}}^{(\mathbb{R}_*^+)^n}(\boldsymbol{\xi}_{\boldsymbol{\tau}}) \right), \\ &\exp_{\boldsymbol{U}}^{\mathsf{Gr}_{p,k}}(\boldsymbol{\xi}_{\boldsymbol{U}}) = \boldsymbol{U}\, \boldsymbol{Y}\, \cos(\boldsymbol{\Sigma}) + \boldsymbol{X}\, \sin(\boldsymbol{\Sigma}), \\ &\exp_{\boldsymbol{\tau}}^{(\mathbb{R}_*^+)^n}(\boldsymbol{\xi}_{\boldsymbol{\tau}}) = \boldsymbol{\tau} \odot \exp(\boldsymbol{\tau}^{\odot -1} \odot \boldsymbol{\xi}_{\boldsymbol{\tau}}), \end{aligned} \tag{4.21}$$

where $\boldsymbol{\xi}_{\boldsymbol{U}} = \boldsymbol{X} \boldsymbol{\Sigma} \boldsymbol{Y}^H$ is the SVD such that $\boldsymbol{X} \in \mathbb{C}^{p \times k}$ and $\boldsymbol{\Sigma},\ \boldsymbol{Y} \in \mathbb{C}^{k \times k}$. These operators provide mappings between the manifold and its tangent space, which will notably be instrumental in in Section 4.3.2 to define an estimation error vector, and in Section 4.4 in order to define Riemannian means.

4.3 . Estimation and intrinsic Cramér-Rao bounds

4.3.1 . Maximum Likelihood Estimation with Riemannian optimization

In this section, we cast the MLE as an optimization problem on $\mathcal{M}_{p,k,n}$

$$\underset{\theta \in \mathcal{M}_{p,k,n}}{\text{minimize}}\ \mathcal{L}(\theta), \tag{4.22}$$

where $\mathcal{L} : \mathcal{M}_{p,k,n} \to \mathbb{R}$ is the negative log-likelihood defined in (4.5). To solve this estimation problem, a block coordinate descent (BCD) has been proposed in [28]. Here, we present an alternative algorithm leveraging the information geometry presented in Section 4.2.2.

A first alternative is to use a Riemannian gradient descent (RGD) [1]. An iteration of this algorithm consists in computing the gradient of $\mathcal{L}$ and then retracting minus the gradient multiplied by a step size. Given the iterate $\theta^{(l)}$ represented by $\bar{\theta}^{(l)}$, the RGD algorithm yields

$$\bar{\theta}^{(l+1)} = \overline{R}_{\bar{\theta}^{(l)}}^{\overline{\mathcal{M}}_{p,k,n}} \left(-\nu_t \operatorname{grad}_{\overline{\mathcal{M}}_{p,k,n}} \overline{\mathcal{L}}(\bar{\theta}^{(l)}) \right), \tag{4.23}$$

where ν_t is a step size, $\operatorname{grad}_{\overline{\mathcal{M}}_{p,k,n}} \overline{\mathcal{L}}(\bar{\theta}^{(l)})$ is a representative of the Riemannian gradient associated to the Fisher information metric of Proposition 23, and $\overline{R}_{\bar{\theta}^{(l)}}^{\overline{\mathcal{M}}_{p,k,n}}$ is the retraction defined in (4.16). Hence, it also corresponds to the so-called natural gradient as defined in [4], which regained interest due to its link with second order optimization methods [87].

Here, we propose a more flexible approach following the recent works [16, 67]: we derive a Riemannian stochastic gradient descent (R-SGD) on $\mathcal{M}_{p,k,n}$. The R-SGD is a Riemannian optimization algorithm that computes the gradient of the function to minimize only on a subset A of all measured signals $\{\boldsymbol{x}_i\}_{i=1}^n$. Hence, contrary to the BCD or the RGD, this algorithm can be used on large scale datasets and the cost of an iteration can be modulated according to the computing capacity. Since the number of samples A can be chosen arbitrarily set, this algorithm also encompasses the "plain" R-SGD ($A = \{\boldsymbol{x}_i\}$) and the classical RGD [1] ($A = \{\boldsymbol{x}_i\}_{i=1}^n$). Additionally, the R-SGD will be shown to have a lower complexity (per iteration) than the BCD.

In order to derive the R-SGD, the negative log-likelihood $\mathcal{L}$ defined on $\mathcal{M}_{p,k,n}$ is rewritten

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} \mathcal{L}_i(\theta), \tag{4.24}$$

where $\mathcal{L}_i$ is the negative log-likelihood defined on the sample $\boldsymbol{x}_i$. Hence, the same notation applies to the negative log-likelihood (4.5) defined on $\overline{\mathcal{M}}_{p,k,n}$:

$\overline{\mathcal{L}}(\bar{\theta}) = \sum_{i=1}^{n} \overline{\mathcal{L}}_i(\bar{\theta})$. In short, given the actual iterate $\theta^{(l)}$, an iteration of R-SGD proceeds in three steps:

1. a set A of samples is randomly drawn from $\{\boldsymbol{x}_i\}_{i=1}^n$,
2. then the gradient of $\sum_{x_i \in A} \mathcal{L}_i(\theta^{(l)})$ is computed,
3. finally a new iterate is given by retracting minus the gradient times a step size.

Since a retraction on $\mathcal{M}_{p,k,n}$ is provided in Section 4.2.2, the only remaining element to provide is the Riemannian gradient of $\mathcal{L}_i(\theta)$. This gradient is given in the following proposition:

Proposition 22 (Riemannian gradient)**.** *Given $\theta = \pi(\boldsymbol{U}, \boldsymbol{\tau}) \in \mathcal{M}_{p,k,n}$ represented by $\bar{\theta} = (\boldsymbol{U}, \boldsymbol{\tau}) \in \overline{\mathcal{M}}_{p,k,n}$, the representative in $\mathcal{H}_{\boldsymbol{U}} \times T_{\boldsymbol{\tau}}(\mathbb{R}^{++})^n$ of the Riemannian gradient of $\mathcal{L}_i$ at θ is*

$$\operatorname{grad}_{\overline{\mathcal{M}}_{p,k,n}} \overline{\mathcal{L}}_i(\bar{\theta}) = (\boldsymbol{G}_{\boldsymbol{U}}, \boldsymbol{G}_{\boldsymbol{\tau}})$$

where

$$\boldsymbol{G}_{\boldsymbol{U}} = -\frac{\tau_i}{n c_{\boldsymbol{\tau}} (1+\tau_i)} (\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^H) \boldsymbol{x}_i \boldsymbol{x}_i^H \boldsymbol{U},$$

and the j^{th} element of $\boldsymbol{G}_{\boldsymbol{\tau}}$ is

$$(\boldsymbol{G}_{\boldsymbol{\tau}})_j = \begin{cases} 1 + \tau_i - \frac{1}{k} \boldsymbol{x}_i^H \boldsymbol{U}\boldsymbol{U}^H \boldsymbol{x}_i & for \quad j = i \\ 0 \quad otherwise. \end{cases}$$

Proof. See Appendix 4.A.2. □

Following from this gradient, the resulting R-SGD on $\mathcal{M}_{p,k,n}$ is detailed in the box Algorithm 9. Concerning the computation of the step size, several options exist. When the gradient is computed on all data, *i.e.* $A = \{\boldsymbol{x}_i\}_{i=1}^n$, a line search (*e.g.* [1, §4.2]) is recommended. When the gradient is computed on a subset of all data, a step size proportional to $1/t$, where t is the number of iterations, can be used as in [4].

By rearranging the operations of $\boldsymbol{G}_{\boldsymbol{U}}$ in Proposition 22, the computational complexity of the gradient of $\sum_{x_i \in A} \mathcal{L}_i(\theta)$ is $\mathcal{O}(mpk+n)$, where m the number of samples in A. In practice, $c_{\boldsymbol{\tau}}$ can be approximated using only the textures associated with the samples in A, *i.e.* $c_{\boldsymbol{\tau}} \approx \frac{1}{m} \sum_{x_i \in A} \frac{\tau_i^2}{1+\tau_i}$. Hence, the complexity of the gradient becomes $\mathcal{O}(mpk)$. Then, the complexity of the retraction (4.16) is $\mathcal{O}(pk^2 + m)$, as we only retract the non-zero elements of the gradient $\boldsymbol{G}_{\boldsymbol{\tau}}$ from Proposition 22. Hence, the total complexity of each iteration of Algorithm 9 is $\mathcal{O}(mpk + pk^2)$, which is much lower than the $\mathcal{O}(np^2 + p^3)$ of the BCD in [28] (which involves the SVD of the scaled SCM at each step).

Algorithm 9: Riemannian stochastic gradient descent

Input : Initial iterate $\bar{\theta}^{(0)} \in \overline{\mathcal{M}}_{p,k,n}$.
Output: Sequence of iterates $\{\bar{\theta}^{(l)}\}$.
for $l = 0$ *to convergence* **do**
  Randomly draw a subset $A \subset \{\boldsymbol{x}_i\}_{i=1}^n$ and set
  $\bar{\xi}^{(l)} = \sum_{x_i \in A} \operatorname{grad}_{\overline{\mathcal{M}}_{p,k,n}} \overline{\mathcal{L}}_i(\bar{\theta}^{(l)})$
  Compute a step size ν_l and set
  $\bar{\theta}^{(l+1)} = \overline{R}_{\bar{\theta}^{(l)}}^{\overline{\mathcal{M}}_{p,k,n}}(-\nu_l \bar{\xi}^{(l)})$
end

4.3.2 . Intrinsic Cramér-Rao bounds

Obtaining performance bounds for the model in (4.3) is a complex issue, notably because the signal subspace is represented by a point in $\mathrm{Gr}_{p,k}$. A first approach was proposed in [13] for the model $\boldsymbol{x}_i \sim \mathbb{C}\mathcal{N}(\boldsymbol{0}, \tau_i \boldsymbol{G}\boldsymbol{G}^H + \boldsymbol{I})$, where $\boldsymbol{G} \in \mathbb{C}^{p \times k}$ is a lower-triangular matrix with positive diagonal elements. Such parameterization is carefully chosen in order to obtain a minimal and essentially unconstrained parametrization of the low-rank signal covariance matrix. This allows obtaining the standard Cramér-Rao inequality for the parameter $\boldsymbol{g} = \mathrm{vec}(\boldsymbol{G})$.

$$\mathrm{CRB}(\boldsymbol{\pi}) = \frac{\partial \boldsymbol{\pi}}{\partial \mathbf{g}^T} \mathrm{CRB}(\mathbf{g}) \frac{\partial \boldsymbol{\pi}^T}{\partial \mathbf{g}} \;\Rightarrow\; \mathbb{E}\left[||\boldsymbol{\Pi} - \hat{\boldsymbol{\Pi}}||_F^2 \right] \geq \mathrm{Tr}\left\{ \mathrm{CRB}(\boldsymbol{\pi}) \right\} \tag{4.25}$$

thus enabling to assess approximately the minimum distance between the estimated and the true signal subspace. Another option could have been to start with the constrained parameterization $\boldsymbol{G} = \boldsymbol{U}\boldsymbol{D}^{1/2}$ and to directly handle the orthonormality constraints $\boldsymbol{U}^H\boldsymbol{U} = \boldsymbol{I}_k$ with the the theory of constrained CRBs [60, 123, 95, 99] to obtain $\mathrm{CRB}(\mathrm{vec}(\boldsymbol{U}))$, then deriving the same result as in (4.25) from $\boldsymbol{\pi} = \mathrm{vec}(\boldsymbol{U}\boldsymbol{U}^H)$. This method is expected to yield the same result as in [13] from a different path of derivations.

While the obtained inequality in (4.25) allows for an analysis with numerical experiments, it still lacks some interpretable closed-form. In the following, we will directly treat the signal subspace as a point in $\mathrm{Gr}_{p,k}$[1] and rely on the intrinsic CRB theory from [121, 20]. The interest is twofold: first it will yield a simple and interpretable closed form for the bound on the subspace estimation. Second, this bound will be obtained for natural distance on $\mathrm{Gr}_{p,k}$

[1]Note that we consider the case of equal eigenvalues, but this restriction has been carefully motivated in the model introduction section. The extension to the general case could be considered using recent derivations from [17] but this complex issue goes far beyond the scope of this chapter.

in (4.19), which is expected to better reflect breakdown points at low sample support (cf. [121] for an example regarding covariance matrix estimation).

Intrinsic (*i.e.*, manifold oriented) versions of the Cramér-Rao inequality have been established [121] and extended to quotient manifolds in [20]. They are presented in details in Chapter 2 Section 2.5. The main difference compared to the classical CRBs is that the parameter θ is treated as being in a Riemmanian manifold endowed by an arbitrary chosen "error" metric. The estimation error is thus measured using the Riemannian distance d that emanated from this error metric. The obtained inequality is of the form

$$\boldsymbol{C} \succeq \boldsymbol{F}^{-1} + \text{ curvature terms}, \tag{4.26}$$

where $\boldsymbol{C}$ is the covariance matrix of the error vector (defined as the Riemannian logarithm mapping $\log_\theta(\hat{\theta})$, which is induced by the error metric), and $\boldsymbol{F}^{-1}$ is the inverse of the Fisher information matrix (which depends on both the chosen metric and the Fisher information metric). Neglecting the curvature terms and taking the trace of (4.26) yields the inequality $\mathbb{E}\left[d^2(\theta,\hat{\theta})\right] \geq \mathrm{Tr}(\boldsymbol{F}^{-1})$ for an unbiased estimator $\hat{\theta}$, which will be here our primary interest.

In our context, we consider $\mathcal{M}_{p,k,n}$ endowed with the product metric from Definition 43 in order to bound the error measure defined by $d^2_{\mathcal{M}_{p,k,n}}$ as in (4.18). For the sake of exposition, the obtained results are directly reported in the two following propositions, while the technical details are let in the Appendix 4.A.3.

Proposition 23 (Fisher information matrix)**.** *The Fisher information matrix $\boldsymbol{F}_\theta$ on $\mathcal{M}_{p,k,n}$ admits the structure*

$$\boldsymbol{F}_\theta = \begin{pmatrix} \boldsymbol{F}_{\boldsymbol{U}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{F}_{\boldsymbol{\tau}} \end{pmatrix},$$

with the blocks $\boldsymbol{F}_{\boldsymbol{U}} = 2\,\alpha^{-1}\,n\,c_{\boldsymbol{\tau}}\,\boldsymbol{I}_{2(p-k)k}$, and $\boldsymbol{F}_{\boldsymbol{\tau}} = \beta^{-1}\,k\,\mathrm{diag}\left(\boldsymbol{\tau}^{\odot 2} \odot (\boldsymbol{1}+\boldsymbol{\tau})^{\odot -2}\right)$, and where $\mathrm{diag}(\cdot)$ returns the diagonal matrix formed with the elements of its argument.

Proof. See Appendix 4.A.3. □

Proposition 24 (iCRB)**.** *Let $\{\boldsymbol{x}_i\}_{i=1}^n$ be a sample set following the model in* (4.3)*. Let $\hat{\theta}$ be an estimate of $\theta \in \mathcal{M}_{p,k,n}$ for the model. The estimation error defined by $d^2_{\mathcal{M}_{p,k,n}}$ as in* (4.18) *is bounded as*

$$\mathbb{E}[d^2_{\mathcal{M}_{p,k,n}}(\hat{\theta},\theta)] \geq \alpha\,\mathrm{CRB}_{\boldsymbol{U}} + \beta\,\mathrm{CRB}_{\boldsymbol{\tau}}. \tag{4.27}$$

where

$$\mathsf{CRB}_{\boldsymbol{U}} = \frac{(p-k)\,k}{n\,c_{\boldsymbol{\tau}}} \quad \text{and} \quad \mathsf{CRB}_{\boldsymbol{\tau}} = \frac{1}{k}\sum_{i=1}^{n}\frac{(1+\tau_i)^2}{\tau_i^2}.$$

Furthermore, two iCRB, on $\mathsf{Gr}_{p,k}$ *and* $(\mathbb{R}_*^+)^n$ *respectively, are given by*

$$\mathbb{E}[d^2_{\mathsf{Gr}_{p,k}}(\pi(\hat{\boldsymbol{U}}),\pi(\boldsymbol{U}))] \geq \mathsf{CRB}_{\boldsymbol{U}}, \tag{4.28}$$

$$\mathbb{E}[d^2_{(\mathbb{R}_*^+)^n}(\hat{\boldsymbol{\tau}},\boldsymbol{\tau})] \geq \mathsf{CRB}_{\boldsymbol{\tau}}. \tag{4.29}$$

Proof. See Appendix 4.A.3. □

Notice that the problem of estimating a subspace should not depend on its basis $\boldsymbol{U}$, as two estimates $\hat{\boldsymbol{U}}$ and $\hat{\boldsymbol{U}}\boldsymbol{Q}$ yield the same subspace estimate (but would yield different MSEs for the basis $\boldsymbol{U}$). The obtained bound on $d^2_{\mathsf{Gr}_{p,k}}$ satisfies this property. Furthermore, Proposition 24 shows that the subspace estimation problem for model (4.3) does not depend on the underlying subspace itself, but rather only on its dimension and the SNR, which is theoretically appealing. Conversely, the euclidean CRBs in [13], bounding the MSE on $\boldsymbol{U}\boldsymbol{U}^H$ (orthogonal projector) as in (4.25) does not exhibit such direct interpretability. Finally, in the specific case of data following a Gaussian low-rank (spiked) model for which $\tau_i = \mathsf{SNR}$ so that $\boldsymbol{x}_i \sim \mathbb{C}\mathcal{N}(\boldsymbol{0}, \mathsf{SNR}\times\boldsymbol{U}\boldsymbol{U}^H + \boldsymbol{I}_p)$, we retrieve the iCRB of [121, Eq.145], *i.e.*,

$$\mathbb{E}[d^2_{\mathsf{Gr}_{p,k}}(\pi(\hat{\boldsymbol{U}}),\pi(\boldsymbol{U}))] \geq \frac{(p-k)\,k\,(1+\mathsf{SNR})}{n\,\mathsf{SNR}^2}. \tag{4.30}$$

4.4 . Clustering of subspaces and textures

In this section, we apply the statistical model developed in Section 4.1 with its Riemannian geometry $\mathcal{M}_{p,k,n}$, presented in Section 4.2.3, to clustering problems. More specifically, we assume that we have M batches $\boldsymbol{X}_i$ (*e.g.* sets of local pixels of an image, EEG epochs of measurements, ...). Each $\boldsymbol{X}_i \in \mathbb{C}^{p\times n}$ is a column-wise concatenation of n observations $\boldsymbol{x}_j \in \mathbb{C}^p$ defined in Section 4.1. Furthermore, each batch $\boldsymbol{X}_i$ belongs to an unknown class $y \in [\![1, K]\!]$.

The use of statistical descriptors is a classical procedure in machine learning as they are often more discriminative than raw data (see *e.g.* [8, 134]). Hence, we begin by estimating a descriptor $\theta_i \in \mathcal{M}_{p,k,n}$ of the batch $\boldsymbol{X}_i$ following Section 4.3.1. Then, the aim is to partition the descriptors $\{\theta_i\}_{i=1}^M$

in $S = \{S_1, S_2, \cdots, S_K\}$. Thus, we get a partition of the original batches $\{\boldsymbol{X}_i\}_{i=1}^M$.

