






Riemannian Optimization for Non-Centered Mixture of Scaled Gaussian Distributions

Antoine Collas , Arnaud Breloy , *Member, IEEE*, Chengfang Ren , *Member, IEEE*, Guillaume Ginolhac , *Senior Member, IEEE*, and Jean-Philippe Ovarlez , *Member, IEEE*

Abstract—This article studies the statistical model of the non-centered mixture of scaled Gaussian distributions (NC-MSG). Using the Fisher-Rao information geometry associated with this distribution, we derive a Riemannian gradient descent algorithm. This algorithm is leveraged for two minimization problems. The first is the minimization of a regularized negative log-likelihood (NLL). The latter makes the trade-off between a white Gaussian distribution and the NC-MSG. Conditions on the regularization are given so that the existence of a minimum to this problem is guaranteed without assumptions on the samples. Then, the Kullback-Leibler (KL) divergence between two NC-MSG is derived. This divergence enables us to define a second minimization problem. The latter is the computation of centers of mass of several NC-MSGs. Numerical experiments show the good performance and the speed of the Riemannian gradient descent on the two problems. Finally, a *Nearest centroid classifier* is implemented leveraging the KL divergence and its associated center of mass. Applied on the large-scale dataset *Breizhcroops*, this classifier shows good accuracies and robustness to rigid transformations of the test set.

Index Terms—Non-centered mixture of scaled Gaussian distributions, Robust location and scatter estimation, Riemannian optimization, Fisher Information Metric, Classification, Kullback-Leibler divergence, Center of mass.

I. INTRODUCTION

THE first and second-order statistical moments of the sample set $\{\mathbf{x}_i\}_{i=1}^n \in (\mathbb{R}^p)^n$ are ubiquitous features in signal processing and machine learning algorithms. Classically, these parameters are estimated using the empirical mean and the sample covariance matrix (SCM), which correspond to the maximum likelihood estimators (MLE) of the multivariate Gaussian

model. However, these estimates tend to perform poorly in the context of heavy-tailed distributions, or when the sample set contains outliers. In such setups, one can obtain a better fit to empirical distributions by considering more general statistical models, such as the elliptical distributions [1]. Within this broad family of distributions, M -estimators of the location and scatter [2] appear as generalized MLEs and have been leveraged for their robustness properties in many fields (cf. [3] for an extensive review).

An important subfamily of elliptical distributions is the compound Gaussian distributions, which models samples as $\mathbf{x} \stackrel{d}{=} \boldsymbol{\mu} + \sqrt{\tau} \mathbf{u}_i$, where $\boldsymbol{\mu} \in \mathbb{R}^p$ is the center (also referred to as location) of the distribution, $\mathbf{u}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ 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 probability density function for τ results in various models for \mathbf{x} . Compound Gaussian distributions encompass the t -distribution (including 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., $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \tau_i \boldsymbol{\Sigma})$. Such model will be referred to as *mixture of scaled Gaussian distributions* (MSG) [4]. The MLE of the scatter matrix $\boldsymbol{\Sigma}$ of this model coincides with Tyler's M -estimator of the scatter [5], which attracted considerable activity due to its robustness and distribution-free properties over the elliptical distributions family [6], [7], [8], [9]. However, its transposition to the non-centered case from the model $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \tau_i \boldsymbol{\Sigma})$ received less interest. This might be because the usual fixed-point iterations to evaluate its maximum likelihood may diverge in practice, which motivated the present work.

In this article, we tackle optimization problems related to parameter estimation and classification for a non-centered mixture of scaled Gaussian distributions (NC-MSG). The contribution is threefold:

First, we derive a new Riemannian gradient descent algorithm based on the Fisher-Rao information geometry of the considered statistical model. Indeed the parameter space (location, scatter, textures) is a product manifold that can be endowed with a Riemannian metric to leverage the Riemannian optimization framework [10], [11]. The Fisher-Rao information geometry corresponds to the one induced by the Fisher information metric. It is of particular interest since it is inherently well suited to the natural geometry of the data [12]. In this scope, we derive the

Manuscript received 22 September 2022; revised 9 February 2023 and 28 April 2023; accepted 25 June 2023. Date of publication 28 June 2023; date of current version 11 July 2023. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ya-Feng Liu. This work was supported by ANR MASSILIA under Grant ANR-21-CE23-0038-01. (Corresponding author: Arnaud Breloy.)

Antoine Collas and Chengfang Ren are with the SONDRRA, Centrale-Supélec, Université Paris-Saclay, 91192 Gif-sur-Yvette, France (e-mail: antoine.collas@centralesupelec.fr; chengfang.ren@centralesupelec.fr).

Arnaud Breloy is with the LEME (EA4416), University Paris Nanterre, 92410 Ville-d'Avray, France (e-mail: abreloy@parisnanterre.fr).

Guillaume Ginolhac is with the LISTIC (EA3703), University Savoie Mont Blanc, 74940 Annecy, France (e-mail: guillaume.ginolhac@univ-smb.fr).

Jean-Philippe Ovarlez is with the SONDRRA, CentraleSupélec, Université Paris-Saclay, 91192 Gif-sur-Yvette, France, and also with the DEMR, ON-ERA, Université Paris-Saclay, 91192 Gif-sur-Yvette, France (e-mail: jean-philippe.ovarlez@mac.com).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TSP.2023.3290354>, provided by the authors.

Digital Object Identifier 10.1109/TSP.2023.3290354

Riemannian gradient (also referred to as the natural gradient) and a second-order retraction of this geometry. These tools are enough to cast a gradient descent applicable to any function of the parameters. We focus on two prominent examples that are regularized maximum likelihood estimation and center of mass computation. Simulations evidence that the proposed approach allows for fast computation of critical points, as it can converge with up to one order of magnitude less of iterations compared to other 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 [2], [3], [5], for example when there is insufficient sample support ($n < p$). In such setups, it is 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 scatter matrix [13], [14], [15], and regularizing both the mean and the scatter for the non-centered t -distribution was studied in [16]. Other regularizations formulated on the spectrum of the scatter matrix were proposed in [4], [17], [18] 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 scatter matrix to a pre-defined $\kappa \in \mathbb{R}_*^+$). We derive the general conditions for these penalties to ensure the existence of a solution for 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 study the invariance properties of the resulting estimators.

Finally, we apply the proposed algorithm to perform Riemannian classification. We consider the framework where statistical features of sample batches are used to discriminate between classes [19], [20], [21], [22]. The Riemannian approach then consists in generalizing usual classification algorithms (e.g., the *Nearest centroid classifier*) by replacing the Euclidean distance and arithmetic mean by divergence and its corresponding center of mass [23], [24], [25]. In this setup, the information geometry can help design meaningful distances between the features, and improve the output performance [19], [22]. 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 [26], [27], [28]). 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 *Breizhcroops* dataset [29]. Our experiments evidence that regularizing the estimation greatly improves the accuracy. Thanks to the invariance

properties of the proposed estimators, we also show that this process exhibits good robustness to rigid transformations of the samples during the inference.

The rest of the article is organized as follows. Section II presents NC-MSGs and casts their parameter space as a manifold. Section III presents elements of Riemannian geometry, and studies the Fisher-Rao information geometry for this model. Section IV derives a Riemannian gradient descent algorithm following this geometry. Section V discusses parameter estimation in the considered model, presents a new class of regularized estimators, and studies some of their properties (existence, invariances). Section VI derives the KL divergence of the model and its associated center of mass. Section VII concludes with validation simulations and an application to Riemannian classification of the *Breizhcroops* dataset. For conciseness, some technical proofs are in appendices that are provided as Supplementary materials.

II. NON-CENTERED MIXTURE OF SCALED GAUSSIAN DISTRIBUTIONS AND ITS PARAMETER SPACE $\mathcal{M}_{p,n}$

A. Data Model

Let a set of n data points $\{\mathbf{x}_i\}_{i=1}^n$ belonging to \mathbb{R}^p and distributed according to the following statistical model

$$\mathbf{x}_i \stackrel{d}{=} \boldsymbol{\mu} + \sqrt{\tau_i} \boldsymbol{\Sigma}^{\frac{1}{2}} \mathbf{u}_i, \quad (1)$$

where \mathbf{u}_i follows a white circular Gaussian distribution i.e. $\mathbf{u}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p)$. The variables $\boldsymbol{\mu} \in \mathbb{R}^p$ and $\boldsymbol{\Sigma} \in \mathcal{S}_p^{++}$ (set of $p \times p$ symmetric positive definite matrices) are respectively named the location and scatter parameters. Then, the unknown texture parameters $\{\tau_i\}_{i=1}^n$ are stacked into the vector $\boldsymbol{\tau} \in (\mathbb{R}_*^+)^n$ (set of strictly positive vectors). If these textures admit a probability density function (p.d.f.), then the random variables (r.v.) \mathbf{x}_i follow a Compound Gaussian distribution [3], [30]. However, in general, this p.d.f. is unknown. In order not to rely on additional pdf assumptions on the textures, these are often assumed to be unknown and deterministic [7], [31]. In this case, the r.v. \mathbf{x}_i follow a NC-MSG, i.e.

$$\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \tau_i \boldsymbol{\Sigma}). \quad (2)$$

Thus \mathbf{x}_i admits a p.d.f. f defined from the Gaussian one f_G

$$f(\mathbf{x}_i | (\boldsymbol{\mu}, \boldsymbol{\Sigma}, \tau_i)) = f_G(\mathbf{x}_i | (\boldsymbol{\mu}, \tau_i \boldsymbol{\Sigma})), \quad (3)$$

with $\forall \mathbf{x} \in \mathbb{R}^p$

$$f_G(\mathbf{x} | (\boldsymbol{\mu}, \boldsymbol{\Sigma})) = (2\pi)^{-\frac{p}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp \left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right]. \quad (4)$$

The negative log-likelihood (NLL) of the sample set $\{\mathbf{x}_i\}_{i=1}^n$ 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)

$$\mathcal{L}(\theta | \{\mathbf{x}_i\}_{i=1}^n) = \frac{1}{2} \sum_{i=1}^n \left[\log |\tau_i \boldsymbol{\Sigma}| + \frac{(\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu})}{\tau_i} \right]. \quad (5)$$

One can observe an ambiguity between the textures τ and the scatter matrix Σ . Indeed, $\forall \alpha > 0$, we have

$$\mathcal{L}((\boldsymbol{\mu}, \alpha \Sigma, \alpha^{-1} \boldsymbol{\tau}) | \{\mathbf{x}_i\}_{i=1}^n) = \mathcal{L}((\boldsymbol{\mu}, \Sigma, \boldsymbol{\tau}) | \{\mathbf{x}_i\}_{i=1}^n). \quad (6)$$

Thus, to identify the textures and scatter matrix parameters, a constraint on $\boldsymbol{\tau}$ or Σ 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 of the article could be obtained by constraining the scatter matrix instead of the textures, with a unit determinant constraint, i.e. $|\Sigma| = 1$ [32], [33]. Then, the parameter space of interest is

$$\mathcal{M}_{p,n} = \mathbb{R}^p \times \mathcal{S}_p^{++} \times \mathcal{S}(\mathbb{R}_*^+)^n, \quad (7)$$

where $\mathcal{S}(\mathbb{R}_*^+)^n$ is the set of textures with the unit product,

$$\mathcal{S}(\mathbb{R}_*^+)^n = \left\{ \boldsymbol{\tau} \in (\mathbb{R}_*^+)^n : \prod_{i=1}^n \tau_i = 1 \right\}. \quad (8)$$

The choice of adding a constraint is motivated by two results additional to the identifiability: (i) it reduces the dimension of the parameter space by removing the indeterminacy (6), (ii) the associated FIM (see Proposition 1) admits a simpler expression, which will be instrumental in the rest of the article 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 Σ).

B. Related Works

When $\{\mathbf{x}_i\}_{i=1}^n$ is sampled from an underlying heavy-tailed Compound Gaussian distribution, the empirical mean and SCM do not provide robust and accurate estimates of $\boldsymbol{\mu}$ and Σ . In this setup, M -estimators [2], raised increasing interest in the past decades (see e.g. [3]). These estimators are expressed through the two joint fixed-point equations

$$\begin{aligned} \boldsymbol{\mu} &= \left(\sum_{i=1}^n u_1(t_i) \right)^{-1} \sum_{i=1}^n u_1(t_i) \mathbf{x} \triangleq \mathcal{H}_{\boldsymbol{\mu}}(\boldsymbol{\mu}, \Sigma), \\ \Sigma &= \frac{1}{n} \sum_{i=1}^n u_2(t_i) (\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T \triangleq \mathcal{H}_{\Sigma}(\boldsymbol{\mu}, \Sigma), \end{aligned} \quad (9)$$

where $t_i \triangleq (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})$, u_1 and u_2 are functions that respect Maronna's conditions¹ [2]. Under certain conditions [2], these estimators can be computed with fixed-point iterations

$$\begin{aligned} \boldsymbol{\mu}_{k+1} &= \mathcal{H}_{\boldsymbol{\mu}}(\boldsymbol{\mu}_k, \Sigma_k), \\ \Sigma_{k+1} &= \mathcal{H}_{\Sigma}(\boldsymbol{\mu}_{k+1}, \Sigma_k), \end{aligned} \quad (10)$$

that converge towards a unique solution satisfying (9). Interestingly, some M -estimators also appear as MLE when $u_1(t) = u_2(t)$ is linked to the p.d.f. of an elliptical distribution [3].

¹Notice that [2] rather uses a formulation of (9) involving “ $u_1(t_i)$ ” and “ $u_2(t_i^2)$ ”, with $t_i = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})}$. Without loss of generality, this article uses the present notation to simplify some discussions.

Expressing these estimators as the solution of an optimization problem drove a more recent line of work leveraging optimization theory allowing, e.g., for generalizations to structured scatter matrix matrices [34], [35], [36] or regularized location and scatter matrix estimation [16].

In the context of scatter matrix estimation, Tyler's M -estimator [5] is especially interesting thanks to its robustness and “distribution-free” properties over the elliptical distributions family. Tyler's M -estimator is obtained for $\boldsymbol{\mu} = \mathbf{0}$ and $u_2(t) = p/t$, and also coincide with the MLE of the centered MSG [6], [7]. However, this estimator cannot trivially be transposed to the case of joint mean-scatter matrix estimation. Indeed, the MLE solution associated with NC-MSG is obtained with $u_1(t) = u_2(t) = p/t$, which does not satisfy Maronna's conditions [2], and for which the fixed-point iterations (10) generally diverge. Thus, Tyler's M -estimator of the scatter matrix is usually applied on demeaned data, where the mean is estimated in a prior step². It was yet experienced that the MLE of NC-MSG could be evaluated in practice with Riemannian optimization rather than potentially unstable fixed-point iterations in [37] (still, without any theoretical guarantees). The following of this article builds upon this finding in several directions: optimization in Sections III-B and IV, regularized estimation with theoretical guarantees in Section V, and classification in Sections VI and VII.

III. RIEMANNIAN GEOMETRY OF $\mathcal{M}_{p,n}$

The objective of this section is to present the information geometry of the NC-MSG (2); i.e. the Riemannian geometry of $\mathcal{M}_{p,n}$ with the FIM as a Riemannian metric. This Riemannian geometry is leveraged to optimize several cost functions $h : \mathcal{M}_{p,n} \rightarrow \mathbb{R}$. Notably, two cost functions will be studied: a regularized NLL in Section V, and a cost function to compute centers of mass of sets of points $\{\theta_i\} \subset \mathcal{M}_{p,n}$ in Section VI. Before turning $\mathcal{M}_{p,n}$ into a Riemannian manifold, a brief introduction to Riemannian geometry is made. For a complete introduction to the topic, see [10], [11].