Each parameter θ_i is represented by a couple, *i.e.* $\theta_i = \pi(\boldsymbol{U}_i, \boldsymbol{\tau}_i)$. Our contribution is to cluster both components (subspace and power) in a unified manner, leveraging the geometry of $\mathcal{M}_{p,k,n}$ featured in Section 4.2.3. This section is focused on the application of a *K-means++*[7] on $\mathcal{M}_{p,k,n}$ with the tools developed earlier. However, the proposed methodology is flexible: *(i)* descriptors θ_i can be replaced by other statistical estimates with their associated Riemannian geometries, *(ii)* many Euclidean based clustering Algorithms can be transformed to Riemannian ones (replacing distances and means by their Riemannian counterparts).

4.4.1 . Distance and mean computations

Most clustering Algorithms, including *K-means++* [7], rely on distance and mean computations. Since θ_i lies on a Riemannian manifold we first need to define distance and mean computations other than simple Euclidean ones.

A natural choice is the use of the distance $d_{\mathcal{M}_{p,k,n}}$ defined in (4.18). In the context of clustering, the distance on $\mathsf{Gr}_{p,k}$ and the one on $(\mathbb{R}_*^+)^n$ do not necessarily have the same amplitude or the same ability to discriminate. Thus, the parameters α, β of the metric of Definition 43 are to be chosen carefully. We propose a 2-step strategy to select α, β: (*i*) correction of the scale effect and (*ii*) choice of a trade-off between the distances on $\mathsf{Gr}_{p,k}$ and $(\mathbb{R}_*^+)^n$. To correct the scale effect we propose to normalize the squared distances by their mean values on the samples $\{\theta_i\}_{i=1}^M$. Then, a trade-off can be made between the two distances. More precisely, $\forall \gamma \in [0,1]$, we define

$$\begin{aligned} \alpha &= \frac{1-\gamma}{\frac{1}{M^2}\sum_{q,l\in[\![1,M]\!]} d^2_{\mathsf{Gr}_{p,k}}(\boldsymbol{U}_q, \boldsymbol{U}_l)}, \\ \beta &= \frac{\gamma}{\frac{1}{M^2}\sum_{q,l\in[\![1,M]\!]} d^2_{(\mathbb{R}_*^+)^n}(\boldsymbol{\tau}_q, \boldsymbol{\tau}_l)}. \end{aligned} \tag{4.31}$$

It remains to define a mean computation Algorithm on a set of parameters S_j. In [75], the mean of a set of points on a Riemannian manifold is defined as the minimizer of the variance of this set. Let $m = \mathrm{Card}(S_j)$, the variance V of S_j at $\theta = \pi(\bar{\theta}) \in \mathcal{M}_{p,k,n}$ is defined as,

$$V(\bar{\theta}) = \frac{1}{m}\sum_{\theta_i \in S_j} d^2_{\mathcal{M}_{p,k,n}}(\theta, \theta_i). \tag{4.32}$$

The mean $c = \pi(\bar{c}) \in \mathcal{M}_{p,k,n}$ of the set of points S_j is obtained from the minimization of the variance,

$$\bar{c} = \underset{\theta \in \mathcal{M}_{p,k,n}}{\arg\min}\ \frac{1}{2} V(\bar{\theta}). \tag{4.33}$$

Denoting $\bar{c} = (\boldsymbol{U}, \boldsymbol{\tau})$, one can check that the mean $\boldsymbol{\tau}$ corresponding to the distance $d_{(\mathbb{R}_*^+)^n}$ is simply the geometric mean

$$\boldsymbol{\tau} = \left(\prod_{\theta_i \in S_j}^{\odot} \boldsymbol{\tau}_i \right)^{\odot 1/m}, \tag{4.34}$$

where $\prod^{\odot}$ is the elementwise product. Similarly, the mean corresponding to distance $d_{\mathsf{Gr}_{p,k}}$ is well-known [2]. Unfortunately, no closed form is known to compute it. It is obtained through the following Riemannian gradient descent: given $\boldsymbol{U}^{(l)}$, the iterate $\boldsymbol{U}^{(l+1)}$ is

$$\boldsymbol{U}^{(l+1)} = \exp_{\boldsymbol{U}^{(l)}}^{\mathsf{Gr}_{p,k}} \left(\frac{\nu_t}{m} \sum_{\theta_i \in S_j} \log_{\boldsymbol{U}^{(l)}}^{\mathsf{Gr}_{p,k}} (\boldsymbol{U}_i) \right), \tag{4.35}$$

where ν_t is the step size which can be computed thanks to a line search [1].

Since we get one mean per class, in the rest of the Chapter, the mean of S_j is noted c_j.

4.4.2 . *K-means++* on $\mathcal{M}_{p,k,n}$

With the distance and mean computation Algorithms explained above, we use a *K-means++* on $\mathcal{M}_{p,k,n}$ to partition $\{\theta_i\}_{i=1}^M$ in S (and thus partition $\{\boldsymbol{X}_i\}_{i=1}^M$). The *K-means++* Algorithm on a given set endowed with a divergence and a center of mass computation has been presented in the subsection 1.4.2. We briefly recall the main steps of a *K-means++*, here adapted to the Riemannian manifold $\mathcal{M}_{p,k,n}$.

Instead of choosing class centers c_j uniformly at random from $\{\theta_i\}_{i=1}^M$, *K-means++* initializes them by recursively choosing a new center θ_i with probability $\frac{D(\theta_i)^2}{\sum_{\theta_j} D(\theta_j)^2}$ [7]. Here, $D(\theta_i)$ denotes the distance $d_{\mathcal{M}_{p,k,n}}$ from θ_i to the closest center among those already chosen. Once these class centers are initialized, *K-means++* on $\mathcal{M}_{p,k,n}$ iteratively applies two steps [7]:

1. **Assignment step**: each θ_i is assigned to the cluster S_j whose center c_j is the closest using the distance $d_{\mathcal{M}_{p,k,n}}$,

2. **Update step**: each new class center c_j is computed using (4.34) and (4.35).

Once terminated, *K-means++* on $\mathcal{M}_{p,k,n}$ outputs the partition S. Intuitively, *K-means++* finds clusters S_j whose points $\theta_i \in S_j$ are close to each other using the distance $d_{\mathcal{M}_{p,k,n}}$.

4.4.3 . Theoretical properties

We recall the main theoretical property of the *K-means++* Algorithm, presented in the subsection 1.4.2 and here adapted to $\mathcal{M}_{p,k,n}$. To analyze the performance of *K-means++* on $\mathcal{M}_{p,k,n}$, we define the within-cluster sum of squares (WCSS),

$$\phi(S) = \sum_{j=1}^{K} \sum_{\theta_i \in S_j} d^2_{\mathcal{M}_{p,k,n}}(c_j, \theta_i). \tag{4.36}$$

The main property of the *K-means++* Algorithm is its output partition satisfies

$$\mathbb{E}[\phi] \leq 8(\ln K + 2)\phi_{\mathsf{OPT}} \tag{4.37}$$

where the expectation is taken with respect to the seeding procedure and ϕ_{OPT} is a minimum of (4.36). This property is central to *K-means++* since it is proven that a plain *K-means* [80] cannot admit such a bound. However, the clustering from a *K-means++* is not necessarily a global minimum of the WCSS (4.36). Hence, a standard practice is to run the Algorithm several times with different initializations and then to keep the clustering with the lowest inertia (4.36). *K-means++* on $\mathcal{M}_{p,k,n}$ with the strategy of several initializations is presented in Algorithm 10.

4.5 . Numerical experiments

4.5.1 . Simulations

This section illustrates the performance of the Algorithm 9 as well as the Cramér-Rao bounds developed in Section 4.3. The covariance matrix of the simulated data follows the model $\boldsymbol{\Sigma}_i = \boldsymbol{I}_p + \tau_i \boldsymbol{U}\boldsymbol{U}^H$. The basis $\boldsymbol{U}$ is a random matrix in $\mathrm{St}_{p,k}$. The textures τ_i are randomly drawn from a Log-normal$(-\frac{s^2}{2}, s^2)$ multiplied by the desired SNR. Hence, we get $\mathbb{E}[\tau_i] =$ SNR. The shape parameter s^2 controls the heterogeneity of the textures: the higher the s^2, the greater the heterogeneity. We generate sets $\{\boldsymbol{x}_i\}_{i=1}^n$, with $n \in [\![10, 1000]\!]$, from the zero mean complex Gaussian multivariate distribution with covariance $\boldsymbol{\Sigma}_i$. For each value of n, N sets $\{\boldsymbol{x}_i\}_{i=1}^n$ are simulated and estimators $\hat{\boldsymbol{U}}, \hat{\boldsymbol{\tau}}$ are computed in each case.

Here are the considered estimators in the simulations:

1. SCM: the k first principal eigenvectors of the SCM of $\{\boldsymbol{x}_i\}_{i=1}^n$ are concatenated to get $\boldsymbol{U}^{\mathsf{SCM}}$.

2. BCD: the MLE estimate is done using BCD algorithm on $\{\boldsymbol{x}_i\}_{i=1}^n$ [28] . The estimators are denoted $\boldsymbol{U}^{\mathsf{BCD}}$ and $\boldsymbol{\tau}^{\mathsf{BCD}}$.

Algorithm 10: *K-means++* on $\mathcal{M}_{p,k,n}$

Input : A set $\{\theta_i\}_{i=1}^M \subset \mathcal{M}_{p,k,n}$ to partition, a number of clusters K and a number of initializations n_{init}.
Output: Best partition $S^\star$.
$\phi^\star \leftarrow +\infty$
for 1 *to* n_{init} **do**
  # Initialization
  Take one center c_1, chosen uniformly at random from $\{\theta_i\}_{i=1}^M$.
  while $\#\{c_i\} < K$ **do**
    Take a new center c_j, choosing $\theta_i \in \{\theta_i\}_{i=1}^M$ with probability $\dfrac{D(\theta_i)^2}{\sum_{\theta_m} D(\theta_m)^2}$.
  end
  # K-means
  while *no convergence* **do**
    Assignment step: $\forall i \in [\![1, M]\!]$ assign θ_i to the cluster S_j with the nearest c_j, $j \in [\![1, K]\!]$.
    Update step: Calculate new centers c_j of clusters S_j, $\forall j \in [\![1, K]\!]$, using (4.34) and (4.35).
  end
  Compute $\phi(S)$ with (4.36).
  if $\phi(S) < \phi^\star$ **then**
    $S^\star \leftarrow S$
    $\phi^\star \leftarrow \phi(S)$
  end
end

3. RGD: Algorithm 9 is performed using all samples at each iteration, *i.e.* $A = \{\boldsymbol{x}_i\}_{i=1}^n$. Pymanopt library [132] (builds upon the Manopt library [24]) achieves this optimization. The estimators are denoted $\boldsymbol{U}^{\text{RGD}}$ and $\boldsymbol{\tau}^{\text{RGD}}$.

To measure the subspace estimation performance of the considered estimators, we compute the mean squared error (MSE) between the estimators $\hat{\boldsymbol{U}} \in \{\boldsymbol{U}^{\text{SCM}}, \boldsymbol{U}^{\text{BCD}}, \boldsymbol{U}^{\text{RGD}}\}$ and the real parameter $\boldsymbol{U}$. We compute the MSE as the mean squared distance on $\text{Gr}_{p,k}$ (4.19) between estimated parameters $\hat{\boldsymbol{U}}$ and real parameter $\boldsymbol{U}$. Texture estimation performance is also assessed. The MSE is computed between the estimators $\hat{\boldsymbol{\tau}} \in \{\boldsymbol{\tau}^{\text{BCD}}, \boldsymbol{\tau}^{\text{RGD}}\}$ and real parameter $\boldsymbol{\tau}$ as the mean squared distance on $(\mathbb{R}_*^+)^n$ (4.19).

The subspace estimation performance is studied for two different s^2 along two SNR in Figure 4.2. Firstly, we observe that our proposed estimation

algorithm performs identically to the block coordinate Algorithm [28] in every scenario. Also, both estimators are statistically efficient, *i.e.* reach the lower bound (4.28) when n is sufficiently large. Finally, in the case of a low SNR (*i.e.*, SNR $= 1$), the block coordinate descent and our Riemannian gradient descent outperform the projected SCM regardless of texture heterogeneity.

Figure 4.3 presents the texture estimation error as a function of SNR with two different s^2. Firstly, our proposed estimation algorithm performs identically to the block coordinate Algorithm [28]. Interestingly, the rate of convergence of the estimation error in the case of low heterogeneity, *i.e.* $s^2 = 2$, is much faster than in the case of high heterogeneity, *i.e.* $s^2 = 4$. Moreover, both estimators reach the lower bound (4.29) for a high enough SNR.

A final simulation is conducted on high dimensional data. In Section 4.3, we recalled that the complexity of the BCD grows linearly with the number of data n and quadratically with the dimension p of the data. Hence, the BCD is no longer practicable when both n and p get large. However, in Section 4.3, we showed that the R-SGD proposed in Algorithm 9 has a constant complexity for the number of data and linear for the dimension of the data. Figure 4.4 illustrates this situation with $n \in [\![10^3, 10^4]\!]$, $p = 10^4$ and $k = 10$ (dimensions for which the iteration of BCD cannot be computed on the tested setup). This shows the efficiency of the proposed R-SGD.

4.5.2 . Clustering: application to image segmentation

To illustrate the interest of the Riemannian geometry $\mathcal{M}_{p,k,n}$ and of the parameters of the statistical model (4.3) used as descriptors, we apply the Algorithm 10 to the hyperspectral image segmentation problem Indian Pines presented in the Chapter 1. Figure 4.5 shows the ground truth with the 16 classes.

After centering the image by subtracting the global mean, a sliding window of size $w \times w$ is applied to the image. One descriptor θ_i is estimated using the $n = w^2$ observations in each window denoted $\boldsymbol{X}_i \in \mathbb{R}^{p \times n}$. Thus, we get a set of descriptors $\{\theta_i\}$ to cluster using a *K-means++* [7].

We compare the descriptors of the considered statistical model (MSG+WGN) with different descriptors and geometries. Due to the data's high dimensionality, some methods require a PCA on the whole image as a preprocessing. Then, we keep only the k first components. We begin by presenting these different methods:

1. "center pixel": we extract the center vector of the window. *K-means++* cluster these pixels using the Euclidean metric (*i.e.*, classical inner product). It amounts to cluster directly the image using a classical *K-means++*.

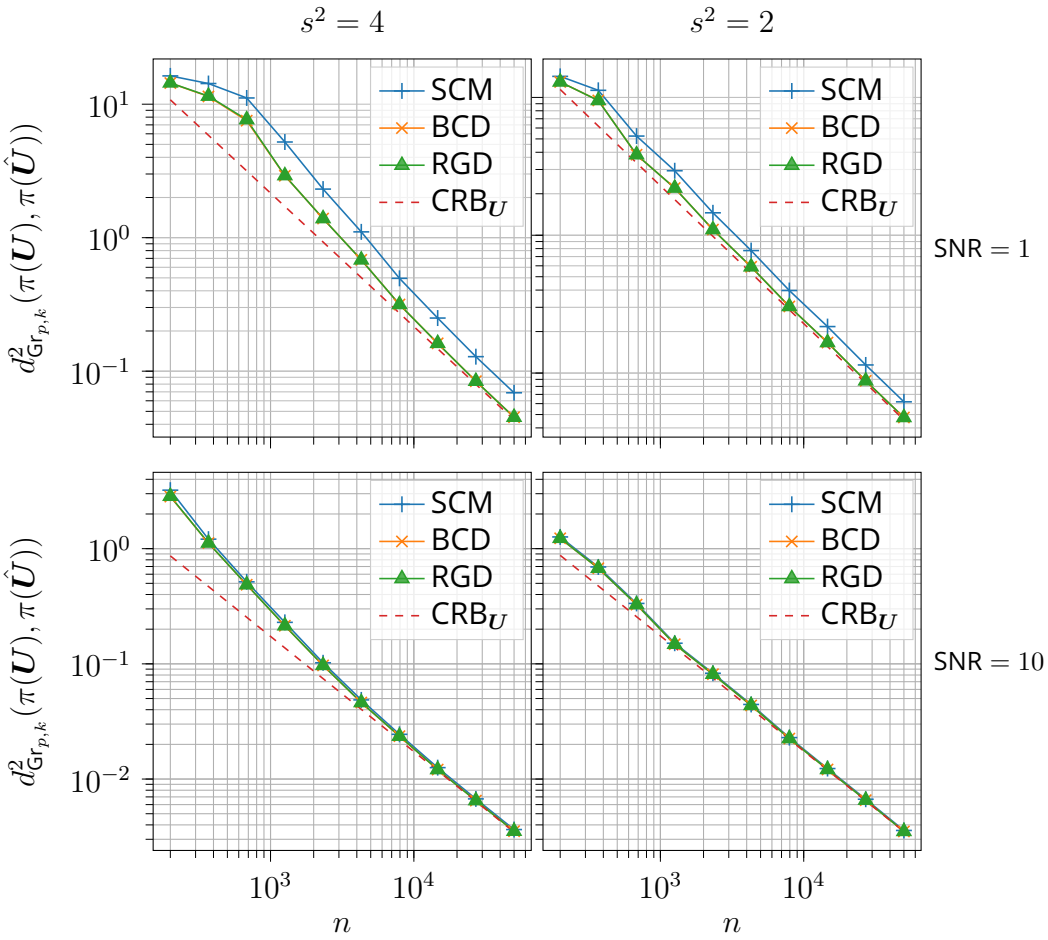


Figure 4.2: MSE over $N = 100$ simulated sets $\{\boldsymbol{x}_i\}_{i=1}^n$ ($p = 100$ and $k = 20$) with respect to the number of samples n for the three considered estimators. The textures are generated with $s^2 = 4$ (left part), $s^2 = 2$ (right part), SNR $= 1$ (upper part), SNR $= 10$ (lower part).

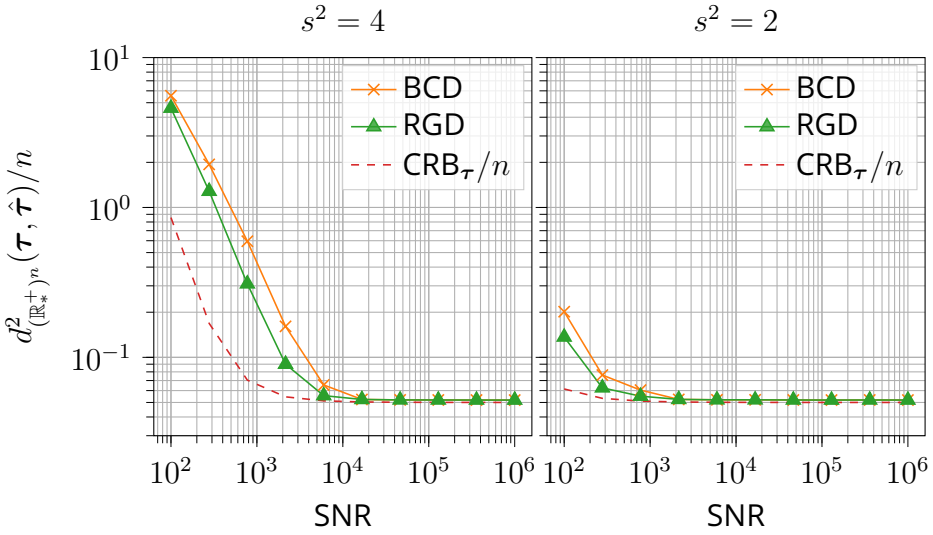


Figure 4.3: MSE over $N = 100$ simulated sets $\{\boldsymbol{x}_i\}_{i=1}^n$ ($n = 10^4$, $p = 100$ and $k = 20$) with respect to the SNR for the BCD and RGD estimators. The textures are generated with $s^2 = 4$ (left) and $s^2 = 2$ (right).

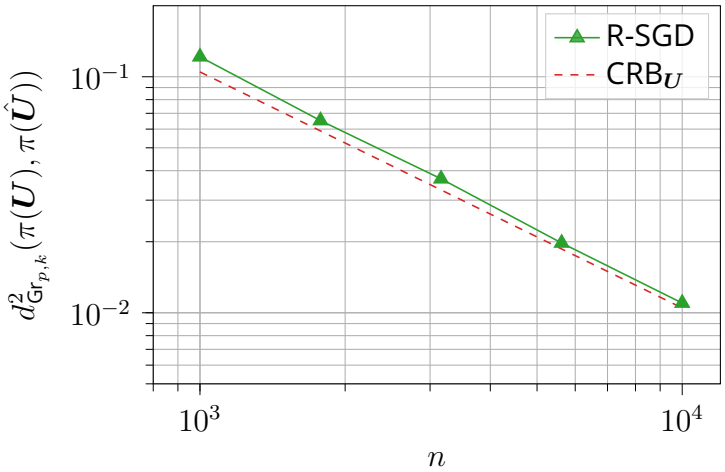


Figure 4.4: MSE over $N = 100$ simulated sets $\{\boldsymbol{x}_i\}_{i=1}^n$ ($p = 10^4$ and $k = 10$) with respect to the number of samples n for the R-SGD estimator. 150 samples are used for each computation of the gradient. The textures are generated with $s^2 = 2$ and $\mathrm{SNR} = 10^3$.

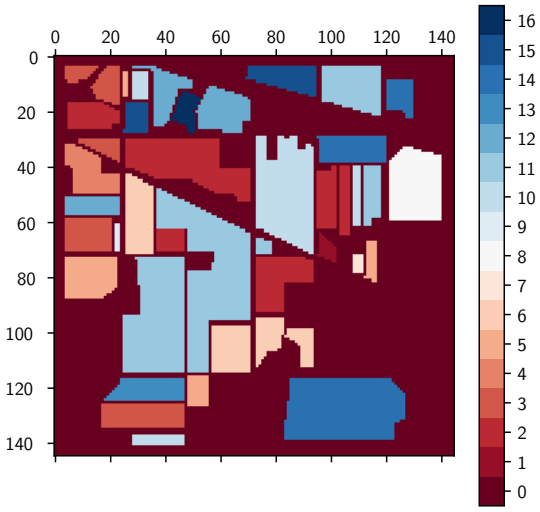


Figure 4.5: Ground truth of the image *Indian Pines* [9]. The background (no class available) is represented by the class 0.

2. "mean pixel": we average the pixels inside the window. Then *K-means++* cluster these means using the Euclidean metric.

3. "SCM": we compute the SCM using pixels inside the window. *K-means++* cluster these matrices using the Riemannian geometry of symmetric positive definite matrices $\mathcal{S}_p^{++}$ (see [120, 14, 113]).

Next, we present the different methods that take into account this high dimensionality. Therefore, we do not use any dimensional reduction preprocessing.

1. "subspace SCM": the k first eigenvectors of the SCM are retained. Then, they are clustered using a *K-means++* on $\mathrm{Gr}_{p,k}$.

2. "robust subspace $\gamma = 0$": our method. Subspaces and textures are estimated following statistical model (4.3). Only the subspaces are clustered using a *K-means++* on $\mathrm{Gr}_{p,k}$. σ^2 is pre-estimated using the $p - k$ lowest eigenvalues of the SCM.

3. "robust subspace $\gamma > 0$": our method. Subspaces and textures are estimated following statistical model (4.3). The textures and subspaces are clustered using a *K-means++* on $\mathcal{M}_{p,k,n}$ as explained in Section 4.4 and detailed in Algorithm 10. σ^2 is pre-estimated using the $p - k$ lowest eigenvalues of the SCM.

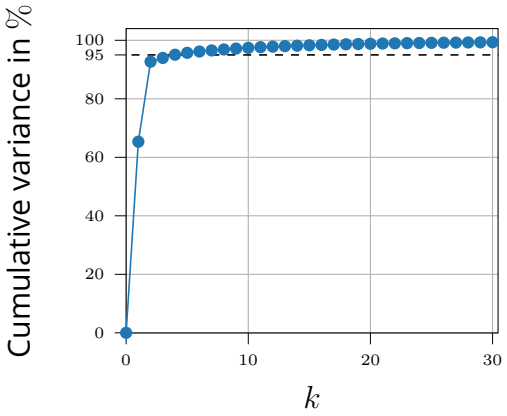


Figure 4.6: Cumulative variance, *i.e.* $\sum_{i=1}^{k} \lambda_i / \sum_{i=1}^{p} \lambda_i$, with respect to $k \in [\![1, 30]\!]$. $\{\lambda_i\}_{i=1}^{p}$ are the eigenvalues in descending order of the SCM computed with all pixels of *Indian Pines* [9]. Only the first 30 eigenvalues out of $p = 200$ are plotted. We notice that the first 5 principle eigenvectors contain more than 95% of the cumulative variance.

Because *Indian Pines* [9] has 16 classes, we set the number of clusters K to 16. Furthermore, we set $k = 5$. Indeed, from Figure 4.6, we observe that the first 5 principal eigenvectors of the SCM calculated on *Indian Pines* [9] contain more than 95% of the total variance. Since we use an unsupervised algorithm, the output classes are not necessarily the same as the ground truth. Hence, a Kuhn–Munkres algorithm is applied to the segmentation to recover ground truth's classes. Furthermore, we do 10 different initializations (parameter l in Algorithm 10) and keep the clustering with the lowest inertia (4.36). To measure the variability of the results, each *K-means++* is run 10 times. The averaged Overall Accuracy (OA), as well as the averaged mean Intersection over Union (mIoU), are reported with their standard deviations (std) in Table 4.3.

First of all, the methods based on non-Euclidean geometries all surpass the other methods ("center pixel" and "mean pixel") by at least 8.9% in terms of averaged Overall Accuracy. This proves the interest in using Riemannian geometries other than the simple Euclidean one. Secondly, "robust subspace, $\gamma = 0$" slightly exceeds "subspace SCM" which shows the interest of robust estimation of subspaces. Thirdly, "robust subspace" with $\gamma = 0.1$ outperform "robust subspace $\gamma = 0$" by nearly 4%. Finally, our method "robust subspace $\gamma = 0.1$" outperforms the strong baseline "SCM" by 2.8% in terms of Overall Accuracy. However, "SCM" performs better in terms of mIoU, by nearly 2%, compared to "robust subspace, $\gamma = 0.1$". This means "SCM" better classifies classes with small number of samples.

As mentioned in Section 4.2, a trade-off must be made between the

PCA	Descriptor	OA $\pm$ std	mIoU $\pm$ std
Yes	center pixel	32.66 ± 0.84	18.30 ± 0.82
	mean pixel	34.02 ± 0.48	20.17 ± 2.00
	SCM	45.08 ± 1.58	$\mathbf{29.95 \pm 1.87}$
No	subspace SCM	42.95 ± 0.71	27.06 ± 0.76
	robust subspace, $\gamma = 0$	43.93 ± 0.93	28.11 ± 0.63
	robust subspace, $\gamma = 0.1$	$\mathbf{47.89 \pm 2.67}$	28.00 ± 1.49

Table 4.3: Performance of the different descriptors on *Indian Pines* [9] with $w = 7$ and $k = 5$.

subspaces' distance and textures' distance. A hyperparameter $\gamma \in [0, 1]$ realizes this trade-off. Figure 4.7 shows that our method "robust subspace" outperforms the "SCM" when we emphasis the $\mathrm{Gr}_{p,k}$ distance. Figure 4.7 illustrates that our method works for an interval of γ and therefore does not need a critical choice to maximize Overall Accuracy. However, to maximize mIoU, the smaller γ the better.

Figure 4.9 presents the segmentations of 4 methods: "center pixel", "SCM", "robust subspace $\gamma = 0$" and "robust subspace $\gamma = 0.1$". The segmentations are those with the lowest inertia (4.36) for each method. We note a significant improvement occurs on class 14 (lower right part) between baseline "SCM" in Figure 4.9b and our method "robust subspace $\gamma = 0.1$" in Figure 4.9d. Also, the textures help to better cluster classes 8 and 14, see Figure 4.9c versus 4.9d.

Finally, our method "robust subspace $\gamma = 0.1$" converges quickly, *i.e.* in less than 20 iterations (see Figure 4.8). Interestingly, the WCSS (4.36) decreases a lot in the first iterations and hence the *K-means++* can be stopped after few iterations to faster computation.

4.6 . Conclusions

This chapter proposed to study the information geometry of heteroscedastic signals embedded in WGN. This geometric approach offered a unified framework in order to 1) derive new optimization algorithm based on Riemannian stochastic gradient descent; 2) obtain iCRBs (error bounds

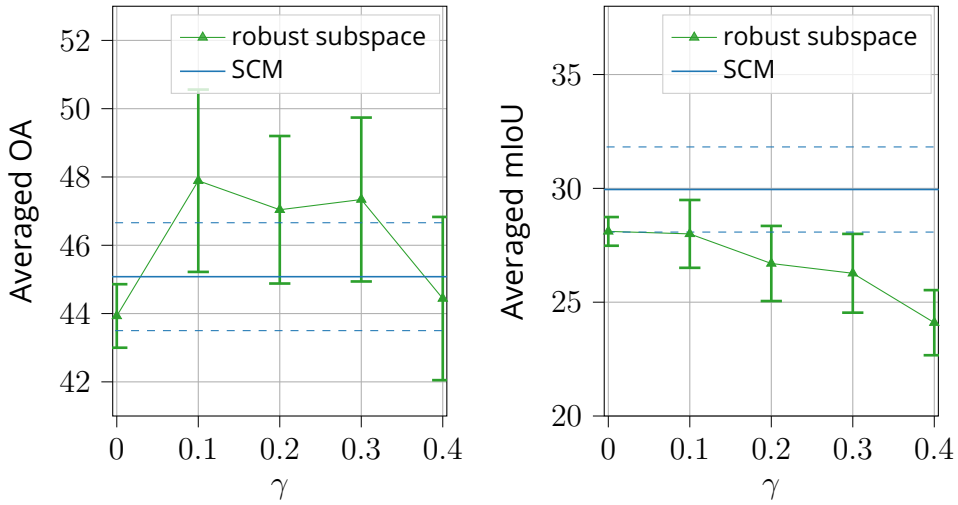


Figure 4.7: Overall accuracy and mIoU of our method "robust subspace" with respect to parameter γ on *Indian Pines* [9] with $w = 7$ and $k = 5$. Mean performance are reported with their standard deviations (with error bars for "robust subspace" and in dashed blue lines for "SCM").

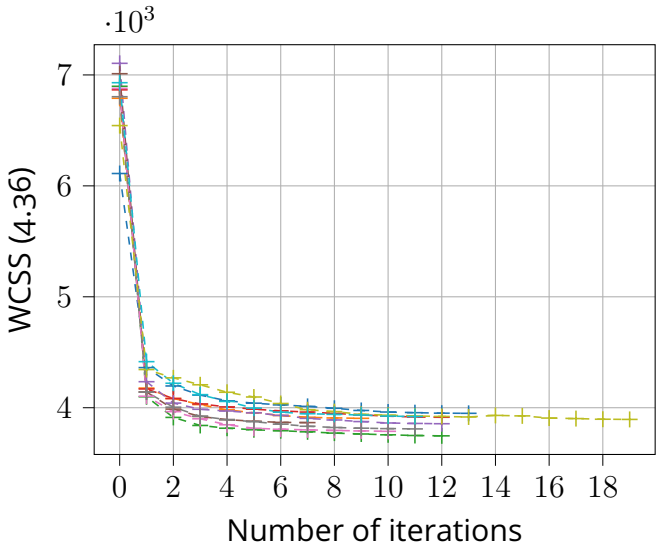
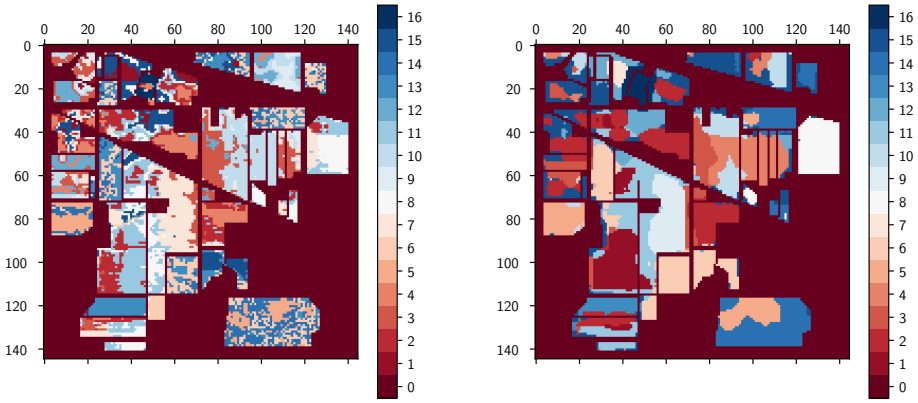
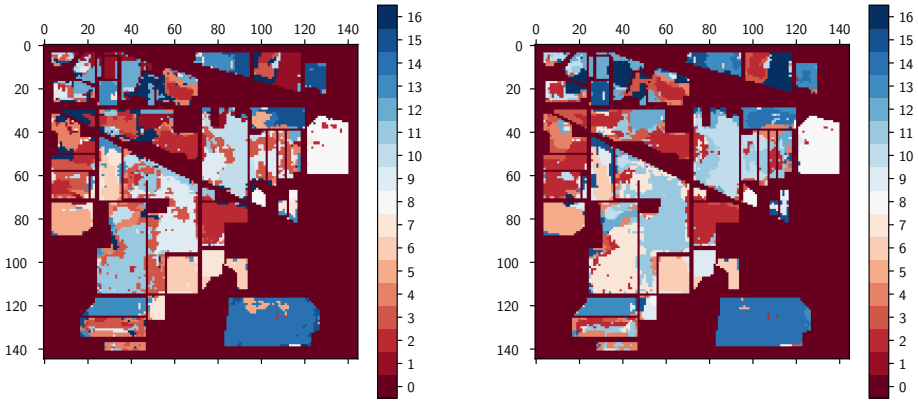


Figure 4.8: WCSS (4.36) versus the iterations of *K-means++* [7] for "robust subspace" $\gamma = 0.1$ corresponding to Figure 4.9d. The curves correspond to 10 initializations.



(a) “center pixel”:
OA $= 31.2\%$, mIoU $= 18.8\%$

(b) “SCM”:
OA $= 45.2\%$, mIoU $= 31.5\%$



(c) “robust subspace $\gamma = 0$”:
OA $= 43.3\%$, mIoU $= 27.3\%$

(d) “robust subspace $\gamma = 0.1$”:
OA $= 47.2\%$, mIoU $= 29.3\%$

Figure 4.9: *Indian Pines* [9] segmentation results achieved using 4 methods: “center pixel”, “SCM”, “robust subspace” $\gamma = 0$ and “robust subspace” $\gamma = 0.1$ ($w = 7$ and $k = 5$ for all methods). These segmentations are those with the lowest WCSS computed with their respective distances.

driven by a Riemannian distance) with interesting interpretations; 3) propose a new Riemannian clustering algorithm based on the model features, which was applied it to a hyperspectral image to illustrate the interest of the approach.

4.A . Appendix

4.A.1 . Proof of Proposition 21

By definition of the Fisher information metric [121],

$$\langle \bar{\xi}, \bar{\eta} \rangle_{\bar{\theta}}^{\mathsf{FIM}} = \mathbb{E}[\mathrm{D}\,\overline{\mathcal{L}}(\bar{\theta})[\bar{\xi}]\,\mathrm{D}\,\overline{\mathcal{L}}(\bar{\theta})[\bar{\eta}]] = \mathbb{E}[\mathrm{D}^2\,\overline{\mathcal{L}}(\bar{\theta})[\bar{\xi}, \bar{\eta}]]$$

$\overline{\mathcal{L}}$ defined in (4.5) can be written as

$$\overline{\mathcal{L}}(\bar{\theta}) = \sum_{i=1}^{n} \mathcal{L}_{\boldsymbol{x}}^{g}(\overline{\psi}_i(\bar{\theta})),$$

where $\mathcal{L}_{\boldsymbol{x}}^{g}(\boldsymbol{\Sigma}) = \log|\boldsymbol{\Sigma}| + \boldsymbol{x}^H \boldsymbol{\Sigma}^{-1} \boldsymbol{x}$ is the negative Gaussian log-likelihood on $\mathcal{H}_p^{++}$. Thus, following the reasoning of [17, Proposition 6] and [18, Proposition 3.1], one can show

$$\langle \bar{\xi}, \bar{\eta} \rangle_{\bar{\theta}}^{\mathsf{FIM}} = \sum_{i=1}^{n} \langle \mathrm{D}\,\overline{\psi}_i(\bar{\theta})[\bar{\xi}], \mathrm{D}\,\overline{\psi}_i(\bar{\theta})[\bar{\eta}] \rangle_{\overline{\psi}_i(\bar{\theta})}^{\mathsf{FIM},g}, \tag{4.38}$$

where $\langle \boldsymbol{\xi}_{\boldsymbol{\Sigma}}, \boldsymbol{\eta}_{\boldsymbol{\Sigma}} \rangle_{\boldsymbol{\Sigma}}^{\mathsf{FIM},g} = \mathrm{Tr}(\boldsymbol{\Sigma}^{-1} \boldsymbol{\xi}_{\boldsymbol{\Sigma}} \boldsymbol{\Sigma}^{-1} \boldsymbol{\eta}_{\boldsymbol{\Sigma}})$ is the Fisher information metric of the Gaussian distribution on $\mathcal{H}_p^{++}$; see *e.g.* [121]. The definition (4.4) of $\overline{\psi}_i(\bar{\theta})$ and $\mathrm{D}\,\overline{\psi}_i(\bar{\theta})[\bar{\xi}] = \tau_i(\boldsymbol{U}\boldsymbol{\xi}_{\boldsymbol{U}}^H + \boldsymbol{\xi}_{\boldsymbol{U}}\boldsymbol{U}^H) + (\boldsymbol{\xi}_{\boldsymbol{\tau}})_i \boldsymbol{U}\boldsymbol{U}^H$ yields

$$\begin{aligned}
&\langle \mathrm{D}\,\overline{\psi}_i(\bar{\theta})[\bar{\xi}], \mathrm{D}\,\overline{\psi}_i(\bar{\theta})[\bar{\eta}] \rangle_{\overline{\psi}_i(\bar{\theta})}^{\mathsf{FIM},g} = \\
&\quad (\boldsymbol{\xi}_{\boldsymbol{\tau}})_i (\boldsymbol{\eta}_{\boldsymbol{\tau}})_i \langle \boldsymbol{U}\boldsymbol{U}^H, \boldsymbol{U}\boldsymbol{U}^H \rangle_{\overline{\psi}_i(\bar{\theta})}^{\mathsf{FIM},g} \\
&\quad + (\boldsymbol{\xi}_{\boldsymbol{\tau}})_i\, \tau_i \langle \boldsymbol{U}\boldsymbol{U}^H, \boldsymbol{U}\boldsymbol{\eta}_{\boldsymbol{U}}^H + \boldsymbol{\eta}_{\boldsymbol{U}}\boldsymbol{U}^H \rangle_{\overline{\psi}_i(\bar{\theta})}^{\mathsf{FIM},g} \\
&\quad + \tau_i\, (\boldsymbol{\eta}_{\boldsymbol{\tau}})_i \langle \boldsymbol{U}\boldsymbol{\xi}_{\boldsymbol{U}}^H + \boldsymbol{\xi}_{\boldsymbol{U}}\boldsymbol{U}^H, \boldsymbol{U}\boldsymbol{U}^H \rangle_{\overline{\psi}_i(\bar{\theta})}^{\mathsf{FIM},g} \\
&\quad + (\tau_i)^2 \langle \boldsymbol{U}\boldsymbol{\xi}_{\boldsymbol{U}}^H + \boldsymbol{\xi}_{\boldsymbol{U}}\boldsymbol{U}^H, \boldsymbol{U}\boldsymbol{\eta}_{\boldsymbol{U}}^H + \boldsymbol{\eta}_{\boldsymbol{U}}\boldsymbol{U}^H \rangle_{\overline{\psi}_i(\bar{\theta})}^{\mathsf{FIM},g}
\end{aligned} \tag{4.39}$$

Then we compute each term separately:

$$\langle \boldsymbol{U}\boldsymbol{U}^H, \boldsymbol{U}\boldsymbol{U}^H \rangle_{\overline{\psi}_i(\bar{\theta})}^{\mathsf{FIM},g} = \frac{k}{(1+\tau_i)^2} \tag{4.40}$$

$$\langle \boldsymbol{U}\,\boldsymbol{U}^H, \boldsymbol{U}\,\boldsymbol{\eta}_{\boldsymbol{U}}^H + \boldsymbol{\eta}_{\boldsymbol{U}}\,\boldsymbol{U}^H \rangle_{\overline{\psi}_i(\bar{\theta})}^{\mathsf{FIM},g} = 0 \tag{4.41}$$

$$\langle \boldsymbol{U}\,\boldsymbol{\xi}_{\boldsymbol{U}}^H + \boldsymbol{\xi}_{\boldsymbol{U}}\,\boldsymbol{U}^H, \boldsymbol{U}\,\boldsymbol{U}^H \rangle_{\overline{\psi}_i(\bar{\theta})}^{\mathsf{FIM},g} = 0 \tag{4.42}$$

$$\begin{aligned}
&\langle \boldsymbol{U}\boldsymbol{\xi}_{\boldsymbol{U}}^H + \boldsymbol{\xi}_{\boldsymbol{U}}\boldsymbol{U}^H, \boldsymbol{U}\boldsymbol{\eta}_{\boldsymbol{U}}^H + \boldsymbol{\eta}_{\boldsymbol{U}}\boldsymbol{U}^H \rangle_{\overline{\psi}_i(\bar{\theta})}^{\mathsf{FIM},g} = \\
&\frac{2}{1+\tau_i} \mathfrak{Re}\left(\mathrm{Tr}\left(\boldsymbol{\xi}_{\boldsymbol{U}}^H \boldsymbol{\eta}_{\boldsymbol{U}}\right)\right)
\end{aligned} \tag{4.43}$$

The Fisher information metric stated in Proposition 21 is obtained by combining (4.38), (4.39), (4.42), and (4.43).