A. Riemannian Geometry

Let \mathcal{E} be a linear space of dimension d . Informally, a smooth embedded manifold $\mathcal{M} \subset \mathcal{E}$ of dimension $l \leq d$ is a nonempty set that locally resembles a l -dimensional linear space. Indeed, \mathcal{M} is a smooth embedded manifold of \mathcal{E} if and only if it is locally diffeomorphic³ with open sets of a l -dimensional linear subspace in \mathbb{R}^d . Then, smooth curves c are smooth functions from open intervals I of \mathbb{R} to \mathcal{M} ; i.e. $c : I \rightarrow \mathcal{M}$. Collecting velocities of the curves passing through $x \in \mathcal{M}$, we get the tangent space at x :

$$T_x \mathcal{M} = \{c'(0) \mid c : I \rightarrow \mathcal{M} \text{ is smooth and } c(0) = x\}. \quad (11)$$

²We point out that a closely related estimator proposed in [5] uses $u_1(t) = \sqrt{p/t}$ and $u_2(t) = p/t$, which yields converging fixed-point iterations in practice despite being a limit case of Maronna's conditions. This estimate, however, is not obtained as the solution of an underlying optimization problem, i.e., has no MLE interpretation.

³A diffeomorphism is a bijective map $f : U \rightarrow V$ where U, V are open sets and such that both f and f^{-1} are smooth (or infinitely differentiable).

This tangent space corresponds to a linearization of \mathcal{M} at x . The tangent bundle of \mathcal{M} is then the disjoint union of all the tangent spaces of \mathcal{M} , i.e., $T\mathcal{M} = \{(x, \xi) : x \in \mathcal{M} \text{ and } \xi \in T_x\mathcal{M}\}$.

So far, we have defined the notion of the smooth embedded manifold of a linear space. To turn \mathcal{M} into a *Riemannian manifold*, its tangent spaces $T_x\mathcal{M}$ are equipped with a *Riemannian metric* which is an *inner product*⁴ $\langle \cdot, \cdot \rangle_x^{\mathcal{M}} : T_x\mathcal{M} \times T_x\mathcal{M} \rightarrow \mathbb{R}$ that varies smoothly with respect to x .⁵

Then, to move on \mathcal{M} , a *geodesic* is a smooth curve on \mathcal{M} with zero acceleration along its path. In a *Euclidean space* \mathcal{E} the acceleration is classically defined as the second derivative. Thus, a geodesic $c : I \rightarrow \mathcal{E}$ is such that $\ddot{\gamma}(t) = 0 \forall t \in I$. If $\gamma(0) = x$ and $\dot{\gamma}(0) = \xi$, then, by integrating, we recover the classical straight line $\gamma(t) = x + t\xi$. This notion of acceleration is generalized to manifolds using the *Levi-Civita connection* denoted by ∇ . This notion requires first defining smooth vector fields, which are smooth mappings that associate a vector in $T\mathcal{M}$ for each point of the manifold \mathcal{M} , i.e.:

$$\begin{aligned} \xi : \mathcal{M} &\rightarrow T\mathcal{M} \\ x &\mapsto \xi(x). \end{aligned} \quad (12)$$

Notice that given this definition, $\xi(x) \in T_x\mathcal{M} \forall x \in \mathcal{M}$, so we also use the symbol ξ (respectively η) to denote a tangent vector when there is no ambiguity. Now, the Levi-Civita connection itself is defined as an operator that generalizes the directional derivative of vectors fields to Riemannian manifolds, and associates to every couple of smooth vector fields (ξ, η) on \mathcal{M} a new vector field $\nabla_\xi\eta$ on \mathcal{M} . Given a Riemannian manifold \mathcal{M} , the Levi-Civita connection is unique and defined by the *Koszul* formula. It should be noted that the Levi-Civita connection depends on the chosen Riemannian metric. Using this object, a geodesic $\gamma : I \rightarrow \mathcal{M}$ with initial conditions $\gamma(0) = x$ and $\dot{\gamma}(0) = \xi$ is defined as a smooth curve having zero acceleration as defined by the Levi-Civita connection

$$\nabla_{\dot{\gamma}(t)}\dot{\gamma}(t) = 0_{\gamma(t)}, \quad \forall t \in I \quad (13)$$

where $\dot{\gamma}(t) = \frac{d}{dt}\gamma(t)$ and $0_{\gamma(t)}$ is the zero element of $T_{\gamma(t)}\mathcal{M}$. Let γ be a geodesic defined on $[0, 1]$ with initial conditions $\gamma(0) = x$ and $\dot{\gamma}(0) = \xi$. Then, the *Riemannian exponential mapping* $\exp_x^{\mathcal{M}} : T_x\mathcal{M} \rightarrow \mathcal{M}$ at $x \in \mathcal{M}$ is defined as $\exp_x^{\mathcal{M}}(\xi) = \gamma(1)$. For $x, y \in \mathcal{M}$, its inverse function, the *Riemannian logarithm mapping*, is defined as $\log_x^{\mathcal{M}}(y) = \arg \min_{\xi \in T_x\mathcal{M}} \|\xi\|_x^2$ subject to $\exp_x^{\mathcal{M}}(\xi) = y$ with $\|\xi\|_x^2 = \langle \xi, \xi \rangle_x^{\mathcal{M}}$. Finally, the *Riemannian distance* between two points $x, y \in \mathcal{M}$ is computed as $d_{\mathcal{M}}(x, y) = \|\log_x^{\mathcal{M}}(y)\|_x$.

B. Description of the Riemannian Manifold $\mathcal{M}_{p,n}$

This subsection gives the Riemannian structure, induced by the FIM, of the parameter set $\mathcal{M}_{p,n}$. To specify the latter, we begin by defining the ambient space

$$\mathcal{E}_{p,n} = \mathbb{R}^p \times \mathbb{R}^{p \times p} \times \mathbb{R}^n. \quad (14)$$

⁴An *inner product* is a bilinear, symmetric, positive definite function on a \mathbb{R} -vector space.

⁵For all smooth vector fields ξ, η on \mathcal{M} the function $x \mapsto \langle \xi, \eta \rangle_x^{\mathcal{M}}$ is smooth.

Therefore, the tangent space of $\mathcal{M}_{p,n}$ at θ is a subspace of the ambient space $\mathcal{E}_{p,n}$

$$T_\theta\mathcal{M}_{p,n} = \left\{ \xi = (\xi_\mu, \xi_\Sigma, \xi_\tau) \in \mathbb{R}^p \times \mathcal{S}_p \times \mathbb{R}^n : \xi_\tau^T \tau^{\odot -1} = 0 \right\}, \quad (15)$$

where \mathcal{S}_p is the set of symmetric matrices and $\cdot^{\odot -1}$ is the elementwise inverse operator. To turn $\mathcal{M}_{p,n}$ into a Riemannian manifold, we must equip $\mathcal{M}_{p,n}$ with a Riemannian metric. Many possibilities are available to us, however, a preferable one is the FIM [38] derived in Proposition 1. Indeed, it is calculated using the NLL (5) and thus is associated with the statistical model (1).

Proposition 1 (Fisher Information Metric): Let $\theta \in \mathcal{M}_{p,n}$ and $\xi, \eta \in T_\theta\mathcal{M}_{p,n}$, the Fisher Information Metric at θ associated with the NLL (5) is

$$\begin{aligned} \langle \xi, \eta \rangle_\theta^{\mathcal{M}_{p,n}} &= \sum_{i=1}^n \left(\frac{1}{\tau_i} \right) \xi_\mu^T \Sigma^{-1} \eta_\mu + \frac{n}{2} \text{Tr} \left(\Sigma^{-1} \xi_\Sigma \Sigma^{-1} \eta_\Sigma \right) \\ &\quad + \frac{p}{2} (\xi_\tau \odot \tau^{\odot -1})^T (\eta_\tau \odot \tau^{\odot -1}), \end{aligned}$$

where \odot is the elementwise product operator.

Proof: See Supplementary material A. \square

Then, the orthogonal projection according to the FIM from $\mathcal{E}_{p,n}$ onto $T_\theta\mathcal{M}_{p,n}$ is given in Proposition 2.