4.A.2 . Proof of Proposition 22

Since $\mathsf{Gr}_{p,k}$ is a quotient manifold of $\mathsf{St}_{p,k}$, $\operatorname{grad}_{\mathcal{M}_{p,k,n}} \mathcal{L}_i(\theta)$ is represented by $\operatorname{grad}_{\overline{\mathcal{M}}_{p,k,n}} \overline{\mathcal{L}}_i(\bar{\theta}) \in \mathcal{H}_{\boldsymbol{U}} \times T_{\boldsymbol{\tau}}(\mathbb{R}^{++})^n$. By definition, $\forall \bar{\xi} \in T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$, $\mathrm{D}\,\overline{\mathcal{L}}_i(\bar{\theta})[\bar{\xi}] = \langle \operatorname{grad}_{\overline{\mathcal{M}}_{p,k,n}} \overline{\mathcal{L}}_i(\bar{\theta}), \bar{\xi} \rangle_{\bar{\theta}}^{\mathsf{FIM}}$ [1]. Notice that $\left|\overline{\psi}_i(\bar{\theta})\right| = (1+\tau_i)^k$ and $(\overline{\psi}_i(\bar{\theta}))^{-1} = \boldsymbol{I}_p - \dfrac{\tau_i}{1+\tau_i} \boldsymbol{U}\boldsymbol{U}^H$ (Woodbury formula). It follows that

$$\begin{aligned}
\mathrm{D}\,\overline{\mathcal{L}}_i(\bar{\theta})[\bar{\xi}] &= -2\frac{\tau_i}{1+\tau_i} \mathfrak{Re}\left(\operatorname{Tr}\left(\boldsymbol{x}\boldsymbol{x}^H \boldsymbol{U} \boldsymbol{\xi}_{\boldsymbol{U}}^H\right)\right) + \frac{k\,(1+\tau_i) - \boldsymbol{x}^H \boldsymbol{U}\boldsymbol{U}^H \boldsymbol{x}}{(1+\tau_i)^2} (\boldsymbol{\xi}_{\boldsymbol{\tau}})_i \\
&= 2nc_{\boldsymbol{\tau}} \langle -\frac{\tau_i}{nc_{\boldsymbol{\tau}}(1+\tau_i)} \boldsymbol{x}\,\boldsymbol{x}^H \boldsymbol{U}, \boldsymbol{\xi}_{\boldsymbol{U}} \rangle_{\boldsymbol{U}}^{\mathsf{St}_{p,k}} + \langle \boldsymbol{a}, \boldsymbol{\xi}_{\boldsymbol{\tau}} \rangle_{\boldsymbol{\tau}}^{(\mathbb{R}_*^+)^n}
\end{aligned}$$

where $\boldsymbol{a} \in \mathbb{R}^n$ is a vector such that

$$\boldsymbol{a}_j = \begin{cases} 1 + \tau_i - \frac{1}{k} \boldsymbol{x}^H \boldsymbol{U}\boldsymbol{U}^H \boldsymbol{x} & \text{for} \quad j = i \\ 0 \quad \text{otherwise.} \end{cases}$$

To obtain the Riemannian gradient $\operatorname{grad}_{\overline{\mathcal{M}}_{p,k,n}} \overline{\mathcal{L}}(\bar{\theta})$ by identification, it remains to project $-\frac{\tau_i}{nc_{\boldsymbol{\tau}}(1+\tau_i)} \boldsymbol{x}\,\boldsymbol{x}^H \boldsymbol{U}$ onto $\mathcal{H}_{\boldsymbol{U}}$ with $P_{\boldsymbol{U}}^{\mathsf{Gr}_{p,k}}(\boldsymbol{\xi}_{\boldsymbol{U}}) = \left(\boldsymbol{I}_p - \boldsymbol{U}\,\boldsymbol{U}^H\right) \boldsymbol{\xi}_{\boldsymbol{U}}$ [1], which is enough to conclude.

4.A.3 . Proof of Proposition 23 and 24

In this section we derive the elements of the generic iCRB inequality (4.26) for the estimation problem of $\theta \in \mathcal{M}_{p,k,n}$ (and data model in (4.3)) when the chosen error metric is the product one from Definition 43. To do so, we need to select a proper system of coordinates of the tangent space $T_\theta \mathcal{M}_{p,k,n}$ so that the entries of $\boldsymbol{F}^{-1}$ can be actually obtained: $\mathcal{M}_{p,k,n}$ being a quotient manifold, there are two solutions in order to represent this object. The first one is to simply consider coordinates of $T_{\bar{\theta}}\overline{\mathcal{M}}_{p,k,n}$ without restrictions. The resulting Fisher information matrix will then be singular, but its pseudo-inverse still yields the desired inequality [20]. The second option, which will be chosen here, is to consider only coordinates in the horizontal space $\mathcal{H}_{\bar{\theta}}$, which is given in our case in (4.14).

Two ingredients are thus needed to establish the Fisher information matrix as in (4.26):

(i) The Fisher information metric $\langle \cdot, \cdot \rangle_{\bar{\theta}}^{\mathsf{FIM}}$, which was given in Proposition 21.

(ii) A basis of the horizontal space $\mathcal{H}_{\bar{\theta}}$ in (4.14) that is orthonormal with respect to the error metric (*i.e.*, the product metric from Definition (43)), which is given in the following proposition.

Proposition 25 (Orthonormal basis)**.** *Given $\bar{\theta} \in \overline{\mathcal{M}}_{p,k,n}$, an orthonormal basis of the horizontal space $\mathcal{H}_{\bar{\theta}}$ defined in* (4.14) *with respect to the Riemannian metric of Definition 43 is*

$$\{e^q_{\bar{\theta}}\}_{1\leq q\leq 2(p-k)k+n} = B_{\boldsymbol{U}} \cup B_{\boldsymbol{\tau}},$$

with

$$B_{\boldsymbol{U}} = \bigcup_{\substack{1\leq i\leq p-k\\ 1\leq j\leq k}} \left\{ \left(\alpha^{-\frac{1}{2}} \boldsymbol{U}_\perp \boldsymbol{K_{ij}}, \mathbf{0}\right), \left(\alpha^{-\frac{1}{2}} i\, \boldsymbol{U}_\perp \boldsymbol{K_{ij}}, \mathbf{0}\right) \right\},$$
$$B_{\boldsymbol{\tau}} = \bigcup_{1\leq i\leq n} \{(\mathbf{0}, \beta^{-\frac{1}{2}} \tau_i \boldsymbol{e_i})\},$$

where $\boldsymbol{U}_\perp \in \mathrm{St}_{p,k_{p,p-k}}$ such that $\boldsymbol{U}^H \boldsymbol{U}_\perp = \mathbf{0}$; $\boldsymbol{K_{ij}} \in \mathbb{R}^{(p-k)\times k}$: its ij^{th} element is 1, zeros elsewhere; and $\boldsymbol{e_i} \in \mathbb{R}^n$: its i^{th} element is 1, zero elsewhere.

Proof. As $\{e^q_{\bar{\theta}}\}$ contains the right amount of elements, it suffices to show that, $\forall q, l \in [\![1, 2(p-k)k+n]\!]$ such that $q \neq l$, we have $\langle e^q_{\bar{\theta}}, e^l_{\bar{\theta}}\rangle^{\overline{\mathcal{M}}_{p,k,n}}_{\bar{\theta}} = 0$ and $\langle e^q_{\bar{\theta}}, e^q_{\bar{\theta}}\rangle^{\overline{\mathcal{M}}_{p,k,n}}_{\bar{\theta}} = 1$. This can easily be checked by calculation. □

Using this system of coordinates, the ql^{th} element of the Fisher information matrix $\boldsymbol{F}_\theta$ is then represented by

$$(\boldsymbol{F}_\theta)_{ql} = \langle e^q_{\bar{\theta}}, e^l_{\bar{\theta}}\rangle^{\mathsf{FIM}}_{\bar{\theta}}. \tag{4.44}$$

Remarkably, $\boldsymbol{F}_\theta$ will turn to be diagonal which enables us to obtain closed forms iCRB on $\mathcal{M}_{p,k,n}$, $\mathsf{Gr}_{p,k}$ and $(\mathbb{R}^+_*)^n$ respectively. To show that $\boldsymbol{F}_\theta$ is block diagonal, it suffices to notice that there are no crossed terms between tangent vectors of $\boldsymbol{U}$ and $\boldsymbol{\tau}$ in the Fisher information metric of Proposition 23. Computing the elements of $\boldsymbol{F}_{\boldsymbol{U}}$ yields

$$\langle(\alpha^{-\frac{1}{2}}\boldsymbol{U}\boldsymbol{K}_{ij}, \mathbf{0}), (\alpha^{-\frac{1}{2}}\boldsymbol{U}\boldsymbol{K}_{lm}, \mathbf{0})\rangle^{\mathsf{FIM}}_{\bar{\theta}} = \begin{cases} 2\,\alpha^{-1}\, n\, c_{\boldsymbol{\tau}} & \text{if } ij = l\,m \\ 0 & \text{otherwise} \end{cases}$$

$$\langle(\alpha^{-\frac{1}{2}} i\boldsymbol{U}\boldsymbol{K}_{ij}, \mathbf{0}), (\alpha^{-\frac{1}{2}} i\,\boldsymbol{U}\boldsymbol{K}_{lm}, \mathbf{0})\rangle^{\mathsf{FIM}}_{\bar{\theta}} = \begin{cases} 2\alpha^{-1}\, n\, c_{\boldsymbol{\tau}} & \text{if } ij = l\,m \\ 0 & \text{otherwise} \end{cases}$$

$$\langle(\alpha^{-\frac{1}{2}}\boldsymbol{U}\boldsymbol{K}_{ij}, \mathbf{0}), (\alpha^{-\frac{1}{2}} i\,\boldsymbol{U}\boldsymbol{K}_{lm}, \mathbf{0})\rangle^{\mathsf{FIM}}_{\bar{\theta}} = 0$$

Hence, $\boldsymbol{F}_{\boldsymbol{U}} = 2\alpha^{-1}\, n\, c_{\boldsymbol{\tau}}\, \boldsymbol{I}_{2\,(p-k)\,k}$. Computing the elements of $\boldsymbol{F}_{\boldsymbol{\tau}}$ yields

$$\begin{aligned}\langle(\mathbf{0}, \beta^{-\frac{1}{2}}\tau_i \boldsymbol{e_i}), (\mathbf{0}, \beta^{-\frac{1}{2}}\tau_j \boldsymbol{e_j})\rangle_{\bar{\theta}}^{\mathsf{FIM}} &= \beta^{-1}\, k\, \frac{\tau_i\, \tau_j}{(1+\tau_i)\,(1+\tau_j)} \boldsymbol{e_i}^T \boldsymbol{e_j} \\ &= \begin{cases} \beta^{-1}\, k\, \dfrac{\tau_i^2}{(1+\tau_i)^2} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}\end{aligned}$$

Hence, $\boldsymbol{F}_{\boldsymbol{\tau}} = \beta^{-1} k\, \mathrm{diag}\,(\boldsymbol{\tau}^{\odot 2} \odot (\mathbf{1} + \boldsymbol{\tau})^{\odot -2})$, which concludes the part concerning the proof of Proposition 23.

Finally, we note that

$$\mathrm{Tr}\left(\boldsymbol{F}_{\boldsymbol{U}}^{-1}\right) = \frac{\alpha(p-k)\,k}{n\, c_{\boldsymbol{\tau}}} \text{ and } \mathrm{Tr}\left(\boldsymbol{F}_{\boldsymbol{\tau}}^{-1}\right) = \frac{\beta}{k} \sum_{i=1}^{n} \frac{(1+\tau_i)^2}{\tau_i^2}.$$

Furthermore, we get,

$$\mathrm{Tr}\left(\boldsymbol{F}_{\theta}^{-1}\right) = \frac{\alpha(p-k)\,k}{n\, c_{\boldsymbol{\tau}}} + \frac{\beta}{k} \sum_{i=1}^{n} \frac{(1+\tau_i)^2}{\tau_i^2}.$$

It follows that the error of an unbiased estimator $\hat{\theta}$ of the true parameter θ in $\mathcal{M}_{p,k,n}$ admits the iCRB

$$\mathbb{E}[d^2_{\mathcal{M}_{p,k,n}}(\hat{\theta}, \theta)] \geq \mathrm{Tr}\left(\boldsymbol{F}_{\theta}^{-1}\right) \tag{4.45}$$

if we neglect the curvature terms when applying Theorem 2 of [121]. Since $\boldsymbol{F}_{\bar{\theta}}$ is block-diagonal we also get two separated iCRB for the parameters on $\mathsf{Gr}_{p,k}$ and $(\mathbb{R}_*^+)^n$ respectively, *i.e.*:

$$\mathbb{E}[d^2_{\mathsf{Gr}_{p,k}}(\pi(\hat{\boldsymbol{U}}), \pi(\boldsymbol{U}))] \geq \alpha^{-1}\, \mathrm{Tr}(\boldsymbol{F}_{\boldsymbol{U}}^{-1}) = \frac{(p-k)k}{n c_{\boldsymbol{\tau}}}, \tag{4.46}$$

$$\mathbb{E}[d^2_{(\mathbb{R}_*^+)^n}(\hat{\boldsymbol{\tau}}, \boldsymbol{\tau})] \geq \beta^{-1}\, \mathrm{Tr}(\boldsymbol{F}_{\boldsymbol{\tau}}^{-1}) = \frac{1}{k} \sum_{i=1}^{n} \frac{(1+\tau_i)^2}{\tau_i^2}. \tag{4.47}$$

This concludes the proof of Proposition 24.

5 - Robust Geometric Metric Learning

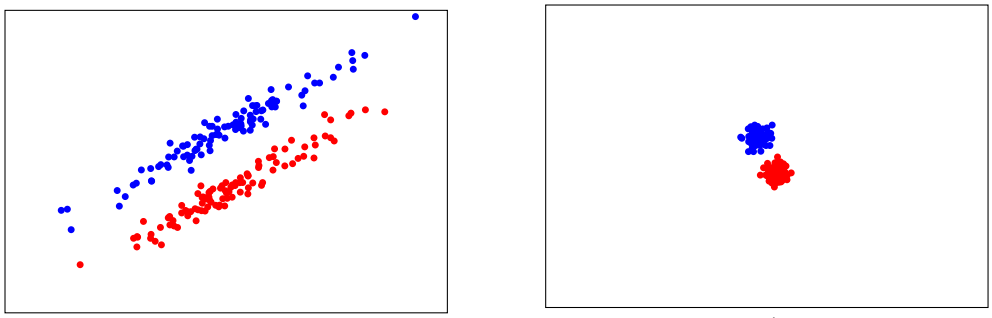
Many classification algorithms rely on the distance between data points. These algorithms include the classical *K-means*, *Nearest centroïd classifier*, *k-nearest neighbors* and their variants. The definition of the distance is thus of crucial importance since it determines which points will be considered similar or not, thus implies the classification rule. In previous chapters, statistical features θ were estimated from samples sets $\{\boldsymbol{x}_i\}_{i=1}^n$. Then, these features were classified using divergences that respect constraints of the parameter space and are associated to the considered statistical model. Here, the approach is different: the classification is performed directly on the data $\boldsymbol{x}_i$ and thus no statistical estimation is performed. To do so, classification algorithms most generally rely on the Euclidean distance, which is $d_{\boldsymbol{I}_p}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$ for $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^p$. However, this distance is prone to several issues. A pathological example is when two classes have a high variance along one common axis: within this configuration, two data points from the same class can be far away from each other, while two data points from two different classes can be very close.

To find a more relevant distance for classification, the problem of *metric learning* has been proposed. Metric learning aims at finding a Mahalanobis distance

$$d_{\boldsymbol{A}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \boldsymbol{A}^{-1} (\boldsymbol{x}_i - \boldsymbol{x}_j)}\,, \tag{5.1}$$

that brings data points from same class closer, and furthers data points from different classes away. Mathematically, metric learning is an optimization problem of a loss function that relies on $d_{\boldsymbol{A}}$. This minimization is achieved over $\boldsymbol{A}$, a matrix that belongs to $\mathcal{S}_p^{++}$ the set of $p \times p$ symmetric positive definite matrices. The constraints of symmetry and positivity are enforced so that $d_{\boldsymbol{A}}$ is a distance. An illustration of data $\{\boldsymbol{x}_i\}$ and their whitened counterpart $\{\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{x}_i\}$ is presented in Figure 5.1. In this chapter, we focus on developing metric learning methods that are robust to outliers using robust statistics (as presented in Chapter 1) and fast using Riemannian optimization (theory presented in Chapter 2).

In the following, we consider being in a supervised regime with K classes, *i.e.* m data points $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}$ in $\mathbb{R}^p$ with their labels in $[\![1, K]\!]$ are available. Data points can be grouped by classes and the elements of the k^{th} class are denoted $\{\boldsymbol{x}_{kl}\}$. Then, n_k pairs, $(\boldsymbol{x}_{kl}, \boldsymbol{x}_{kq})$ with $kl \neq kq$, of elements of the class k are formed. The set S_k contains all these pairs and S contains the $n_S = \sum_{k=1}^K n_k$ pairs of all the classes. When S is used, the class of a pair is not relevant, thus it is denoted by $(\boldsymbol{x}_l, \boldsymbol{x}_q)$ instead of $(\boldsymbol{x}_{kl}, \boldsymbol{x}_{kq})$. The ratio $\frac{n_k}{n_S}$ is denoted π_k. Then, each vector $\boldsymbol{s}_{ki}$ is defined as the subtraction



$\{\boldsymbol{x}_i\}$ $\{\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{x}_i\}$

Figure 5.1: Example of the whitening effect by a matrix $\boldsymbol{A}$ learned with a metric learning optimization problem. On the left: data that belong to two classes, blue and red, are stretched along a common axis. On the right: the same data are whitened by $\boldsymbol{A}$.

of the elements of each pair in S_k, *i.e.* $\boldsymbol{s}_{ki} = \boldsymbol{x}_{kl} - \boldsymbol{x}_{kq}$ for $(\boldsymbol{x}_{kl}, \boldsymbol{x}_{kq}) \in S_k$, i being the index of the pair and l, q the indices of the elements of this i^{th} pair. Thus, the set $\{\boldsymbol{s}_{ki}\}$ contains n_k elements. Then, the set D contains n_D pairs of vectors that do not belong to the same class. Each vector $\boldsymbol{d}_i$ is defined as the subtraction of the elements of each pair in D, *i.e.* $\boldsymbol{d}_i = \boldsymbol{x}_l - \boldsymbol{x}_q$ for $(\boldsymbol{x}_l, \boldsymbol{x}_q) \in D$. Finally, $\mathcal{S}_p$ is the set of $p \times p$ symmetric matrices, $\mathcal{S}_p^{++}$ is the set of $p \times p$ symmetric positive definite matrices, and $\mathcal{SS}_p^{++}$ is the set of $p \times p$ symmetric positive definite matrices with unit determinant.

This chapter is organized as follows. Section 5.1 presents the state of the art of metric learning and relates it to covariance estimation. Then, Section 5.2 introduces the *RGML* estimation problem. Solving this minimization problem estimates a covariance matrix that is meant to be used in the Mahalanobis distance 5.1. The formulation of RGML is general and two costs functions called *RGML Gaussian* and *RGML Tyler* are specified. In Section 5.3, two Riemannian gradient descents are proposed to minimize RGML Gaussian and RGML Tyler. Finally, these two algorithms are applied on real datasets in Section 5.4.

5.1 . Metric learning: state of the art and covariance estimation

5.1.1 . State of the art

Many metric learning problems have been formulated over the years (see e.g. [124] for a complete survey). In the following, we present notable ones that are related to our proposal.

MMC [142] (Mahalanobis Metric for Clustering) was one of the earliest paper in this field. This method minimizes the sum of squared distances

over similar data while constraining dissimilar data to be far away from each other. MMC writes

$$\begin{aligned} &\underset{\boldsymbol{A}\in\mathcal{S}_p^{++}}{\text{minimize}} \sum_{(\boldsymbol{x}_l,\boldsymbol{x}_q)\in S} d^2_{\boldsymbol{A}}(\boldsymbol{x}_l,\boldsymbol{x}_q) \\ &\text{subject to} \sum_{(\boldsymbol{x}_l,\boldsymbol{x}_q)\in D} d_{\boldsymbol{A}}(\boldsymbol{x}_l,\boldsymbol{x}_q) \geq 1. \end{aligned} \tag{5.2}$$

Notice that $d_{\boldsymbol{A}}$ (rather than $d^2_{\boldsymbol{A}}$) is involved in the constraint in order to avoid a trivial rank-one solution.

Then, *ITML* [50] (Information-Theoretic Metric Learning) proposed to find a matrix $\boldsymbol{A}$ that stays close to a predefined matrix $\boldsymbol{A}_0$ while respecting constraints of similarities and dissimilarities. The proximity between $\boldsymbol{A}$ and $\boldsymbol{A}_0$ is measured with the *Gaussian Kullback-Leibler divergence* $D_{KL}(\boldsymbol{A}_0,\boldsymbol{A}) = \operatorname{Tr}(\boldsymbol{A}^{-1}\boldsymbol{A}_0) + \log|\boldsymbol{A}\boldsymbol{A}_0^{-1}|$. ITML writes

$$\begin{aligned} \underset{\boldsymbol{A}\in\mathcal{S}_p^{++}}{\text{minimize}} \quad & \operatorname{Tr}(\boldsymbol{A}^{-1}\boldsymbol{A}_0) + \log|\boldsymbol{A}| \\ \text{subject to} \quad & d^2_{\boldsymbol{A}}(\boldsymbol{x}_l,\boldsymbol{x}_q) \leq u, \quad (\boldsymbol{x}_l,\boldsymbol{x}_q)\in S, \\ & d^2_{\boldsymbol{A}}(\boldsymbol{x}_l,\boldsymbol{x}_q) \geq l, \quad (\boldsymbol{x}_l,\boldsymbol{x}_q)\in D, \end{aligned} \tag{5.3}$$

where $u, v \in \mathbb{R}$ are threshold parameters, chosen to enforce closeness of similar points and farness of dissimilar points. Usually $\boldsymbol{A}_0$ is chosen as the identity matrix or as the sample covariance matrix (SCM) of the set $\{\boldsymbol{s}_{ki}\}$.

Next, *GMML* (Geometric Mean Metric Learning) [146] is an algorithm of great interest. Indeed, it achieves impressive performance on several datasets while being very fast thanks to a closed form formula. The GMML problem writes

$$\underset{\boldsymbol{A}\in\mathcal{S}_p^{++}}{\text{minimize}} \frac{1}{n_S}\sum_{(\boldsymbol{x}_l,\boldsymbol{x}_q)\in S} d^2_{\boldsymbol{A}}(\boldsymbol{x}_l,\boldsymbol{x}_q) + \frac{1}{n_D}\sum_{(\boldsymbol{x}_l,\boldsymbol{x}_q)\in D} d^2_{\boldsymbol{A}^{-1}}(\boldsymbol{x}_l,\boldsymbol{x}_q). \tag{5.4}$$

The intuition behind this problem is that $d_{\boldsymbol{A}^{-1}}$ should be able to further away dissimilar points while $d_{\boldsymbol{A}}$ close together similar points. Then, GMML formulation (5.4) can be rewritten

$$\underset{\boldsymbol{A}\in\mathcal{S}_p^{++}}{\text{minimize}} \operatorname{Tr}(\boldsymbol{A}^{-1}\boldsymbol{S}) + \operatorname{Tr}(\boldsymbol{A}\boldsymbol{D}), \tag{5.5}$$

where $\boldsymbol{S} = \frac{1}{n_S}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\boldsymbol{s}_{ki}\boldsymbol{s}_{ki}^T$ and $\boldsymbol{D} = \frac{1}{n_D}\sum_{i=1}^{n_D}\boldsymbol{d}_i\boldsymbol{d}_i^T$. In [146], the solution of (5.5) is derived. It is the geodesic mid-point between $\boldsymbol{S}^{-1}$ and $\boldsymbol{D}$, *i.e.* $\boldsymbol{A}^{-1} = \boldsymbol{S}^{-1}\#_{\frac{1}{2}}\boldsymbol{D}$ where

$$\boldsymbol{S}^{-1}\#_t\boldsymbol{D} = \boldsymbol{S}^{-\frac{1}{2}}\left(\boldsymbol{S}^{\frac{1}{2}}\boldsymbol{D}\boldsymbol{S}^{\frac{1}{2}}\right)^t\boldsymbol{S}^{-\frac{1}{2}} \text{ with } t\in[0,1]. \tag{5.6}$$

Then, [146] proposes to generalize this solution by $\boldsymbol{A}^{-1} = \boldsymbol{S}^{-1}\#_t\boldsymbol{D}$ with $t\in[0,1]$ (*i.e.* t is no longer necessarily $\frac{1}{2}$).

5.1.2 . Metric learning as covariance matrix estimation

In this sub-section, some metric learning problems are expressed as covariance matrix estimation problems.

The first remark concerns the ITML formulation (5.3). Indeed, when the latter is written with the SCM as a prior matrix, it amounts to maximizing the likelihood of a multivariate Gaussian distribution under constraints. Therefore, ITML can be viewed as a *covariance* matrix estimation problem.

The second remark concerns the GMML solution of (5.5) which is generalized to $\boldsymbol{A}^{-1} = \boldsymbol{S}^{-1}\#_t\boldsymbol{D}$ with $t \in [0,1]$. In their experiments on real datasets, the authors often get their best performance with t small (or even null) (see Figure 3 of [146]). In this case, the GMML algorithm gives $\boldsymbol{A} = \boldsymbol{S}$. This simple, yet effective, solution can be reinterpreted with an additional assumption on the data. Let us assume that data points of each class are realizations of independent random vectors with class-dependent first and second order moments,

$$\boldsymbol{x}_{kl} \overset{d}{=} \boldsymbol{\mu}_k + \boldsymbol{\Sigma}_k^{\frac{1}{2}}\boldsymbol{u}_{kl}\,, \tag{5.7}$$

with $\boldsymbol{\mu}_k \in \mathbb{R}^p$, $\boldsymbol{\Sigma}_k \in \mathcal{S}_p^{++}$, $\mathbb{E}[\boldsymbol{u}_{kl}] = \boldsymbol{0}$ and $\mathbb{E}[\boldsymbol{u}_{kl}\boldsymbol{u}_{kq}^T] = \boldsymbol{I}_p$ if $kl = kq$, $\boldsymbol{0}_p$ otherwise. Thus, it follows that $\boldsymbol{s}_{ki} \overset{d}{=} \boldsymbol{\Sigma}_k^{\frac{1}{2}}(\boldsymbol{u}_{kl} - \boldsymbol{u}_{kq})$. Hence, the covariance matrix of $\boldsymbol{s}_{ki}$ is twice the covariance matrix of the k^{th} class, $\mathbb{E}[\boldsymbol{s}_{ki}\boldsymbol{s}_{ki}^T] \overset{d}{=} 2\boldsymbol{\Sigma}_k$. It results that, in expectation, $\boldsymbol{S}$ is twice the arithmetic mean of the covariance matrices of the different classes,

$$\mathbb{E}[\boldsymbol{S}] = \frac{1}{n_S}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\mathbb{E}[\boldsymbol{s}_{ki}\boldsymbol{s}_{ki}^T] = 2\sum_{k=1}^{K}\pi_k\boldsymbol{\Sigma}_k. \tag{5.8}$$

The only additional assumption added to GMML to get (5.8) is (5.7). This hypothesis is broad since it encompasses classical assumptions such as the Gaussian one. Also notice that using $\boldsymbol{S}$ in the Mahalanobis distance (5.1) is reminiscent of the linear discriminant analysis (LDA) pre-whitening step of the data.

5.1.3 . Motivations and contributions

From Section 5.1.2, GMML can be interpreted as a 2-steps method that computes, first, the SCM of each class and, two, their arithmetic mean. Thus, this simple approach is not robust to outliers (*e.g.* mislabeled data) since it uses the SCM as an estimator. Moreover, other mean computation can be used, such as the Riemannian mean which benefits from many properties compared to its Euclidean counterpart [145]. We propose a metric learning framework that jointly estimates regularized covariance matrices, in a robust manner, while computing their Riemannian mean. We name this framework *Riemannian Geometric Metric Learning* (*RGML*).

This idea of estimating covariance matrices while averaging them was firstly proposed in [102]. The novelty here is fourfold:

1. this formulation is applied to the problem of metric learning (see Section 5.2),
2. it makes use of the Riemannian distance on $\mathcal{S}_p^{++}$ which was not covered by [102] (see Section 5.2),
3. we leverage the Riemannian geometries of $\mathcal{S}_p^{++}$ and $\mathcal{SS}_p^{++}$ [120, 113] along with the framework of Riemannian optimization [1] and hence the proposed algorithms are flexible and could be applied to other cost functions than the Gaussian and Tyler [136] ones (see Section 5.3),
4. the framework is applied on real datasets and shows strong performance while being robust to mislabeled data (see Section 5.4).

5.2 . Problem formulation of Robust Geometric Metric Learning

5.2.1 . General formulation of RGML

The formulation of the RGML optimization problem is

$$\underset{\theta \in \mathcal{M}_{p,K}}{\text{minimize}} \left\{ \mathcal{L}(\theta) = \sum_{k=1}^{K} \pi_k \left[\mathcal{L}_k(\boldsymbol{A}_k) + \lambda d^2(\boldsymbol{A}, \boldsymbol{A}_k) \right] \right\}, \tag{5.9}$$

where $\theta = (\boldsymbol{A}, \{\boldsymbol{A}_k\})$, $\mathcal{M}_{p,K}$ is the $K+1$ product set of $\mathcal{S}_p^{++}$, *i.e.* $\mathcal{M}_{p,K} = \left(\mathcal{S}_p^{++}\right)^{K+1}$, $\mathcal{L}_k$ is a covariance matrix estimation loss on $\{\boldsymbol{s}_{ki}\}$, $\lambda > 0$ and d is a distance between matrices. In the next subsections two costs will be considered: the Gaussian negative log-likelihood and the Tyler cost function. Once (5.9) is achieved, the center matrix $\boldsymbol{A}$ is used in the Mahalanobis distance (5.1) and the $\boldsymbol{A}_k$ are discarded. The cost function $\mathcal{L}$ is explained more in details in the following.

First of all, for a fixed center matrix $\boldsymbol{A}$, (5.9) reduces to k separable problems

$$\underset{\boldsymbol{A}_k \in \mathcal{S}_p^{++}}{\text{minimize}} \ \mathcal{L}_k(\boldsymbol{A}_k) + \lambda d^2(\boldsymbol{A}, \boldsymbol{A}_k), \tag{5.10}$$

whose solutions are estimates of $\{\boldsymbol{\Sigma}_k\}$ that are regularized towards $\boldsymbol{A}$.

Second, for $\{\boldsymbol{A}_k\}$ fixed, solving (5.9) averages the matrices $\{\boldsymbol{A}_k\}$. Indeed, in this case, (5.9) reduces to

$$\underset{\boldsymbol{A} \in \mathcal{S}_p^{++}}{\text{minimize}} \sum_{k=1}^{K} \pi_k d^2(\boldsymbol{A}, \boldsymbol{A}_k). \tag{5.11}$$

For example, if d is the Euclidean distance $d_E(\boldsymbol{A}, \boldsymbol{A}_k) = \|\boldsymbol{A} - \boldsymbol{A}_k\|_2$, then the minimum of (5.11) is the arithmetic mean $\sum_{k=1}^{K} \pi_k \boldsymbol{A}_k$. In the rest of the chapter, we consider the Riemannian distance on $\mathcal{S}_p^{++}$ [120], that is

$$d_{\mathcal{S}_p^{++}}(\boldsymbol{A}, \boldsymbol{A}_k) = \left\| \log\left(\boldsymbol{A}^{-\frac{1}{2}} \boldsymbol{A}_k \boldsymbol{A}^{-\frac{1}{2}}\right) \right\|_2. \tag{5.12}$$

A nice property of $d_{\mathcal{S}_p^{++}}$ (5.12) is its affine invariance. Indeed, for any $\boldsymbol{C}$ invertible, we have $d_{\mathcal{S}_p^{++}}(\boldsymbol{C}\boldsymbol{A}\boldsymbol{C}^T, \boldsymbol{C}\boldsymbol{A}_k\boldsymbol{C}^T) = d_{\mathcal{S}_p^{++}}(\boldsymbol{A}, \boldsymbol{A}_k)$. Thus, if $\{\boldsymbol{s}_{ki}\}$ is transformed to $\{\boldsymbol{C}\boldsymbol{s}_{ki}\}$ then the minimum $(\boldsymbol{A}, \{\boldsymbol{A}_k\})$ of (5.13) becomes $\left(\boldsymbol{C}\boldsymbol{A}\boldsymbol{C}^T, \left\{\boldsymbol{C}\boldsymbol{A}_k\boldsymbol{C}^T\right\}\right)$. Another nice property of this distance is its geodesic convexity, as it will be discussed in Section 5.3.

With this Riemannian distance, the general formulation of the RGML optimization problem (5.9) becomes

$$\underset{\theta \in \mathcal{M}_{p,K}}{\text{minimize}} \left\{ \mathcal{L}(\theta) = \sum_{k=1}^{K} \pi_k \left[\mathcal{L}_k(\boldsymbol{A}_k) + \lambda d^2_{\mathcal{S}_p^{++}}(\boldsymbol{A}, \boldsymbol{A}_k) \right] \right\}. \tag{5.13}$$

We emphasis that the optimization of (5.13) is performed with respect to all the matrices $\boldsymbol{A}$ and $\{\boldsymbol{A}_k\}$ at the same time. Thus it both estimates regularized covariance matrices $\{\boldsymbol{A}_k\}$ while averaging them to estimate their unknown barycenter $\boldsymbol{A}$.

5.2.2 . RGML Gaussian

To get a practical cost function $\mathcal{L}$ (5.13), it only remains to specify the functions $\mathcal{L}_k$. The most classical assumption on the data distribution is the Gaussian one (*e.g.* considered in ITML with the SCM as prior). Thus, the first functions $\mathcal{L}_k$ considered are the centered multivariate Gaussian negative log-likelihoods

$$\mathcal{L}_{G,k}(\boldsymbol{A}) = \frac{1}{n_k} \sum_{i=1}^{n_k} \boldsymbol{s}_{ki}^T \boldsymbol{A}^{-1} \boldsymbol{s}_{ki} + \log|\boldsymbol{A}|. \tag{5.14}$$

With this negative log-likelihood, the RGML optimization problem (5.13) becomes

$$\underset{\theta \in \mathcal{M}_{p,K}}{\text{minimize}} \left\{ \mathcal{L}_G(\theta) = \sum_{k=1}^{K} \pi_k \left[\mathcal{L}_{G,k}(\boldsymbol{A}_k) + \lambda d^2_{\mathcal{S}_p^{++}}(\boldsymbol{A}, \boldsymbol{A}_k) \right] \right\}. \tag{5.15}$$

5.2.3 . RGML Tyler

When data is non-Gaussian, robust covariance matrix estimation methods are a preferred choice. This occurs whenever the probability distribution of the data is heavy-tailed or a small proportion of the samples represents

Algorithm 11: Riemannian gradient descent to minimize $\mathcal{L}_G$ (5.15)

Input: Data $\{\boldsymbol{s}_{ki}\}$, initialization $\theta^{(0)} \in \mathcal{M}_{p,K}$
Output: $\theta^{(l)} \in \mathcal{M}_{p,K}$
for $l = 0$ ***to convergence*** **do**
  Compute a step size α (see [1, Ch. 4]) and set
  $\theta^{(l+1)} = R_{\theta^{(l)}}^{\mathcal{M}_{p,K}} \left(-\alpha \operatorname{grad}_{\mathcal{M}_{p,K}} \mathcal{L}_G(\theta^{(l)})\right)$

outlier behavior. In a classification setting, the latter happens when data are mislabeled. A classical robust estimator is the Tyler's estimator [136] which is a minimizer of the following cost function

$$\mathcal{L}_{T,k}(\boldsymbol{A}) = \frac{p}{n_k} \sum_{i=1}^{n_k} \log\left(\boldsymbol{s}_{ki}^T \boldsymbol{A}^{-1} \boldsymbol{s}_{ki}\right) + \log|\boldsymbol{A}|. \tag{5.16}$$

An important remark is that (5.16) is invariant to the scale of $\boldsymbol{A}$. Indeed $\forall \alpha > 0$, it is easily checked that $\mathcal{L}_{T,k}(\alpha \boldsymbol{A}) = \mathcal{L}_{T,k}(\boldsymbol{A})$. Thus, a constraint of unit determinant is added to (5.13) to fix the scales of $\{\boldsymbol{A}_k\}$. Furthermore, the Riemannian distance (5.12) is also the one on $\mathcal{SS}_p^{++}$. Thus, we choose to also constrain $\boldsymbol{A}$ so that it is the Riemannian mean of $\{\boldsymbol{A}_k\}$ on $\mathcal{SS}_p^{++}$. We denote by $\mathcal{SM}_{p,K}$ this new parameter space

$$\mathcal{SM}_{p,K} = \{\theta \in \mathcal{M}_{p,K}, |\boldsymbol{A}| = |\boldsymbol{A}_k| = 1, \forall k \in [\![1, K]\!]\}. \tag{5.17}$$

Thus, the RGML optimization problem (5.13) with the Tyler cost function (5.16) becomes

$$\underset{\theta \in \mathcal{SM}_{p,K}}{\text{minimize}} \left\{ \mathcal{L}_T(\theta) = \sum_{k=1}^{K} \pi_k \left[\mathcal{L}_{T,k}(\boldsymbol{A}_k) + \lambda d^2_{\mathcal{S}_p^{++}}(\boldsymbol{A}, \boldsymbol{A}_k) \right] \right\}. \tag{5.18}$$

5.3 . Riemannian optimization and geodesic convexity

The objective of this section is to present Algorithms 11 and 12 which minimize (5.15) and (5.18) respectively. They leverage the Riemannian optimization framework [1, 19]. The products manifolds $\mathcal{M}_{p,K}$ and $\mathcal{SM}_{p,K}$ (directly inherited from $\mathcal{S}_p^{++}$ and $\mathcal{SS}_p^{++}$ which have presented in Chapter 2 Section 2.4) are presented.