Proposition 2 (Orthogonal projection): The orthogonal projection associated with the FIM of Proposition 1 from $\mathcal{E}_{p,n}$ onto $T_\theta\mathcal{M}_{p,n}$ is

$$P_\theta^{\mathcal{M}_{p,n}}(\xi) = \left(\xi_\mu, \text{sym}(\xi_\Sigma), \xi_\tau - \frac{\xi_\tau^T \tau^{\odot -1}}{n} \tau \right),$$

where $\text{sym}(\xi) = \frac{1}{2}(\xi + \xi^T)$.

Proof: See Supplementary material B. \square

The orthogonal projection proves helpful to derive elements in tangent spaces such as the Riemannian gradient or the Levi-Civita connection. The latter is given for the manifold $\mathcal{M}_{p,n}$ in Proposition 3.

Proposition 3 (Levi-Civita connection): Let $\theta \in \mathcal{M}_{p,n}$ and ξ, η be smooth vector fields of $\mathcal{M}_{p,n}$, the Levi-Civita connection of $\mathcal{M}_{p,n}$ evaluated at θ is

$$\nabla_\xi\eta = P_\theta^{\mathcal{M}_{p,n}}(\bar{\nabla}_\xi\eta),$$

where

$$\begin{aligned} \bar{\nabla}_\xi\eta &= D\eta[\xi] + \left(-\frac{1}{2} \left[\left(\frac{\xi_\tau^T \tau^{\odot -2}}{\sum_{i=1}^n \frac{1}{\tau_i}} \mathbf{I}_p + \xi_\Sigma \Sigma^{-1} \right) \eta_\mu \right. \right. \\ &\quad \left. \left. + \left(\frac{\eta_\tau^T \tau^{\odot -2}}{\sum_{i=1}^n \frac{1}{\tau_i}} \mathbf{I}_p + \eta_\Sigma \Sigma^{-1} \right) \xi_\mu \right], \right. \\ &\quad \left. \frac{1}{n} \sum_{i=1}^n \left(\frac{1}{\tau_i} \right) \eta_\mu \xi_\mu^T - \xi_\Sigma \Sigma^{-1} \eta_\Sigma, \right. \\ &\quad \left. \frac{1}{p} \xi_\mu^T \Sigma^{-1} \eta_\mu \mathbf{1}_n - \xi_\tau \odot \eta_\tau \odot \tau^{\odot -1} \right). \end{aligned}$$

Proof: See Supplementary material C. \square

Algorithm 1: Riemannian Gradient Descent on $\mathcal{M}_{p,n}$.**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 using Algorithm 2 Set $\theta^{(k+1)} \leftarrow R_{\theta^{(k)}}^{\mathcal{M}_{p,n}} \left(-\alpha \text{grad}_{\mathcal{M}_{p,n}} h(\theta^{(k)}) \right)$ **Algorithm 2:** Riemannian Backtracking on $\mathcal{M}_{p,n}$.**Input:** Current iterate $\theta^{(k)} \in \mathcal{M}_{p,n}$, and constants $\alpha \in]0, t_{\max}[$, $c \in]0, 1[$ and $\varepsilon \in \mathbb{R}_*^+$ **Output:** Step-size α Set $\theta(\alpha) \leftarrow R_{\theta^{(k)}}^{\mathcal{M}_{p,n}} \left(-\alpha \text{grad}_{\mathcal{M}_{p,n}} h(\theta^{(k)}) \right)$ **while** $h(\theta^{(k)}) - h(\theta(\alpha)) < \varepsilon \alpha \|\text{grad}_{\mathcal{M}_{p,n}} h(\theta^{(k)})\|_{\theta^{(k)}}^2$ **do** Set $\alpha \leftarrow c \alpha$ Set $\theta(\alpha) \leftarrow R_{\theta^{(k)}}^{\mathcal{M}_{p,n}} \left(-\alpha \text{grad}_{\mathcal{M}_{p,n}} h(\theta^{(k)}) \right)$

As detailed in Subsection III-A the Levi-Civita connection defines geodesics on a Riemannian manifold. Indeed, for I an open interval of \mathbb{R} , a geodesic $\gamma : I \rightarrow \mathcal{M}_{p,n}$ 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

$$\nabla_{\dot{\gamma}(t)} \dot{\gamma}(t) = 0_{\dot{\gamma}(t)}, \quad \forall t \in I. \quad (16)$$

However, an analytical solution of (16) remains unknown. A retraction (approximation of the geodesic) can still be obtained (see Proposition 5) which allows us to optimize functions on $\mathcal{M}_{p,n}$. Moreover, the geodesic between two points θ_1 and θ_2 is unknown. This implies that 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 [26], [27], [28], [39], [40]. To alleviate this problem, a divergence associated with the NC-MSG (2) is proposed in Section VI.

IV. RIEMANNIAN OPTIMIZATION ON $\mathcal{M}_{p,n}$

The objective of this subsection is to propose tools to perform optimization on the Riemannian manifold $\mathcal{M}_{p,n}$. Indeed, we aim to minimize smooth functions $h : \mathcal{M}_{p,n} \rightarrow \mathbb{R}$,

$$\underset{\theta \in \mathcal{M}_{p,n}}{\text{minimize}} \quad h(\theta). \quad (17)$$

An example of such a function is the NLL (5). As mentioned in Section III, two additional cost functions are studied in Sections V and VI. To realize (17), we consider a Riemannian steepest gradient descent on $\mathcal{M}_{p,n}$. Only the tools required for this algorithm are derived here. For a detailed introduction to optimization on Riemannian manifolds, see [10], [11]. Two optimization tools are needed: (i) the Riemannian gradient of h , (ii) a retraction that maps tangent vectors from $T_{\theta}\mathcal{M}_{p,n} \forall \theta \in \mathcal{M}_{p,n}$ onto $\mathcal{M}_{p,n}$. Once these are defined, the Riemannian steepest gradient descent retracts iteratively minus the gradient of h times a step size onto the manifold.

We begin with the Riemannian gradient of h at θ . For every $\theta \in \mathcal{M}_{p,n}$, it is defined through the Riemannian metric as the unique tangent vector in $T_{\theta}\mathcal{M}_{p,n}$ such that, $\forall \xi \in T_{\theta}\mathcal{M}_{p,n}$,

$$Dh(\theta)[\xi] = \langle \text{grad}_{\mathcal{M}_{p,n}} h(\theta), \xi \rangle_{\theta}^{\mathcal{M}_{p,n}}, \quad (18)$$

where $Dh(\theta)[\xi]$ is the directional derivative of h at θ in the direction ξ . In the case where for every $\theta \in \mathcal{M}_{p,n}$, there exists an open U of $\mathcal{E}_{p,n}$, with $\theta \in U$, and a differentiable function $\bar{h} : U \rightarrow \mathbb{R}$ such that \bar{h} restricted to $\mathcal{M}_{p,n}$ is equal to h , this Riemannian gradient can be computed from the Euclidean gradient of h at θ . In particular, this assumption is met by the different cost functions considered in the rest of the manuscript and the transformation of the Euclidean gradient into the Riemannian one is given in Proposition 4. The latter is very convenient since this Euclidean gradient can be computed using automatic differentiation libraries such as Autograd [41] or JAX [42].

Proposition 4 (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

$$\text{grad}_{\mathcal{M}_{p,n}} h(\theta) = P_{\theta}^{\mathcal{M}_{p,n}} \left(\left(\sum_{i=1}^n \frac{1}{\tau_i} \right)^{-1} \Sigma \mathbf{G}_{\mu}, \frac{2}{n} \Sigma \mathbf{G}_{\Sigma} \Sigma, \frac{2}{p} \tau^{\odot 2} \odot \mathbf{G}_{\tau} \right),$$

where $\text{grad} h(\theta) = (\mathbf{G}_{\mu}, \mathbf{G}_{\Sigma}, \mathbf{G}_{\tau})$ is the Euclidean gradient of h in $\mathbb{R}^p \times \mathbb{R}^{p \times p} \times \mathbb{R}^n$.