5.3.1 . Riemannian optimization and g-convexity on $\mathcal{M}_{p,K}$

Since, $\mathcal{M}_{p,K}$ is an open set in a vector space, the tangent space $T_\theta \mathcal{M}_{p,K}$ (linearization of the Riemannian manifold at a given point) is identified to

$(\mathcal{S}_p)^{K+1}$. Then, the affine invariant metric is chosen as the Riemannian metric [120], $\forall \xi = (\boldsymbol{\xi}, \{\boldsymbol{\xi}_k\}), \forall \eta = (\boldsymbol{\eta}, \{\boldsymbol{\eta}_k\}) \in T_\theta \mathcal{M}_{p,K}$

$$\langle \xi, \eta \rangle_\theta^{\mathcal{M}_{p,K}} = \operatorname{Tr}\left(\boldsymbol{A}^{-1}\boldsymbol{\xi}\boldsymbol{A}^{-1}\boldsymbol{\eta}\right) + \sum_{k=1}^{K} \operatorname{Tr}\left(\boldsymbol{A}_k^{-1}\boldsymbol{\xi}_k\boldsymbol{A}_k^{-1}\boldsymbol{\eta}_k\right). \tag{5.19}$$

Thus the orthogonal projection from the ambient space onto the tangent space at θ is

$$P_\theta^{\mathcal{M}_{p,K}}(\xi) = (\operatorname{sym}(\boldsymbol{\xi}), \{\operatorname{sym}(\boldsymbol{\xi}_k)\}), \tag{5.20}$$

where $\operatorname{sym}(\boldsymbol{\xi}) = \frac{1}{2}(\boldsymbol{\xi} + \boldsymbol{\xi}^T)$. Then, the exponential map (function that maps tangent vectors, such as gradients of loss functions, to points on the manifold) is

$$\exp_\theta^{\mathcal{M}_{p,K}}(\xi) = \left(\exp_{\boldsymbol{A}}^{\mathcal{S}_p^{++}}(\boldsymbol{\xi}), \left\{\exp_{\boldsymbol{A}_k}^{\mathcal{S}_p^{++}}(\boldsymbol{\xi}_k)\right\}\right), \tag{5.21}$$

where $\exp_{\boldsymbol{A}}^{\mathcal{S}_p^{++}}(\boldsymbol{\xi}) = \boldsymbol{A}\exp(\boldsymbol{A}^{-1}\boldsymbol{\xi})$. Then, for a loss function $h : \mathcal{M}_{p,K} \to \mathbb{R}$, the Riemannian gradient at θ denoted $\operatorname{grad}_{\mathcal{M}_{p,K}} h(\theta)$ is defined as the unique element such that $\forall \xi \in T_\theta \mathcal{M}_{p,K}$, $\mathrm{D}\, h(\theta)[\xi] = \langle \operatorname{grad}_{\mathcal{M}_{p,K}} h(\theta), \xi \rangle_\theta^{\mathcal{M}_{p,K}}$ where D is the directional derivative. It results that

$$\operatorname{grad}_{\mathcal{M}_{p,K}} h(\theta) = P_\theta^{\mathcal{M}_{p,K}}\left(\boldsymbol{A}\boldsymbol{G}\boldsymbol{A}, \{\boldsymbol{A}_k\boldsymbol{G}_k\boldsymbol{A}_k\}\right), \tag{5.22}$$

where $(\boldsymbol{G}, \{\boldsymbol{G}_k\})$ is the classical Euclidean gradient of h at θ. With the exponential map (5.21), and the Riemannian gradient (5.22), we have the main tools to minimize (5.15). However, to improve the numerical stability, a retraction (approximation of the exponential map (5.21)) is preferred,

$$R_\theta^{\mathcal{M}_{p,K}}(\xi) = \left(R_{\boldsymbol{A}}^{\mathcal{S}_p^{++}}(\boldsymbol{\xi}), \left\{R_{\boldsymbol{A}_k}^{\mathcal{S}_p^{++}}(\boldsymbol{\xi}_k)\right\}\right), \tag{5.23}$$

where $R_{\boldsymbol{A}}^{\mathcal{S}_p^{++}}(\boldsymbol{\xi}) = \boldsymbol{A} + \boldsymbol{\xi} + \frac{1}{2}\boldsymbol{\xi}\boldsymbol{A}^{-1}\boldsymbol{\xi}$. A Riemannian gradient descent minimizing (5.15) is presented in Algorithm 11.

We finish this subsection by presenting the geodesic convexity of $\mathcal{L}_G$ (5.15) on $\mathcal{M}_{p,K}$ (see [19, Chapter 11] for a presentation of the geodesic convexity). First of all, the geodesic on $\mathcal{M}_{p,K}$ between $a = (\boldsymbol{A}, \{\boldsymbol{A}_k\})$ and $b = (\boldsymbol{B}, \{\boldsymbol{B}_k\})$ is

$$a \#_t b = (\boldsymbol{A} \#_t \boldsymbol{B}, \{\boldsymbol{A}_k \#_t \boldsymbol{B}_k\}), \tag{5.24}$$

where $\#$ is the geodesic (5.6) on $\mathcal{S}_p^{++}$ and $t \in [0, 1]$. Then, a loss function h is said to be geodesically convex (or g-convex) if

$$h\left(a \#_t b\right) \leq t\, h(a) + (1 - t)\, h(b), \quad \forall t \in [0, 1]. \tag{5.25}$$

If h is g-convex, then any local minimizer is a global minimizer. [102] proves that $\mathcal{L}_G$ (5.15) is g-convex. Hence, any local minimizer of (5.15) is a global minimizer.

Algorithm 12: Riemannian gradient descent to minimize $\mathcal{L}_T$ (5.18)

Input: Data $\{\boldsymbol{s}_{ki}\}$, initialization $\theta^{(0)} \in \mathcal{SM}_{p,K}$
Output: $\theta^{(l)} \in \mathcal{SM}_{p,K}$
for $l = 0$ ***to convergence*** **do**
  Compute a step size α (see [1, Ch. 4]) and set
  $\theta^{(l+1)} = R_{\theta^{(l)}}^{\mathcal{SM}_{p,K}}\left(-\alpha \operatorname{grad}_{\mathcal{SM}_{p,K}} \mathcal{L}_T(\theta^{(l)})\right)$

5.3.2 . $\mathcal{SM}_{p,K}$: a geodesic submanifold of $\mathcal{M}_{p,K}$

In (5.17), $\mathcal{SM}_{p,K}$ is defined as a subset of $\mathcal{M}_{p,K}$. In fact, $\mathcal{SM}_{p,K}$ can even be turned into a Riemannian submanifold of $\mathcal{M}_{p,K}$. First of all, the tangent space of $\mathcal{SM}_{p,K}$ at θ is

$$T_\theta \mathcal{SM}_{p,K} = \Big\{\xi \in T_\theta \mathcal{M}_{p,K} : \operatorname{Tr}(\boldsymbol{A}^{-1}\boldsymbol{\xi}) = 0, \\ \operatorname{Tr}(\boldsymbol{A}_k^{-1}\boldsymbol{\xi}_k) = 0 \quad \forall k \in [\![1, K]\!]\Big\}. \quad (5.26)$$

By endowing $\mathcal{SM}_{p,K}$ with the Riemannian metric of $\mathcal{M}_{p,K}$, it becomes a Riemannian submanifold. $\forall \xi, \eta \in T_\theta \mathcal{SM}_{p,K}$ we have $\langle \xi, \eta \rangle_\theta^{\mathcal{SM}_{p,K}} = \langle \xi, \eta \rangle_\theta^{\mathcal{M}_{p,K}}$. The orthogonal projection from the ambient space onto the tangent space at θ is

$$P_\theta^{\mathcal{SM}_{p,K}}(\xi) = \left(P_{\boldsymbol{A}}^{\mathcal{SS}_p^{++}}(\boldsymbol{\xi}), \left\{P_{\boldsymbol{A}_k}^{\mathcal{SS}_p^{++}}(\boldsymbol{\xi}_k)\right\}\right), \quad (5.27)$$

where $P_{\boldsymbol{A}}^{\mathcal{SS}_p^{++}}(\boldsymbol{\xi}) = \operatorname{sym}(\boldsymbol{\xi}) - \frac{1}{p}\operatorname{Tr}\left(\boldsymbol{A}^{-1}\operatorname{sym}(\boldsymbol{\xi})\right)\boldsymbol{A}$. A remarkable result is that $\mathcal{SM}_{p,K}$ is a geodesic submanifold of $\mathcal{M}_{p,K}$, *i.e.*, the geodesics of $\mathcal{SM}_{p,K}$ are those of $\mathcal{M}_{p,K}$. It results that the exponential mapping on $\mathcal{SM}_{p,K}$ is $\exp_\theta^{\mathcal{SM}_{p,K}}(\xi) = \exp_\theta^{\mathcal{M}_{p,K}}(\xi)$. Then, for a loss function $h : \mathcal{SM}_{p,K} \to \mathbb{R}$, the Riemannian gradient at θ is

$$\operatorname{grad}_{\mathcal{SM}_{p,K}} h(\theta) = P_\theta^{\mathcal{SM}_{p,K}}\left(\boldsymbol{AGA}, \{\boldsymbol{A}_k\boldsymbol{G}_k\boldsymbol{A}_k\}\right), \quad (5.28)$$

where $(\boldsymbol{G}, \{\boldsymbol{G}_k\})$ is the classical Euclidean gradient of h at θ. Once again, a retraction that approximates the exponential mapping is leveraged to improve the numerical stability,

$$R_\theta^{\mathcal{SM}_{p,K}}(\xi) = \left(R_{\boldsymbol{A}}^{\mathcal{SS}_p^{++}}(\boldsymbol{\xi}), \left\{R_{\boldsymbol{A}_k}^{\mathcal{SS}_p^{++}}(\boldsymbol{\xi}_k)\right\}\right), \quad (5.29)$$

where $R_{\boldsymbol{A}}^{\mathcal{SS}_p^{++}}(\boldsymbol{\xi}) = \dfrac{\boldsymbol{A} + \boldsymbol{\xi} + \frac{1}{2}\boldsymbol{\xi}\boldsymbol{A}^{-1}\boldsymbol{\xi}}{\left|\boldsymbol{A} + \boldsymbol{\xi} + \frac{1}{2}\boldsymbol{\xi}\boldsymbol{A}^{-1}\boldsymbol{\xi}\right|^{\frac{1}{p}}}$.

Finally, $\mathcal{L}_T$ (5.18) is g-convex on $\mathcal{SM}_{p,K}$. Indeed, [102] proved that $\mathcal{L}_T$ is g-convex on $\mathcal{M}_{p,K}$ and $\mathcal{SM}_{p,K}$ is a geodesic submanifold of $\mathcal{M}_{p,K}$.

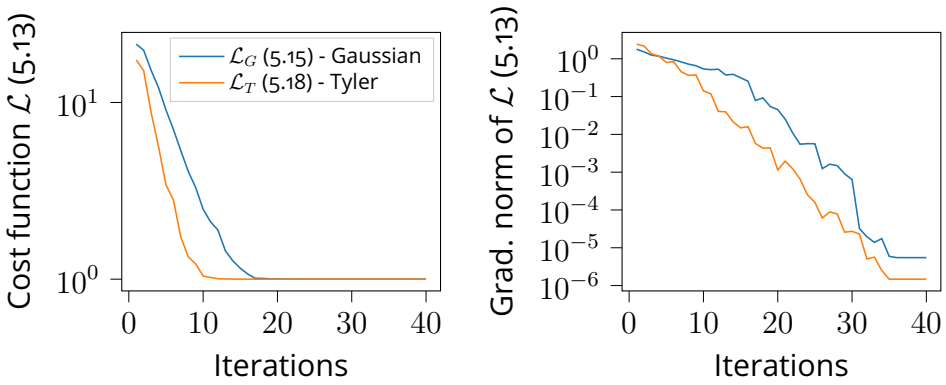


Figure 5.2: Left: Gaussian (5.15) and Tyler (5.18) costs functions with respect to the number of iterations of Algorithms 11 and 12 respectively. Right: Riemannian gradient norms of Gaussian (5.15) and Tyler (5.18) costs functions. The optimization is performed on the *Wine* dataset.

In this section, we exhibit a practical interest of the RGML method developed in Sections 5.2 and 5.3. All implementations of the following experiments are available at https://github.com/antoinecollas/robust_metric_learning. We apply it on real datasets from the *UCI machine learning repository* [51]. The three considered datasets are: *Wine*, *Vehicle*, and *Iris*. They are classification datasets, and their data dimensions along with their number of classes are presented in Table 5.1. These datasets are well balanced, *i.e.* they roughly have the same number of data for all the classes. The numbers of generated pairs in S and D are $n_S = n_D = 75\,K(K-1)$ (as in [50] and [146]).

The classification is done following a very classical protocol in metric learning.

1. A matrix $\boldsymbol{A}$ is estimated via a metric learning method.
2. The data $\{\boldsymbol{x}_l\}$ are multiplied by $\boldsymbol{A}^{-\frac{1}{2}}$ to get $\{\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{x}_l\}$.
3. The data $\{\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{x}_l\}$ are classified using a *k-nearest neighbors* with 5 neighbors.

Thus, the classification is performed using the Mahalanobis distance $d_{\boldsymbol{A}}$ defined by (5.1) in the Introduction. This classification is repeated 200 times via cross-validation. The proportion of the training/test sets is $50/50$. The error of classification is computed for each fold and the mean error is reported in Table 5.1. In order to show the robustness of the proposed method,

mislabeled data are introduced. To do so, we randomly select data in the training set whose labels are then randomly changed for new labels.

The implementations of the cross-validation as well as the k-nearest neighbors are from the scikit-learn library [111]. The proposed methods RGML Gaussian and RGML Tyler have been implemented using JAX [25]. The chosen value of parameter λ is 0.05. Its value has little impact on performance as long as it is neither too small nor too large. The proposed algorithms are compared to the classical metric learning algorithms: the identity matrix (called Euclidean in Table 5.1), the SCM computed on all the data, ITML [50], GMML [146], and LMNN [139]. The implementations of the metric-learn library [137] are used for the last three algorithms.

From Table 5.1, several observations are made. First of all, on the raw data (*i.e.* when the mislabeling rate is 0%) the RGML Gaussian is always the best performing algorithm among those tested. Also, the RGML Tyler always comes close with a maximum discrepancy of 0.26% versus the RGML Gaussian. Then, the RGML Tyler is the best performing algorithm when the mislabeling rate is 5% or 10%. When the mislabeling rate is 15%, RGML Tyler is the best performing algorithm for the Vehicle dataset and it is only beaten by ITML - Identity on the two other datasets. This shows the interest of considering robust cost functions such the Tyler's cost function (5.16) in the presence of poor labeling.

Finally, the RGML algorithms are fast. Indeed, Figure 5.2 shows that both RGML Gaussian and RGML Tyler converge in less than 20 iterations on the Wine dataset.

5.5 . Conclusions

This chapter has proposed to view some classical *metric learning* problems as covariance matrix estimation problems. From this point of view, the *RGML* optimization problem has been formalized. It aims at estimating regularized covariance matrices, in a robust manner, while computing their Riemannian mean. The formulation is broad and several more specific costs functions have been studied. The first one leverages the classical Gaussian likelihood and the second one the Tyler's cost function. In both cases, the *RGML* problem is g-convex and thus any local minimizer is a global one. Two Riemannian-based optimization algorithms are proposed to minimize these cost functions. Finally, the performance of the proposed approach is studied on several datasets. They improve the classification accuracy and are robust to mislabeled data.

	Wine $p = 13$, $n = 178$, $K = 3$				Vehicle $p = 18$, $n = 846$, $K = 4$				Iris $p = 4$, $n = 150$, $K = 3$			
Method	Mislabeling rate				Mislabeling rate				Mislabeling rate			
	0%	5%	10%	15%	0%	5%	10%	15%	0%	5%	10%	15%
Euclidean	30.12	30.40	31.40	32.40	38.27	38.58	39.46	40.35	3.93	4.47	5.31	**6.70**
SCM	10.03	11.62	13.70	17.57	23.59	24.27	25.24	26.51	12.57	13.38	14.93	16.68
ITML - Identity	3.12	4.15	5.40	**7.74**	24.21	23.91	24.77	26.03	3.04	4.47	5.31	**6.70**
ITML - SCM	2.45	4.76	6.71	10.25	23.86	23.82	24.89	26.30	3.05	13.38	14.92	16.67
GMML	2.16	3.58	5.71	9.86	21.43	22.49	23.58	25.11	2.60	5.61	9.30	12.62
LMNN	4.27	6.47	7.83	9.86	20.96	24.23	26.28	28.89	3.53	9.59	11.19	12.22
RGML Gaussian	**2.07**	**2.93**	5.15	9.20	**19.76**	21.19	22.52	24.21	**2.47**	5.10	8.90	12.73
RGML Tyler	**2.12**	**2.90**	**4.51**	8.31	19.90	**20.96**	**22.11**	**23.58**	**2.48**	**2.96**	**4.65**	7.83

Table 5.1: Misclassification errors on 3 datasets: Wine, Vehicle and Iris. Best results and those within 0.05% are in **bold**. The mislabeling rates indicate the percentage of labels that are randomly changed in the training set.

6 - Conclusions and perspectives

6.1 . Conclusions

This manuscript proposed new methods for statistical estimation and classification. They have been tested on remote sensing applications and have shown practical interests whether in terms of speed or precision. To do so, we began this manuscript with the description of a *clustering-classification pipeline*. The latter is based on statistical estimation and decomposes in three steps: *vectors extraction*, *features estimation* and *features clustering or classification*. The first step is a preprocessing step: data are transformed into batches to be clustered or classified. Then, the second step performs *statistical estimation*. Each batch of data is assumed to follow a parametrized statistical distribution. Classically, data are considered to be Gaussian with a known center. In this case, the estimated feature is the *SCM* which is the *MLE*. The third step consists of clustering or classifying these covariance matrices. To do so, we mentioned that, in the literature, the *Riemannian distance* as well as the *Riemannian center of mass*, both on the set of symmetric positive definite matrices, are often leveraged to implement *K-means++* or *Nearest centroïd classifier*. The objective of this manuscript was to go beyond this assumption of Gaussianity in steps two and three. Indeed, data are not necessarily Gaussian due to the presence of outliers (*e.g.* mislabeled data) or heavy tailed distributed data. Furthermore, data can be in high dimension which makes the classical estimation of the SCM ill posed.