Proof: See Supplementary material D. \square

Then, it remains to define a retraction for every θ on $\mathcal{M}_{p,n}$. A retraction $R_{\theta}^{\mathcal{M}_{p,n}}$ maps every $\xi \in T_{\theta}\mathcal{M}_{p,n}$ to a point $R_{\theta}^{\mathcal{M}_{p,n}}(\xi) \in \mathcal{M}_{p,n}$ and is such that $R_{\theta}^{\mathcal{M}_{p,n}}(\xi) = \theta + \xi + o(\|\xi\|)$. 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 $\mathcal{M}_{p,n}$. Thus, we choose to enforce an additional property: the desired retraction must have a zero initial acceleration, i.e.

$$\nabla_{\dot{r}(t)} \dot{r}(t) \Big|_{t=0} = 0, \quad (19)$$

where $\dot{r}(t) = \frac{d}{dt} R_{\theta}^{\mathcal{M}_{p,n}}(t\xi)$ and ∇ is the Levi-Civita connection from Proposition 3. Such a retraction is called a second-order retraction. 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 (13)) whereas here this condition is only needed at $t = 0$. By respecting this property, the retraction is associated with the Riemannian metric of Proposition 1 since the Levi-Civita connection is derived from this Riemannian metric. Such a retraction is presented in Proposition 5.

Proposition 5 (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 Supplementary material) such that $\forall t \in [0, t_{\max}[$, a second order retraction on $\mathcal{M}_{p,n}$ at θ is

$$R_{\theta}^{\mathcal{M}_{p,n}}(t\xi) = \left(\mu + t\xi_{\mu} + \frac{t^2}{2} \left[\frac{\xi_{\tau}^T \tau^{\odot -2}}{\sum_{i=1}^n \frac{1}{\tau_i}} \mathbf{I}_p + \xi_{\Sigma} \Sigma^{-1} \right] \xi_{\mu}, \right.$$

$$\begin{aligned} & \Sigma + t\xi_\Sigma + \frac{t^2}{2} \left(\xi_\Sigma \Sigma^{-1} \xi_\Sigma - \frac{1}{n} \sum_{i=1}^n \left(\frac{1}{\tau_i} \right) \xi_{\mu} \xi_{\mu}^T \right), \\ & N \left(\tau + t\xi_\tau + \frac{t^2}{2} \left(\xi_\tau^{\odot 2} \odot \tau^{\odot -1} - \frac{1}{p} \xi_{\mu}^T \Sigma^{-1} \xi_{\mu} \mathbb{1}_n \right) \right), \end{aligned}$$

where $\forall \mathbf{x} = (x_i)_{1 \leq i \leq n} \in (\mathbb{R}_*^+)^n$, $N(\mathbf{x}) = (\prod_{i=1}^n x_i)^{-\frac{1}{n}} \mathbf{x}$.

Proof: See Supplementary material E. \square

With this retraction and the Riemannian gradient from Proposition 4, we have all the tools required to derive a Riemannian steepest descent. The latter is presented in Algorithm 1. It should be noted that, in practice, the step size is chosen using a backtracking algorithm [11, Ch. 4]. Given an initial step-size $\alpha \in]0, t_{\max}[$ with t_{\max} defined in Proposition 5, the algorithm reduces α by a factor $c \in]0, 1[$ until the Armijo–Goldstein condition is satisfied. Given $\varepsilon \in \mathbb{R}_*^+$ (generally fixed at 10^{-4}) and the tentative next iterate

$$\theta(\alpha) = R_{\theta^{(k)}}^{\mathcal{M}_{p,n}} \left(-\alpha \text{grad}_{\mathcal{M}_{p,n}} h(\theta^{(k)}) \right), \quad (20)$$

the Armijo–Goldstein condition writes

$$h(\theta^{(k)}) - h(\theta(\alpha)) \geq \varepsilon \alpha \left\| \text{grad}_{\mathcal{M}_{p,n}} h(\theta^{(k)}) \right\|_{\theta^{(k)}}^2. \quad (21)$$

This procedure is presented in Algorithm 2.

V. PARAMETER ESTIMATION OF THE NON-CENTERED MIXTURE OF SCALED GAUSSIAN DISTRIBUTIONS

In the previous subsection, tools to perform optimization on $\mathcal{M}_{p,n}$ have been developed. In this subsection, the objective is to leverage these tools to estimate the parameters of an NC-MSG (2). In the following, we assume having $n \geq 1$ data points $\{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^p$. The estimation of the parameters of the statistical model (2) is performed by maximizing the associated likelihood on $\mathcal{M}_{p,n}$:

$$\underset{\theta \in \mathcal{M}_{p,n}}{\text{minimize}} \mathcal{L}(\theta | \{\mathbf{x}_i\}_{i=1}^n), \quad (22)$$

where \mathcal{L} is the NLL (5). However, the existence of a solution to this problem is not guaranteed. To build intuition, we present a short example of a problematical case where $\boldsymbol{\mu}$ gets attracted by one data point \mathbf{x}_j . Let k be the current iteration of a given optimizer of (22). For $k \rightarrow +\infty$, if $\boldsymbol{\mu}^{(k)} \rightarrow \mathbf{x}_j$ faster than $\tau_j^{(k)} \rightarrow 0$ and $\forall i \neq j$, $\tau_i^{(k)} \rightarrow +\infty$, then the quadratic form in \mathcal{L} (5) tends to zero, which is its minimum,

$$\sum_{i=1}^n \frac{(\mathbf{x}_i - \boldsymbol{\mu}^{(k)})^T \left(\boldsymbol{\Sigma}^{(k)} \right)^{-1} (\mathbf{x}_i - \boldsymbol{\mu}^{(k)})}{\tau_i^{(k)}} \xrightarrow{k \rightarrow +\infty} 0. \quad (23)$$

Then, if an eigenvalue $\lambda^{(k)}$ of $\boldsymbol{\Sigma}^{(k)}$ tends 0 slower than the respective limits of $\boldsymbol{\mu}^{(k)}$, $\tau_i^{(k)}$ and $\tau_j^{(k)}$ and since $\sum_{i=1}^n \log |\tau_i \boldsymbol{\Sigma}| = n \log |\boldsymbol{\Sigma}|$, we obtain

$$\mathcal{L}(\theta^{(k)} | \{\mathbf{x}_i\}_{i=1}^n) \xrightarrow{k \rightarrow +\infty} -\infty. \quad (24)$$

Hence, depending on the data points $\{\mathbf{x}_i\}_{i=1}^n$, a solution of the problem (22) does not necessarily exist.

To overcome this issue, we propose a regularization approach to the NLL. Firstly, we prove that this allows the existence of a solution depending on some assumptions on the regularization term in V-A. Some interpretations on the chosen regularization are next given in V-B, and finally, we study the robustness of the solution to rigid transformations in V-C.

A. Existence of Solution With a Regularized Version of the NLL

We present a regularized version of the NLL (5):

$$\mathcal{L}_{\mathcal{R}_\kappa}(\theta | \{\mathbf{x}_i\}_{i=1}^n) = \mathcal{L}(\theta | \{\mathbf{x}_i\}_{i=1}^n) + \beta \mathcal{R}_\kappa(\theta), \quad (25)$$

where $\beta \in \mathbb{R}_*^+$ and $\mathcal{R}_\kappa : \mathcal{M}_{p,n} \rightarrow \mathbb{R}$ is a regularization. Thus, the minimization problem (22) becomes

$$\underset{\theta \in \mathcal{M}_{p,n}}{\text{minimize}} \mathcal{L}_{\mathcal{R}_\kappa}(\theta | \{\mathbf{x}_i\}_{i=1}^n). \quad (26)$$

Though (26) 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}_κ as a sum of regularizations r_κ on the eigenvalues of $\tau_i \boldsymbol{\Sigma}$. This rewriting is formalized in Assumption 1.

Assumption 1: The regularization \mathcal{R}_κ is a positive function that 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 $r_\kappa : \mathbb{R}_*^+ \rightarrow \mathbb{R}$ is a continuous function.