To go beyond the Gaussian assumption, we leveraged the field of *robust statistics*, *i.e.* statistics that are robust to outliers and/or heavy tailed distributed data. Also, we considered structured covariance matrices as well as regularized models to handle high dimensional data. A first contribution was to propose new estimators for such statistics leveraging the theory of *Riemannian geometry*. Indeed, the parameter to estimate belongs to a constrained set that can be formalized as a *Riemannian manifold*. This formalization has many advantages: deriving estimators respecting constraints, flexibility in the optimizers (stochastic, second order, ...), geodesic convexity, ... Among all possible Riemannian manifolds, we focused on *statistical manifolds i.e.* manifolds endowed with the *FIM*. The latter tightly links the parameter space with the considered statistical model. Thus, we derived fast and scalable estimators that minimize negative log-likelihoods. A second contribution was the derivation of *ICRBs* to analyse the performance of estimators of structured covariance matrices. They lower bound the mean squared Riemannian distance between estimated parameters and the true one while taking into account constraints of the parameter space. Then,

the third axis focused on the third step by proposing new divergences. The latter measure the proximity between parameters and are associated to the considered statistical model. We also developed algorithms to compute the associated *Riemannian centers of mass* of parameters. These algorithms are Riemannian based optimizers in order to get fast and scalable estimators that respect the constraints of the parameter space. Finally, a forth contribution was the development of metric learning algorithms. These are different from the *clustering-classification pipeline* presented earlier since they directly operate on the raw data. Indeed, metric learning problems propose to learn a *Mahalanobis distance* such that data from a same class are close from each other whereas data that belong to different classes are far from each other. In this manuscript, we proposed geodesically convex problems, called *RGML*, that are solved efficiently using Riemannian optimization.

All these contributions have been tested on generated data as well as real datasets such as the *Indian pines image* and the large scale crop type mapping dataset *Breizhcrops* and show promising results.

6.2 . Perspectives

Throughout the manuscript choices have been made and many things remain to explore. A first perspective is the application of the metric learning algorithms we derived to more "richer" datasets. Indeed, to show the interest of RGML, we tested it on datasets from the *UCI repository*. These datasets are quite small, old and not related to remote sensing. Therefore, it should interesting to apply RGML to the bigger and newer dataset *Breizhcrops*. RGML is fast and thus this application should enforce this advantage of speed compared to other metric learning algorithms. If RGML is too slow, due to the big amount of data, it could be accelerated using a Riemannian stochastic gradient descent or one of its extensions with variance reduction [16, 147]. Others extensions of RGML are possible such as adding a low-rank structure to covariance matrices [63]. This should help to get the existence of solutions and faster optimization for RGML problems when data are in high dimensions.

A second perspective is to transform the proposed *clustering-classification pipeline* to a fully differentiable one. Indeed, with the advances in *geometric deep learning* [31, 70] and in the associated frameworks such as *JAX* [25], it becomes an increasingly practice to integrate every steps (preprocessing, statistical estimation and clustering-classification) in a single differentiable function. This has the advantage that each step can include parameters that are tuned with gradient descent to maximize the precision on the training set. For example, the preprocessing step can include a projection onto a learnable subspace (instead of a pre-defined PCA) or a learnable data time wrapping [48].

A third perspective is domain adaptation. We showed empirically that the proposed methods in Chapter 3 are robust to transformations of the test set. However, the experiments are limited to geometrical transformations (rotations, scaling factors, translations, ...). It would be valuable to investigate more realistic transformations such as a sensor change for hyperspectral or SAR images and see if results hold. This asks the question of, if a transformation is too strong, how to adapt the proposed *clustering-classification pipeline*. Domain adaptation [49, 150] is the field for these problems: given a test set with a distribution shift from the training set, how to adjust parameters of the pipeline, in an unsupervised manner, to account this shift. An idea would be to re-calebrate the distribution of estimated parameters $\{\theta_i\}$ such that the distribution on the test set is equal to the one on the training set.

7 - Résumé en français

Les systèmes de télédétection offrent une possibilité accrue d'enregistrer des images multitemporelles et multidimensionnelles de la surface de la terre en améliorant la résolution temporelle et spatiale. En effet, ces dernières années, de nombreux pays et entreprises ont déployé des satellites ou utilisé des drones pour l'observation de la terre. Les satellites Sentinel, Landsat et TerraSAR-X ou l'UAVSAR sont des exemples de ces instruments de télédétection. Cette forte augmentation du nombre, de la performance et de la diversité de ces systèmes permet le développement de nombreuses applications telles que la surveillance de l'environnement (par exemple, les glaciers, les forêts, l'urbanisme), les événements majeurs (par exemple, les tremblements de terre, les inondations), l'activité humaine (par exemple, la surveillance maritime et des frontières) ainsi que les prévisions météorologiques. Ces opportunités augmentent considérablement l'intérêt des outils de traitement de données basés sur des séries temporelles d'images multivariées.

Une tendance récente de l'apprentissage automatique, provenant principalement de la communauté EEG/MEG (Electroencephalography / Magnetoencephalography), propose d'estimer les matrices de covariance des données et ensuite de les classer en utilisant la géométrie riemannienne. En effet, la théorie de la géométrie riemannienne et son sous-domaine, la géométrie de l'information, s'adapte bien aux matrices de covariance qui sont alors vues comme des paramètres de distributions gaussiennes multivariées centrées. Dans ce cas, la ligne droite classique est remplacée par des géodésiques, la distance euclidienne par des distances riemanniennes et la moyenne arithmétique par des centres de masse riemanniens. En pratique, l'utilisation de la géométrie riemanienne donne de bien meilleures performances que sa contrepartie euclidienne lorsqu'on traite des matrices de covariance. Dans cette thèse, nous proposons d'appliquer ce pipeline de regroupement-classification aux données de télédétection et de l'étendre de multiples façons. Les contributions sont de quatre ordres.

Premièrement, des estimateurs de statistiques robustes sont développés en s'appuyant sur la théorie de l'optimisation sur les variétés riemanniennes. En particulier, des méthodes de descente de gradient sont développées pour estimer conjointement les localisations (centres de distributions) et les matrices de covariance, ainsi que pour estimer des matrices de covariance structurées à partir de données de haute dimension. Ces estimateurs sont rapides et conviennent bien aux ensembles de données de grande échelle.

Deuxièmement, des bornes intrinsèques de Cramér-Rao (ICRB) sont dérivées pour analyser la performance des estimateurs de matrices de covariance structurées. Ces ICRB bornent les moyennes de distances rieman-

niennes au carré entre les paramètres estimés et les vrais paramètres. Ceci permet de prendre en compte les contraintes de l'espace des paramètres.

Troisièmement, des divergences entre les statistiques et leurs centres de masse associés sont proposés. Ces divergences, et les centres de masse associés, sont choisis par rapport au modèle statistique pour obtenir de meilleures performances en pratique. De plus, des algorithmes d'optimisation riemannienne basés sur le gradient sont développés pour calculer efficacement ces centres de masse.

Une quatrième contribution est le développement d'algorithmes d'apprentissage de distances. Les méthodes d'apprentissage de distances proposent de regrouper ou de classer des données brutes à l'aide d'une distance de Mahalanobis apprise. Dans cette thèse, nous démontrons que certains problèmes classiques d'apprentissage de distances peuvent être considérés comme des problèmes d'estimation de covariance. Avec cette nouvelle vision, nous dérivons deux nouveaux algorithmes d'optimisation riemannienne pour l'apprentissage de distances. Toutes ces contributions sont testées sur des données générées ainsi que sur des jeux de données réels tels que l'image de *Indian pines* et le jeu de données de cartographie de type de culture à *Breizhcrops* et montrent des résultats prometteurs.

Bibliography

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ, USA: Princeton University Press, 2008. isbn: 978-0-691-13298-3.

[2] P.-A. Absil, R. Mahony, and R. Sepulchre. "Riemannian geometry of Grassmann manifolds with a view on algorithmic computation". In: *Acta Applicandae Mathematica* 80.2 (2004), pp. 199–220.

[3] S. Amari. *Information geometry and its applications*. Vol. 194. Springer, 2016.

[4] S. Amari. "Natural Gradient Works Efficiently in Learning". In: *Neural Computation* 10.2 (1998), pp. 251–276.

[5] M. Arnaudon, F. Barbaresco, and L. Yang. "Medians and Means in Riemannian Geometry: Existence, Uniqueness and Computation". In: *Matrix Information Geometry*. Ed. by Frank Nielsen and Rajendra Bhatia. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 169–197. isbn: 978-3-642-30232-9. url: https://doi.org/10.1007/978-3-642-30232-9_8.

[6] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. "Geometric Means in a Novel Vector Space Structure on Symmetric Positive-Definite Matrices". In: *SIAM Journal on Matrix Analysis and Applications* 29.1 (2007), pp. 328–347.

[7] D. Arthur and S. Vassilvitskii. "K-Means++: The Advantages of Careful Seeding". In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '07. New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035. isbn: 9780898716245.

[8] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten. "Multiclass Brain–Computer Interface Classification by Riemannian Geometry". In: *IEEE Transactions on Biomedical Engineering* 59.4 (2012), pp. 920–928.

[9] M. F. Baumgardner, L. L. Biehl, and D. A. Landgrebe. *220 Band AVIRIS Hyperspectral Image Data Set: June 12, 1992 Indian Pine Test Site 3*. 2015.

[10] R. Ben Abdallah, A. Breloy, M.N. El Korso, and D. Lautru. "Bayesian signal subspace estimation with compound Gaussian sources". In: *Signal Processing* 167 (2020), p. 107310. issn: 0165-1684.

[11] T. Bendokat, R. Zimmermann, and P. -A. Absil. *A Grassmann Manifold Handbook: Basic Geometry and Computational Aspects*. 2020.

[12] M. A. Bendoumi, M. He, and S. Mei. "Hyperspectral Image Resolution Enhancement Using High-Resolution Multispectral Image Based on Spectral Unmixing". In: *IEEE Transactions on Geoscience and Remote Sensing* 52.10 (2014), pp. 6574–6583.

[13] O. Besson. "Bounds for a Mixture of Low-Rank Compound-Gaussian and White Gaussian Noises". In: *IEEE Transactions on Signal Processing* 64.21 (2016), pp. 5723–5732.

[14] R. Bhatia. *Positive Definite Matrices*. Princeton University Press, 2007. isbn: 9780691129181.

[15] R. Bhatia. *Positive Definite Matrices*. USA: Princeton University Press, 2015. isbn: 0691168253.

[16] S. Bonnabel. "Stochastic Gradient Descent on Riemannian Manifolds". In: *IEEE Transactions on Automatic Control* 58.9 (2013), pp. 2217–2229.

[17] F. Bouchard, A. Breloy, G. Ginolhac, A. Renaux, and F. Pascal. "A Riemannian Framework for Low-Rank Structured Elliptical Models". In: *IEEE Transactions on Signal Processing* 69 (2021), pp. 1185–1199.

[18] F. Bouchard, A. Mian, J. Zhou, S. Said, G. Ginolhac, and Y. Berthoumieu. "Riemannian geometry for compound Gaussian distributions: Application to recursive change detection". In: *Signal Processing* 176 (2020), p. 107716.

[19] N. Boumal. *An introduction to optimization on smooth manifolds*. To appear with Cambridge University Press. 2022.

[20] N. Boumal. "On intrinsic Cramér-Rao bounds for Riemannian submanifolds and quotient manifolds". In: *IEEE transactions on signal processing* 61.7 (2013), pp. 1809–1821.

[21] N. Boumal. "Optimization and estimation on manifolds". PhD thesis. Université catholique de Louvain, 2014.

[22] N. Boumal and P.-A. Absil. "Low-rank matrix completion via preconditioned optimization on the Grassmann manifold". In: *Linear Algebra and its Applications* 475 (2015), pp. 200–239. issn: 0024-3795.

[23] N. Boumal and P.-A. Absil. “RTRMC: A Riemannian trust-region method for low-rank matrix completion”. In: *Advances in Neural Information Processing Systems*. Ed. by J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger. Vol. 24. Curran Associates, Inc., 2011.

[24] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. “Manopt, a Matlab Toolbox for Optimization on Manifolds”. In: *Journal of Machine Learning Research* 15 (2014), pp. 1455–1459.

[25] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. *JAX: composable transformations of Python+NumPy programs*. Version 0.2.5. 2018.

[26] A. Breloy, G. Ginolhac, F. Pascal, and P. Forster. “Clutter Subspace Estimation in Low Rank Heterogeneous Noise Context”. In: *IEEE Transactions on Signal Processing* 63.9 (2015), pp. 2173–2182.

[27] A. Breloy, G. Ginolhac, A. Renaux, and F. Bouchard. “Intrinsic Cramér–Rao Bounds for Scatter and Shape Matrices Estimation in CES Distributions”. In: *IEEE Signal Processing Letters* 26.2 (2019), pp. 262–266.

[28] A. Breloy, L. Le Magoarou, G. Ginolhac, F. Pascal, and P. Forster. “Maximum likelihood estimation of clutter subspace in non homogeneous noise context”. In: *21st European Signal Processing Conference (EUSIPCO 2013)*. 2013, pp. 1–5.

[29] A. Breloy, E. Ollila, and F. Pascal. “Spectral Shrinkage of Tyler’s M-Estimator of Covariance Matrix”. In: *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. IEEE. 2019, pp. 535–538.

[30] A. Breloy, Y. Sun, P. Babu, G. Ginolhac, D. P. Palomar, and F. Pascal. “A robust signal subspace estimator”. In: *2016 IEEE Statistical Signal Processing Workshop (SSP)*. 2016, pp. 1–4.

[31] M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. “Geometric deep learning: going beyond euclidean data”. In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 18–42.

[32] M. Calvo and J. M. Oller. “AN EXPLICIT SOLUTION OF INFORMATION GEODESIC EQUATIONS FOR THE MULTIVARIATE NORMAL MODEL”. In: *Statistics & Risk Modeling* 9.1-2 (1991), pp. 119–138.

[33] S. Cambanis, S. Huang, and G. Simons. “On the theory of elliptically contoured distributions”. In: *Journal of Multivariate Analysis* 11.3 (1981), pp. 368–385.

[34] J.-F. Cardoso. “Blind signal separation: statistical principles”. In: *Proceedings of the IEEE* 86.10 (1998), pp. 2009–2025.

[35] T. N. Carlson and D. A. Ripley. “On the relation between NDVI, fractional vegetation cover, and leaf area index”. In: *Remote sensing of Environment* 62.3 (1997), pp. 241–252.

[36] T. Chen, E. Martin, and G. Montague. “Robust probabilistic PCA with missing data and contribution analysis for outlier detection”. In: *Computational Statistics & Data Analysis* 53.10 (2009), pp. 3706–3716. issn: 0167-9473.

[37] Y. Chen, A. Wiesel, Y. C. Eldar, and A. O. Hero. “Shrinkage algorithms for MMSE covariance estimation”. In: *IEEE Transactions on Signal Processing* 58.10 (2010), pp. 5016–5029.

[38] S. Chevallier, E. K. Kalunga, Q. Barthélemy, and E. Monacelli. “Review of Riemannian distances and divergences, applied to SSVEP-based BCI”. In: *Neuroinformatics* 19.1 (2021), pp. 93–106.

[39] A. Collas, F. Bouchard, A. Breloy, G. Ginolhac, C. Ren, and J.-P. Ovarlez. “Probabilistic PCA From Heteroscedastic Signals: Geometric Framework and Application to Clustering”. In: *IEEE Transactions on Signal Processing* 69 (2021), pp. 6546–6560.

[40] A. Collas, F. Bouchard, A. Breloy, C. Ren, G. Ginolhac, and J.-P. Ovarlez. “A Tyler-type estimator of location and scatter leveraging Riemannian optimization”. In: *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Toronto, Canada, June 2021.

[41] A. Collas, F. Bouchard, G. Ginolhac, A. Breloy, C. Ren, and J.-P. Ovarlez. “On the Use of Geodesic Triangles between Gaussian Distributions for Classification Problems”. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022, pp. 5697–5701.

[42] A. Collas, A. Breloy, G. Ginolhac, C. Ren, and J.-P. Ovarlez. “Apprentissage robuste de distance par géométrie riemannienne”. In: *GRETSI 2022 XXVIIIème colloque*. Nancy, France, Sept. 2022.

[43] A. Collas, A. Breloy, G. Ginolhac, C. Ren, and J.-P. Ovarlez. “Robust Geometric Metric Learning”. In: *2022 30th European Signal Processing Conference (EUSIPCO)*. Belgrade, Serbia, Sept. 2022.

[44] A. Collas, A. Breloy, C. Ren, G. Ginolhac, and J.-P. Ovarlez. “Riemannian optimization for non-centered mixture of scaled Gaussian distributions”. In: arXiv, 2022.

[45] M. Congedo, A. Barachant, and R. Bhatia. “Riemannian geometry for EEG-based brain-computer interfaces; a primer and a review”. In: *Brain-Computer Interfaces* 4.3 (2017), pp. 155–174.

[46] E. Conte, A. De Maio, and G. Ricci. “Recursive estimation of the covariance matrix of a compound-Gaussian process and its application to adaptive CFAR detection”. In: *IEEE Transactions on Signal Processing* 50.8 (2002), pp. 1908–1915.

[47] H. Cramér. *Mathematical methods of statistics*. 1946.

[48] M. Cuturi and M. Blondel. “Soft-dtw: a differentiable loss function for time-series”. In: *International conference on machine learning*. PMLR. 2017, pp. 894–903.

[49] H. Daume III and D. Marcu. “Domain adaptation for statistical classifiers”. In: *Journal of artificial Intelligence research* 26 (2006), pp. 101–126.

[50] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. “Information-Theoretic Metric Learning”. In: *Proceedings of the 24th International Conference on Machine Learning*. ICML ’07. Corvalis, Oregon, USA: Association for Computing Machinery, 2007, pp. 209–216. isbn: 9781595937933.

[51] D. Dua and C. Graff. *UCI Machine Learning Repository*. 2017.

[52] A. Edelman, T.A. Arias, and S. T. Smith. “The geometry of algorithms with orthogonality constraints”. In: *SIAM journal on Matrix Analysis and Applications* 20.2 (1998), pp. 303–353.

[53] P. T. Fletcher and S. Joshi. “Riemannian geometry for the statistical analysis of diffusion tensor data”. In: *Signal Processing* 87.2 (2007). Tensor Signal Processing, pp. 250–262. issn: 0165-1684.

[54] P. Formont, J.-P. Ovarlez, and F. Pascal. “On the use of matrix information geometry for polarimetric SAR image classification”. In: *Matrix Information Geometry*. Springer, 2013, pp. 257–276.

[55] G. Frahm and U. Jaekel. “Tyler’s M-estimator, random matrix theory, and generalized elliptical distributions with applications to finance”. In: *Random Matrix Theory, and Generalized Elliptical Distributions with Applications to Finance (October 21, 2008)* (2008).

[56] W. C. Franks and A. Moitra. “Rigorous guarantees for Tyler’s M-estimator via quantum expansion”. In: *Conference on Learning Theory*. PMLR. 2020, pp. 1601–1632.