In the following, we assume that \mathcal{R}_κ respects Assumption 1. To prevent the eigenvalues of $\tau_i \boldsymbol{\Sigma}$ from taking values that are too large or too small, a second Assumption is added. Indeed, Assumption 2 states that the regularization of the log function by the penalty function 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 $\mathcal{L}_{\mathcal{R}_\kappa} \rightarrow +\infty$.

Assumption 2: The following function admits the limit $\forall \beta \in \mathbb{R}_*^+$

$$\lim_{x \rightarrow \partial \mathbb{R}_*^+} \log(x) + \beta r_\kappa(x) = +\infty, \quad (27)$$

with $\partial \mathbb{R}_*^+$ is a border of \mathbb{R}_*^+ , i.e. 0^+ or $+\infty$.

Assumptions 1 and 2 are sufficient to solve the problem of existence stated earlier. Indeed, when \mathcal{R}_κ respects these assumptions, Proposition 6 states that the problem (26) has a solution, i.e. $\mathcal{L}_{\mathcal{R}_\kappa}$ admits a minimum in $\mathcal{M}_{p,n}$. Finally, Assumptions 1 and 2 are quite easy to meet in practice. Indeed, several regularizations respecting these assumptions are proposed in Table I.

Proposition 6 (Existence): Under Assumptions 1 and 2, and $\forall \beta \in \mathbb{R}_*^+$, the regularized NLL

$$\theta \mapsto \mathcal{L}_{\mathcal{R}_\kappa}(\theta | \{\mathbf{x}_i\}_{i=1}^n) = \mathcal{L}(\theta | \{\mathbf{x}_i\}_{i=1}^n) + \beta \mathcal{R}_\kappa(\theta),$$

with \mathcal{L} being the NLL (5), admits a minimum in $\mathcal{M}_{p,n}$.

Proof: $\mathcal{L}_{\mathcal{R}_\kappa}$ is a continuous function on $\mathcal{M}_{p,n}$. Hence, to prove the existence of a solution to the minimization problem (26), it is enough to show that for sequences $\theta^{(k)} \rightarrow \partial \theta$,

TABLE I
EXAMPLES OF REGULARIZATIONS \mathcal{R}_κ RESPECTING THE ASSUMPTIONS 1, 2 AND 3

Name	$\mathcal{R}_\kappa(\theta)$	$r_\kappa(x)$
L1 penalty	$\left\ (\text{diag}(\boldsymbol{\tau}) \otimes \boldsymbol{\Sigma})^{-1} - \kappa^{-1} \mathbf{I}_{np} \right\ _1 = \sum_{i,j} (\tau_i \lambda_j)^{-1} - \kappa^{-1} $	$ x^{-1} - \kappa^{-1} $
L2 penalty	$\left\ (\text{diag}(\boldsymbol{\tau}) \otimes \boldsymbol{\Sigma})^{-1} - \kappa^{-1} \mathbf{I}_{np} \right\ _2^2 = \sum_{i,j} \left((\tau_i \lambda_j)^{-1} - \kappa^{-1} \right)^2$	$(x^{-1} - \kappa^{-1})^2$
Bures-Wasserstein squared distance	$d_{\text{BW}}^2((\text{diag}(\boldsymbol{\tau}) \otimes \boldsymbol{\Sigma})^{-1}, \kappa^{-1} \mathbf{I}_{np}) = \sum_{i,j} \left((\tau_i \lambda_j)^{-\frac{1}{2}} - \kappa^{-\frac{1}{2}} \right)^2$	$\left(x^{-\frac{1}{2}} - \kappa^{-\frac{1}{2}} \right)^2$
Gaussian KL divergence	$\delta_{\text{KL}}(\kappa \mathbf{I}_{np}, \text{diag}(\boldsymbol{\tau}) \otimes \boldsymbol{\Sigma}) = \frac{1}{2} \left[\sum_{i,j} (\kappa (\tau_i \lambda_j)^{-1} + \log(\tau_i \lambda_j)) - np(1 + \log(\kappa)) \right]$	$\frac{1}{2} [\kappa x^{-1} + \log(x) - (1 + \log(\kappa))]$

$\forall Q \in \mathbb{N}^*$, $\|\cdot\|_q$ is the Schatten norm, i.e. $\forall \mathbf{A} \in \mathcal{S}_p \|\mathbf{A}\|_q^q = \sum_i |\lambda_i|^q$ where λ_i are the eigenvalues of \mathbf{A} . The diagonal matrix with elements of $\boldsymbol{\tau}$ is denoted $\text{diag}(\boldsymbol{\tau})$. The Kronecker product between matrices is denoted \otimes .

the boundary of $\mathcal{M}_{p,n}$, we have that

$$\lim_{k \rightarrow +\infty} \mathcal{L}_{\mathcal{R}_\kappa}(\theta^{(k)} | \{\mathbf{x}_i\}_{i=1}^n) = +\infty. \quad (28)$$

First, we handle the cases where $\|\boldsymbol{\mu}^{(k)}\|_2 \not\rightarrow +\infty$. Since $\theta^{(k)} \rightarrow \partial\theta$, this means that, at least, one $\lambda_j^{(k)} \rightarrow \partial\mathbb{R}_*^+$ and/or one $\tau_i^{(k)} \rightarrow \partial\mathbb{R}_*^+$, with $\partial\mathbb{R}_*^+$ being the boundary of \mathbb{R}_*^+ , i.e. $\partial\mathbb{R}_*^+ = \{0, +\infty\}$. Using the positivity of the quadratic form in the NLL (5), we get the following inequality

$$\mathcal{L}(\theta^{(k)} | \{\mathbf{x}_i\}_{i=1}^n) \geq \sum_{i=1}^n \log \left| \tau_i^{(k)} \boldsymbol{\Sigma}^{(k)} \right|. \quad (29)$$

Hence, we get the resulting inequality on the regularized cost function

$$\begin{aligned} \mathcal{L}_{\mathcal{R}_\kappa}(\theta^{(k)} | \{\mathbf{x}_i\}_{i=1}^n) \geq \\ \sum_{i=1}^n \sum_{j=1}^p \left[\log(\tau_i^{(k)} \lambda_j^{(k)}) + \beta r_\kappa(\tau_i^{(k)} \lambda_j^{(k)}) \right]. \quad (30) \end{aligned}$$

Then, we give a sufficient condition to prove (28) when $\boldsymbol{\Sigma}^{(k)} \rightarrow \partial\mathcal{S}_p^{++}$ and/or $\boldsymbol{\tau}^{(k)} \rightarrow \partial\mathcal{S}(\mathbb{R}_*^+)^n$, the boundaries of $\partial\mathcal{S}_p^{++}$ and $\partial\mathcal{S}(\mathbb{R}_*^+)^n$ respectively. To give this sufficient condition, we first recall Assumption 1, $\forall \beta \in \mathbb{R}_*^+$

$$\lim_{x \rightarrow \partial\mathbb{R}_*^+} \log(x) + \beta r(x) = +\infty.$$

Thus, to prove (28), a sufficient condition, when $\boldsymbol{\Sigma}^{(k)} \rightarrow \partial\mathcal{S}_p^{++}$ and/or $\boldsymbol{\tau}^{(k)} \rightarrow \partial\mathcal{S}(\mathbb{R}_*^+)^n$ is that there exists at least one term $\tau_i^{(k)} \lambda_j^{(k)}$ such that

$$\tau_i^{(k)} \lambda_j^{(k)} \rightarrow \partial\mathbb{R}_*^+. \quad (31)$$

Since $\boldsymbol{\Sigma}^{(k)} \rightarrow \partial\mathcal{S}_p^{++}$ and/or $\boldsymbol{\tau}^{(k)} \rightarrow \partial\mathcal{S}(\mathbb{R}_*^+)^n$, there exists at least one $\lambda_j^{(k)} \rightarrow \partial\mathbb{R}_*^+$ and/or one $\tau_i^{(k)} \rightarrow \partial\mathbb{R}_*^+$.

The condition (31) is of course met in the four following cases

$$\lambda_j^{(k)} \rightarrow 0^+ \text{ and/or } \tau_i^{(k)} \rightarrow 0^+,$$

$$\lambda_j^{(k)} \rightarrow +\infty \text{ and/or } \tau_i^{(k)} \rightarrow +\infty,$$

$$\lambda_j^{(k)} \rightarrow 0^+ \text{ and } \tau_i^{(k)} \rightarrow +\infty \text{ such that } \tau_i^{(k)} \lambda_j^{(k)} \rightarrow \partial\mathbb{R}_*^+,$$

$$\lambda_j^{(k)} \rightarrow +\infty \text{ and } \tau_i^{(k)} \rightarrow 0^+ \text{ such that } \tau_i^{(k)} \lambda_j^{(k)} \rightarrow \partial\mathbb{R}_*^+.$$

Finally, we treat the case where $\forall l \in \{1, \dots, n\}$, $\lambda_l^{(k)} \rightarrow \partial\mathbb{R}_*^+$ and $\tau_i^{(k)} \rightarrow \partial\mathbb{R}_*^+$ such that $\tau_i^{(k)} \lambda_l^{(k)} \not\rightarrow \partial\mathbb{R}_*^+$. Since $\prod_{m=1}^n \tau_m^{(k)} = 1$, there exists at least one τ_q , with $q \neq i$, such that

$$\tau_q^{(k)} \lambda_j^{(k)} \rightarrow \partial\mathbb{R}_*^+. \quad (32)$$

Hence, the condition (28) is met.

Before going further, we define the two following functions:

$$g_{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma}^{(k)}) = \left\| \log(\boldsymbol{\Sigma}^{(k)}) \right\|_F^2 = \sum_{j=1}^p \log(\lambda_j^{(k)})^2,$$

and

$$g_{(\mathbb{R}_*^+)^n}(\boldsymbol{\tau}^{(k)}) = \left\| \log(\boldsymbol{\tau}^{(k)}) \right\|_2^2 = \sum_{i=1}^n \log(\tau_i^{(k)})^2.$$

It should be noted that $\sup(g_{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma}^{(k)})_{k=0}^{+\infty}) = +\infty$ if and only if there exists j such that $(\lambda_j^{(k)})_{k=0}^{+\infty}$ has $\partial\mathbb{R}_*^+$ as accumulation point. Similarly, $\sup(g_{(\mathbb{R}_*^+)^n}(\boldsymbol{\tau}^{(k)})_{k=0}^{+\infty}) = +\infty$ if and only if there exists i such that $(\tau_i^{(k)})_{k=0}^{+\infty}$ has $\partial\mathbb{R}_*^+$ as accumulation point.

Second, we consider the cases where $\|\boldsymbol{\mu}^{(k)}\|_2 \rightarrow +\infty$, $\sup(g_{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma}^{(k)})_{k=0}^{+\infty}) < +\infty$ and $\sup(g_{(\mathbb{R}_*^+)^n}(\boldsymbol{\tau}^{(k)})_{k=0}^{+\infty}) < +\infty$. In this case, there exists $\lambda_{\min}, \lambda_{\max} > 0$ and $\tau_{\min}, \tau_{\max} > 0$ such that for all k , $\lambda_{\min} \mathbf{I}_p \preceq \boldsymbol{\Sigma}^{(k)} \preceq \lambda_{\max} \mathbf{I}_p$ and $\tau_{\min} \mathbb{1}_n \leq \boldsymbol{\tau}^{(k)} \leq \tau_{\max} \mathbb{1}_n$. Indeed, otherwise there would exist j, i such that $\partial\mathbb{R}_*^+$ is an accumulation point of $(\lambda_j^{(k)})_{k=0}^{+\infty}$ and/or $(\tau_i^{(k)})_{k=0}^{+\infty}$. Using these inequalities and the positivity of the regularization \mathcal{R}_κ , we get that

$$\mathcal{L}_{\mathcal{R}_\kappa}(\theta^{(k)} | \{\mathbf{x}_i\}_{i=1}^n) \geq \sum_{i=1}^n \frac{\|\mathbf{x}_i - \boldsymbol{\mu}^{(k)}\|^2}{\lambda_{\max} \tau_{\max}} + \text{const},$$

where the constant is independent from $\boldsymbol{\mu}^{(k)}$. Thus, we have

$$\lim_{k \rightarrow +\infty} \mathcal{L}_{\mathcal{R}_\kappa}(\theta^{(k)} | \{\mathbf{x}_i\}_{i=1}^n) = +\infty.$$

Third, it remains to check the cases where $\|\boldsymbol{\mu}^{(k)}\| \rightarrow +\infty$, $\sup(g_{\mathcal{S}_p^{++}}(\boldsymbol{\Sigma}^{(k)})_{k=0}^{+\infty}) = +\infty$ and/or $\sup(g_{(\mathbb{R}_*^+)^n}(\boldsymbol{\tau}^{(k)})_{k=0}^{+\infty}) = +\infty$. Thus, as said previously, there exists at least one j and/or one i such that $(\lambda_j^{(k)})_{k=0}^{+\infty}$ and/or $(\tau_i^{(k)})_{k=0}^{+\infty}$ has/have $\partial\mathbb{R}_*^+$ as accumulation point. For each each of those j, i , we extract subsequences from $(\theta^{(k)})_{k=0}^{+\infty}$ whose limits in λ_j and/or τ_i are these

problematic accumulation points. Then, we construct a partition of \mathbb{N} with the indices corresponding to the elements of these sub-sequences and the indices of the remaining elements of the initial sequence $(\theta^{(k)})_{k=0}^{+\infty}$. Let \mathcal{I} be such partition of \mathbb{N} . If \mathcal{I} has a finite number of elements and if for every $(k_\ell)_{\ell=0}^{+\infty} \in \mathcal{I}$ we have

$$\lim_{\ell \rightarrow +\infty} \mathcal{L}_{\mathcal{R}_\kappa}(\theta^{(k_\ell)} | \{\mathbf{x}_i\}_{i=1}^n) = +\infty, \quad (33)$$

then we get (28).

To do so, in the space $\overline{\mathbb{R}^+} = [0, +\infty]$ equipped with the metric $d(x, y) = |\arctan(x) - \arctan(y)|$, given a $(\lambda_j^{(k)})_{k=0}^{+\infty}$ that has $L \in \partial\mathbb{R}_*^+$ as accumulation point, one can extract a sub-sequence of indices $(k_\ell)_{\ell=0}^{+\infty}$ such that $\lambda_j^{(k_\ell)} \rightarrow L$ and for $(k_m)_{m=0}^{+\infty} = \mathbb{N} \setminus (k_\ell)_{\ell=0}^{+\infty}$, $(\lambda_j^{(k_m)})_{m=0}^{+\infty}$ does not have L as accumulation point. The same process is repeated iteratively from the remaining indices $(k_m)_{m=0}^{+\infty}$ for all j such that $(\lambda_j^{(k_m)})_{m=0}^{+\infty}$ still has an $L \in \partial\mathbb{R}_*^+$ as accumulation point. It finishes when the sequence associated with the remaining elements of the original sequence $(\lambda_j^{(k)})_{k=0}^{+\infty}$ has no accumulation points in $\partial\mathbb{R}_*^+$. Let s denotes $(k_q)_{q=0}^{+\infty}$ the remaining indices. Then, the same process is also performed on $(\tau_i^{(k_q)})_{q=0}^{+\infty}$ if $\sup(g_{(\mathbb{R}_*^+)^n}(\boldsymbol{\tau}^{(k_q)}))_{q=0}^{+\infty} = +\infty$. All the obtained sequences of indices $(k_\ell)_{\ell=0}^{+\infty}$ along with the remaining elements of the original indices form a partition of \mathbb{N} . Due to its construction, this partition has at most $\text{card}(\partial\mathbb{R}_*^+)^{p+n} + 1 = 2^{p+n} + 1$ elements. Furthermore, we point out that, since $\|\boldsymbol{\mu}^{(k)}\| \rightarrow +\infty$, we have that for every sub-sequence $(\boldsymbol{\mu}^{(k_\ell)})_{\ell=0}^{+\infty}$, $\|\boldsymbol{\mu}^{(k_\ell)}\| \rightarrow +\infty$. Thus, for every $(k_\ell)_{\ell=0}^{+\infty} \in \mathcal{I}$, we have

- either $\|\boldsymbol{\mu}^{(k_\ell)}\| \rightarrow +\infty$, $\sup(g_{S_p^{++}}(\boldsymbol{\Sigma}^{(k_\ell)}))_{\ell=0}^{+\infty} < +\infty$ and $\sup(g_{(\mathbb{R}_*^+)^n}(\boldsymbol{\tau}^{(k_\ell)}))_{\ell=0}^{+\infty} < +\infty$,
- or $\|\boldsymbol{\mu}^{(k_\ell)}\| \rightarrow +\infty$ and there exists i and/or j such that $\lambda_j^{(k_\ell)} \rightarrow \partial\mathbb{R}_*^+$ and/or $\tau_i^{(k_\ell)} \rightarrow \partial\mathbb{R}_*^+$.