[57] J. Friedman, T. Hastie, and R. Tibshirani. “Sparse inverse covariance estimation with the graphical lasso”. In: *Biostatistics* 9.3 (Dec. 2007), pp. 432–441. issn: 1465-4644.

[58] J. Frontera-Pons, M. A. Veganzones, F. Pascal, and J.-P. Ovarlez. “Hyperspectral Anomaly Detectors Using Robust Estimators”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 9.2 (2016), pp. 720–731.

[59] A. F. H. Goetz, G. Vane, J. E. Solomon, and B. N. Rock. “Imaging spectrometry for earth remote sensing”. In: *science* 228.4704 (1985), pp. 1147–1153.

[60] J. D. Gorman and A. Hero. “Lower bounds for parametric estimation with constraints”. In: *IEEE Transactions on Information Theory* 36.6 (1990), pp. 1285–1301.

[61] W. W. Hager and H. Zhang. “A survey of nonlinear conjugate gradient methods”. In: *Pacific journal of Optimization* 2.1 (2006), pp. 35–58.

[62] M. Harandi, R. Hartley, M. Salzmann, and J. Trumpf. “Dictionary Learning on Grassmann Manifolds”. In: *Algorithmic Advances in Riemannian Geometry and Applications: For Machine Learning, Computer Vision, Statistics, and Optimization*. Ed. by Hà Quang Minh and Vittorio Murino. Cham: Springer International Publishing, 2016, pp. 145–172. isbn: 978-3-319-45026-1.

[63] M. Harandi, M. Salzmann, and R. Hartley. “Joint Dimensionality Reduction and Metric Learning: A Geometric Take”. In: *Proceedings of the 34th International Conference on Machine Learning*. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 1404–1413.

[64] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, 2009.

[65] D. Hong, L. Balzano, and J. A. Fessler. “Probabilistic PCA for Heteroscedastic Data”. In: *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. 2019, pp. 26–30.

[66] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu. “An Augmented Linear Mixing Model to Address Spectral Variability for Hyperspectral Unmixing”. In: *IEEE Transactions on Image Processing* 28.4 (2019), pp. 1923–1938.

[67] R. Hosseini and S. Sra. “An alternative to EM for Gaussian mixture models: batch and stochastic Riemannian optimization”. In: *Math. Program.* 181 (2020), pp. 187–223.

[68] R. Hosseini and S. Sra. “Matrix Manifold Optimization for Gaussian Mixtures”. In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc., 2015.

[69] Z. Huang, R. Wang, S. Shan, and X. Chen. “Projection Metric Learning on Grassmann Manifold with Application to Video based Face Recognition”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 140–149.

[70] Zhiwu Huang and Luc Van Gool. “A riemannian network for spd matrix learning”. In: *Thirty-first AAAI conference on artificial intelligence*. 2017.

[71] P. J. Huber. “Robust statistics”. In: *International encyclopedia of statistical science*. Springer, 2011, pp. 1248–1251.

[72] I. Jolliffe. “Principal Component Analysis”. In: Springer, 2011.

[73] M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre. “Generalized Power Method for Sparse Principal Component Analysis”. In: *Journal of Machine Learning Research* 11.15 (2010), pp. 517–553.

[74] F. Kai-Tai and Z. Yao-Ting. *Generalized multivariate analysis*. Vol. 19. Science Press Beijing and Springer-Verlag, Berlin, 1990.

[75] H. Karcher. “Riemannian center of mass and mollifier smoothing”. In: *Communications on Pure and Applied Mathematics* 30.5 (1977), pp. 509–541.

[76] S. M. Kay. *Fundamentals of statistical signal processing: estimation theory*. Prentice-Hall, Inc., 1993.

[77] N. Keshava and J.F. Mustard. “Spectral unmixing”. In: *IEEE Signal Processing Magazine* 19.1 (2002), pp. 44–57.

[78] O. Ledoit and M. Wolf. “A well-conditioned estimator for large-dimensional covariance matrices”. In: *Journal of multivariate analysis* 88.2 (2004), pp. 365–411.

[79] D. Maclaurin, D. Duvenaud, and Adams R.P. “Autograd: Effortless Gradients in Pure Numpy”. In: *AutoML workshop ICML* (2015).

[80] J. Macqueen. “Some methods for classification and analysis of multivariate observations”. In: *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*. 1967, pp. 281–297.

[81] M. Mahajan, P. Nimbhorkar, and K. Varadarajan. “The Planar k-Means Problem is NP-Hard”. In: *WALCOM: Algorithms and Computation*. Ed. by Sandip Das and Ryuhei Uehara. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 274–285. isbn: 978-3-642-00202-1.

[82] D. Manolakis, E. Truslow, M. Pieper, T. Cooley, and M. Brueggeman. “Detection Algorithms in Hyperspectral Imaging Systems: An Overview of Practical Algorithms”. In: *IEEE Signal Processing Magazine* 31.1 (2014), pp. 24–33.

[83] D. G. Manolakis, D. Marden, J. P. Kerekes, and G. A. Shaw. “Statistics of hyperspectral imaging data”. In: *Algorithms for Multispectral, Hyperspectral, and Ultraspectral Imagery VII*. Ed. by Sylvia S. Shen and Michael R. Descour. Vol. 4381. International Society for Optics and Photonics. SPIE, 2001, pp. 308–316. doi: 10.1117/12.437021. url: https://doi.org/10.1117/12.437021.

[84] D. G. Manolakis, D. Marden, J. P. Kerekes, and G. A. Shaw. “Statistics of hyperspectral imaging data”. In: *Algorithms for Multispectral, Hyperspectral, and Ultraspectral Imagery VII*. Ed. by Sylvia S. Shen and Michael R. Descour. Vol. 4381. International Society for Optics and Photonics. SPIE, 2001, pp. 308–316.

[85] J. H. Manton. “Optimization algorithms exploiting unitary constraints”. In: *IEEE Transactions on Signal Processing* 50.3 (2002), pp. 635–650.

[86] R. A. Maronna. “Robust M-estimators of multivariate location and scatter”. In: *The annals of statistics* (1976), pp. 51–67.

[87] J. Martens. “New Insights and Perspectives on the Natural Gradient Method”. In: *Journal of Machine Learning Research* 21.146 (2020), pp. 1–76.

[88] T. Maunu, T. Zhang, and G. Lerman. “A Well-Tempered Landscape for Non-convex Robust Subspace Recovery”. In: *Journal of Machine Learning Research* 20.37 (2019), pp. 1–59.

[89] F. Mezzadri. “How to generate random matrices from the classical compact groups”. In: *Notices of the AMS* (2006).

[90] A. Mian, A. Collas, A. Breloy, G. Ginolhac, and J-P. Ovarlez. “Robust Low-Rank Change Detection for Multivariate SAR Image Time Series”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13 (2020), pp. 3545–3556.

[91] A. Mian, G. Ginolhac, J. Ovarlez, and A. M. Atto. “New Robust Statistics for Change Detection in Time Series of Multivariate SAR Images”. In: *IEEE Transactions on Signal Processing* 67.2 (2019), pp. 520–534.

[92] T. Minka. “Automatic choice of dimensionality for PCA”. In: *Advances in neural information processing systems* 13 (2000).

[93] B. Mishra, N. T. V. Satyadev, H. Kasai, and P. Jawanpuria. *Manifold optimization for non-linear optimal transport problems*. 2021.

[94] M. Moakher. “A Differential Geometric Approach to the Geometric Mean of Symmetric Positive-Definite Matrices”. In: *SIAM Journal on Matrix Analysis and Applications* 26.3 (2005), pp. 735–747.

[95] T. J. Moore, R. J. Kozick, and B. M. Sadler. “The constrained Cramér–Rao bound from the perspective of fitting a model”. In: *IEEE Signal Processing Letters* 14.8 (2007), pp. 564–567.

[96] S. Neumayer, M. Nimmer, S. Setzer, and G. Steidl. “On the rotational invariant L1-norm PCA”. In: *Linear Algebra and its Applications* 587 (2020), pp. 243–270. issn: 0024-3795.

[97] F. Nielsen. “The many faces of information geometry”. In: *Not. Am. Math. Soc* 69 (2022), pp. 36–45.

[98] F. Nielsen and R. Nock. *Total Jensen divergences: Definition, Properties and k-Means++ Clustering*. 2013.

[99] E. Nitzan, T. Routtenberg, and J. Tabrikian. “Cramér–Rao Bound for Constrained Parameter Estimation Using Lehmann-Unbiasedness”. In: *IEEE Transactions on Signal Processing* 67.3 (2018), pp. 753–768.

[100] E. Ollila, D. P. Palomar, and F. Pascal. “Shrinking the eigenvalues of M-estimators of covariance matrix”. In: *IEEE Transactions on Signal Processing* 69 (2020), pp. 256–269.

[101] E. Ollila and E. Raninen. “Optimal Shrinkage Covariance Matrix Estimation Under Random Sampling From Elliptical Distributions”. In: *IEEE Transactions on Signal Processing* 67.10 (2019), pp. 2707–2719.

[102] E. Ollila, I. Soloveychik, D. E. Tyler, and A. Wiesel. *Simultaneous penalized M-estimation of covariance matrices using geodesically convex optimization*. 2016.

[103] E. Ollila and D. E. Tyler. “Regularized M-estimators of scatter matrix”. In: *IEEE Transactions on Signal Processing* 62.22 (2014), pp. 6059–6070.

[104] E. Ollila, D. E. Tyler, V. Koivunen, and H. V. Poor. “Complex Elliptically Symmetric Distributions: Survey, New Results and Applications”. In: *IEEE Transactions on Signal Processing* 60.11 (2012), pp. 5597–5625.

[105] E. Ollila, D. E. Tyler, V. Koivunen, and H. V. Poor. “Compound-Gaussian Clutter Modeling With an Inverse Gaussian Texture Distribution”. In: *IEEE Signal Processing Letters* 19.12 (2012), pp. 876–879.

[106] J.-P. Ovarlez, G. Ginolhac, and A. M. Atto. “Multivariate Linear Time-Frequency modeling and adaptive robust target detection in highly textured monovariate SAR image”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017, pp. 4029–4033.

[107] J.-P. Ovarlez, F. Pascal, and P. Forster. “Covariance Matrix Estimation in SIRV and Elliptical Processes and Their Applications in Radar Detection”. In: *Modern Radar Detection Theory*. Ed. by IET. chapter 8. Scitech Publishing, 2015, pp. 295–332.

[108] F. Pascal, Y. Chitour, J.-P. Ovarlez, P. Forster, and P. Larzabal. “Covariance structure maximum-likelihood estimates in compound Gaussian noise: Existence and algorithm analysis”. In: *IEEE Transactions on Signal Processing* 56.1 (2007), pp. 34–48.

[109] F. Pascal, Y. Chitour, and Y. Quek. “Generalized robust shrinkage estimator and its application to STAP detection problem”. In: *IEEE Transactions on Signal Processing* 62.21 (2014), pp. 5640–5651.

[110] F. Pascal, P. Forster, J.-P. Ovarlez, and P. Larzabal. “Performance analysis of covariance matrix estimates in impulsive noise”. In: *IEEE Transactions on signal processing* 56.6 (2008), pp. 2206–2217.

[111] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[112] X. Pennec. “Statistical computing on manifolds: from Riemannian geometry to computational anatomy”. In: *Emerging Trends in Visual Computing*. Ed. by Frank Nielsen. Vol. 5416. LNCS. Springer, 2008, pp. 347–386.

[113] X. Pennec, P. Fillard, and N. Ayache. “A Riemannian framework for tensor computing”. In: *International Journal of computer vision* 66.1 (2006), pp. 41–66.

[114] C. Qi, K. A. Gallivan, and P.-A. Absil. “An efficient BFGS algorithm for Riemannian optimization”. In: *Proceedings of the 19th International Symposium on Mathematical Theory of Network and Systems (MTNS 2010)*. Vol. 1. 2010, pp. 2221–2227.

[115] R. S. Raghavan. “Statistical Interpretation of a Data Adaptive Clutter Subspace Estimation Algorithm”. In: *IEEE Transactions on Aerospace and Electronic Systems* 48.2 (2012), pp. 1370–1384.

[116] C. R. Rao. “Information and the accuracy attainable in the estimation of statistical parameters”. In: *Reson. J. Sci. Educ* 20 (1945), pp. 78–90.

[117] I. S. Reed and X. Yu. “Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 38.10 (1990), pp. 1760–1770.

[118] M. Rußwurm, C. Pelletier, M. Zollner, S. Lefèvre, and M. Körner. “BreizhCrops: A Time Series Dataset for Crop Type Mapping”. In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences ISPRS (2020)* XLIII-B2-2020 (2020), pp. 1545–1551.

[119] Y.E. Shimabukuro and J.A. Smith. “The least-squares mixing models to generate fraction images derived from remote sensing multispectral data”. In: *IEEE Transactions on Geoscience and Remote Sensing* 29.1 (1991), pp. 16–20.

[120] L. T. Skovgaard. “A Riemannian Geometry of the Multivariate Normal Model”. In: *Scandinavian Journal of Statistics* 11.4 (1984), pp. 211–223. issn: 03036898, 14679469.

[121] S. T. Smith. “Covariance, subspace, and intrinsic Cramér-Rao bounds”. In: *IEEE Transactions on Signal Processing* 53.5 (2005), pp. 1610–1630. issn: 1941-0476.

[122] S. Sra and R. Hosseini. “Conic Geometric Optimization on the Manifold of Positive Definite Matrices”. In: *SIAM Journal on Optimization* 25.1 (2015), pp. 713–739.

[123] P. Stoica and B. C. Ng. “On the Cramér-Rao bound under parametric constraints”. In: *IEEE Signal Processing Letters* 5.7 (1998), pp. 177–179.

[124] J. L. Suárez, S. García, and F. Herrera. “A tutorial on distance metric learning: Mathematical foundations, algorithms, experimental analysis, prospects and challenges”. In: *Neurocomputing* 425 (2021), pp. 300–322. issn: 0925-2312.

[125] Y. Sun, P. Babu, and D. P. Palomar. "Regularized robust estimation of mean and covariance matrix under heavy-tailed distributions". In: *IEEE Transactions on Signal Processing* 63.12 (2015), pp. 3096–3109.

[126] Y. Sun, P. Babu, and D. P. Palomar. "Regularized Tyler's Scatter Estimator: Existence, Uniqueness, and Algorithms". In: *IEEE Transactions on Signal Processing* 62.19 (2014), pp. 5143–5156.

[127] Y. Sun, P. Babu, and D. P. Palomar. "Robust Estimation of Structured Covariance Matrix for Heavy-Tailed Elliptical Distributions". In: *IEEE Transactions on Signal Processing* 64.14 (2016), pp. 3576–3590.

[128] Y. Sun, A. Breloy, P. Babu, D. P. Palomar, F. Pascal, and G. Ginolhac. "Low-Complexity Algorithms for Low Rank Clutter Parameters Estimation in Radar Systems". In: *IEEE Transactions on Signal Processing* 64.8 (2016), pp. 1986–1998.

[129] M. Tang, Y. Rong, J. Zhou, and X. Li. "Information Geometric Approach to Multisensor Estimation Fusion". In: *IEEE Transactions on Signal Processing* 67.2 (2019), pp. 279–292.

[130] Y. Thanwerdas and X. Pennec. *O(n)-invariant Riemannian metrics on SPD matrices*. 2021.

[131] M. E. Tipping and C. M. Bishop. "Probabilistic principal component analysis". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.3 (1999), pp. 611–622.

[132] J. Townsend, N. Koep, and S. Weichwald. "Pymanopt: A Python Toolbox for Optimization on Manifolds Using Automatic Differentiation". In: *J. Mach. Learn. Res.* 17.1 (Jan. 2016), pp. 4755–4759. issn: 1532-4435.

[133] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. "Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.11 (2011), pp. 2273–2286.

[134] O. Tuzel, F. Porikli, and P. Meer. "Human Detection via Classification on Riemannian Manifolds". In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. 2007, pp. 1–8.

[135] O. Tuzel, F. Porikli, and P. Meer. "Pedestrian detection via classification on Riemannian manifolds". In: *IEEE transactions on pattern analysis and machine intelligence* 30.10 (2008), pp. 1713–1727.

[136] D. E. Tyler. "A Distribution-Free M-Estimator of Multivariate Scatter". In: *The Annals of Statistics* 15.1 (1987), pp. 234–251.

[137] W. de Vazelhes, CJ Carey, Y. Tang, N. Vauquier, and A. Bellet. "metric-learn: Metric Learning Algorithms in Python". In: *Journal of Machine Learning Research* 21.138 (2020), pp. 1–6.

[138] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, Eric W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272.

[139] K. Q. Weinberger and L. K. Saul. "Distance metric learning for large margin nearest neighbor classification". In: *The Journal of Machine Learning Research* 10 (2009), pp. 207–244.

[140] A. Wiesel. "Geodesic convexity and covariance estimation". In: *IEEE transactions on signal processing* 60.12 (2012), pp. 6182–6189.

[141] A. Wiesel. "Regularized covariance estimation in scaled Gaussian models". In: *2011 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. 2011, pp. 309–312.

[142] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. "Distance Metric Learning, With Application To Clustering With Side-Information". In: *Advances in Neural Information Processing Systems 15*. MIT Press, 2003, pp. 505–512.

[143] O. Yamaguchi, K. Fukui, and K.-i. Maeda. "Face recognition using temporal image sequence". In: *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*. 1998, pp. 318–323.

[144] M. Yi and D. E. Tyler. "Shrinking the covariance matrix using convex penalties on the matrix-log transformation". In: *Journal of Computational and Graphical Statistics* 30.2 (2020), pp. 442–451.

[145] X. Yuan, W. Huang, P.-A. Absil, and K. A. Gallivan. "Averaging Symmetric Positive-Definite Matrices". In: *Handbook of Variational Methods for Nonlinear Geometric Data*. Ed. by Philipp Grohs, Martin Holler, and Andreas Weinmann. Cham: Springer International Publishing, 2020, pp. 555–575. isbn: 978-3-030-31351-7.

[146] P. Zadeh, R. Hosseini, and S. Sra. “Geometric Mean Metric Learning”. In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 2016, pp. 2464–2471.

[147] H. Zhang, S. J. Reddi, and S. Sra. “Riemannian SVRG: Fast Stochastic Optimization on Riemannian Manifolds”. In: *Advances in Neural Information Processing Systems 29 (NIPS 2016)*. 2016.

[148] H. Zhang and S. Sra. “First-order Methods for Geodesically Convex Optimization”. In: *29th Annual Conference on Learning Theory*. Ed. by Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir. Vol. 49. Proceedings of Machine Learning Research. Columbia University, New York, New York, USA: PMLR, 2016, pp. 1617–1638.

[149] T. Zhang, X. Cheng, and A. Singer. “Marčenko–Pastur law for Tyler’s M-estimator”. In: *Journal of Multivariate Analysis* 149 (2016), pp. 114–123.

[150] Y. Zhang, T. Liu, M. Long, and M. Jordan. “Bridging Theory and Algorithm for Domain Adaptation”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 7404–7413.

[151] Y. Zhang, Q. Qu, and J. Wright. “From Symmetry to Geometry: Tractable Nonconvex Problems”. In: *arXiv preprint arXiv:2007.06753* (2020).

[152] J. Zhou and S. Said. “Fast, Asymptotically Efficient, Recursive Estimation in a Riemannian Manifold”. In: *Entropy* 21.10 (2019).

[153] H. Zou, T. Hastie, and R. Tibshirani. “Sparse Principal Component Analysis”. In: *Journal of Computational and Graphical Statistics* 15.2 (2006), pp. 265–286.