The former case has already been treated earlier. For the latter case, one can reuse the arguments between (29) and (32) to prove (33). Indeed, (29) discards the quadratic form in $\boldsymbol{\mu}^{(k_\ell)}$ and hence the equations between (29) and (32) hold. Thus, the condition (28) is met. \square

B. Interpretations of the Regularization Term

So far, the regularization penalty has been chosen to guarantee the existence of a solution to the problem (26) without having specific insights on its impact on the estimate. Therefore, this section thus discusses the interpretations of various classes of penalties and their related shrinkage effect.

A Bayesian interpretation of the considered penalties \mathcal{R}_κ requires first discussing the case where it is decoupled in terms of $\{\tau_i\}_{i=1}^n$ and $\{\lambda_j\}_{j=1}^p$, i.e., when it can be expressed as

$$\mathcal{R}_\kappa(\theta) = p \sum_{i=1}^n r_\kappa^\tau(\tau_i) + n \sum_{j=1}^p r_\kappa^\lambda(\lambda_j). \quad (34)$$

In such cases,

- r_κ^τ can be linked to a pdf on τ , denoted f_τ . Assuming that $r_\kappa^\lambda(t) = 0$ the optimization problem relates to the maximum a posteriori estimation of the Compound Gaussian

model $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with $\tau \sim f_\tau$ [2], [3]. Such a procedure is not often put into practice because it is generally possible (and preferable) to study the resulting pdf of the observations \mathbf{x} :

$$f_{CG}(\mathbf{x}) \propto \int f_G(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) f_\tau(\tau) d\tau, \quad (35)$$

whose MLE estimator appears as a special case of M -estimators of location and scatter, and is tractable with a fixed point algorithm [2], [3].

- The penalty r_κ^λ could also be interpreted as a pdf on the eigenvalues of $\boldsymbol{\Sigma}$. This approach is less often studied from the Bayesian point of view because it does not have a clear interpretation of the distribution of the resulting $\boldsymbol{\Sigma}$. Still, such penalties were leveraged to ensure existence of solutions of regularized M -estimators when $n < p$, e.g, in [17], [18], [43].

When additional prior information is available (power constraints that bound the eigenvalues, a rough estimate of the textures pdf, etc.) a Bayesian approach can be practically leveraged to select the form of the regularization penalty and the regularization parameters κ and β .

In the general case of Assumption 1, i.e., where \mathcal{R}_κ is possibly not decoupled, a Bayesian interpretation of \mathcal{R}_κ is not as apparent. Still, we can show that when the penalty can be interpreted as a divergence, it allows for explaining its effect on the estimate. First, we recall the definition of a divergence:

Definition 1 (Divergence): A divergence on a set E is a function $\delta(\cdot, \cdot) : E \times E \rightarrow \mathbb{R}$ satisfying, $\forall x, y \in E$:

- 1) $\delta(x, y) \geq 0$,
- 2) $\delta(x, y) = 0 \iff x = y$.

We can then state the following assumption, which is notably verified for all regularization examples given in Table 1:

Assumption 3: The regularization \mathcal{R}_κ can be written as

$$\mathcal{R}_\kappa(\theta) = \delta_{S_p^{++}}(\text{diag}(\boldsymbol{\tau}) \otimes \boldsymbol{\Sigma}, \kappa \mathbf{I}_{np}),$$

where $\delta_{S_p^{++}}$ is a divergence on the set S_p^{++} and $\kappa \in \mathbb{R}_*^+$.

This assumption allows us to state the following proposition:

Proposition 7 (Minima of \mathcal{R}_κ): Under the Assumption 3, the set of minima in $\mathcal{M}_{p,n}$ of the regularization \mathcal{R}_κ is

$$\{\theta = (\boldsymbol{\mu}, \kappa \mathbf{I}_p, \mathbb{1}_n) : \boldsymbol{\mu} \in \mathbb{R}^p\}.$$

Proof: The objective of this proof is to solve

$$\underset{\theta \in \mathcal{M}_{p,n}}{\text{minimize}} \mathcal{R}_\kappa(\theta).$$

Using Assumption 3, we know that $\mathcal{R}_\kappa(\theta) \geq 0$ and $\mathcal{R}_\kappa(\theta) = 0 \iff \text{diag}(\boldsymbol{\tau}) \otimes \boldsymbol{\Sigma} = \kappa \mathbf{I}_{np}$. Thus, the minimum of \mathcal{R}_κ is 0 and is reached at $\text{diag}(\boldsymbol{\tau}) \otimes \boldsymbol{\Sigma} = \kappa \mathbf{I}_{np}$, $\forall \boldsymbol{\mu} \in \mathbb{R}^p$. This implies that the minimum satisfies the following system of equations

$$\tau_i \lambda_j = \kappa \quad \forall i, j.$$

Hence, we deduce that $\tau_1 = \dots = \tau_n$. Using the constraint $\prod_{i=1}^n \tau_i = 1$, we get that $\tau_1 = \dots = \tau_n = 1$. Thus, $\lambda_1 = \dots = \lambda_p = \kappa$. This means that

$$\{(\boldsymbol{\mu}, \kappa \mathbf{I}_p, \mathbb{1}_n) : \boldsymbol{\mu} \in \mathbb{R}^p\} = \arg \min_{\theta \in \mathcal{M}_{p,n}} \mathcal{R}_\kappa(\theta),$$

which characterizes Proposition 7. \square

Thus, under Assumption 3, the minimum of (26) tends to $(\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i, \kappa \mathbf{I}_p, \mathbb{1}_n)$ as $\beta \rightarrow +\infty$. This corresponds to the MLE of a Gaussian distribution with a covariance matrix $\kappa \mathbf{I}$. Thus, the hyperparameter β makes the trade-off between an NC-MSG (2) and a white Gaussian distribution. Hence, one can set in practice the hyperparameter κ as $\kappa = \frac{1}{p} \text{Tr}(\frac{1}{n} \mathbf{X} \mathbf{X}^T) = \frac{1}{np} \sum_{i=1}^n \|\mathbf{x}_i\|_2^2$, meaning that the eigenvalues will be shrunk towards their empirical mean. The effect of the regularization then echoes to existing shrinkage of M -estimators that have the same action [13], [14], [15], [43], [44].

To conclude, Assumption 2 and Proposition 6 provide the conditions that ensure the existence of a solution of the regularized MLE for any $\kappa > 0$, whether \mathcal{R}_κ is decoupled (with a Bayesian interpretation), interpretable as a divergence (following Assumption 3 and Proposition 7), or not. This class of regularization penalties thus allows going beyond the Bayesian estimation framework. In practice, we mostly consider minimizing the estimation bias induced by the penalty and set β close to 0. For other tasks such as estimates used in classification, we resort to cross-validation procedures to select β (see example in Fig. 7).

C. Robustness to Rigid Transformations

We finish this section with a remark on estimating the parameter θ when data undergo a rigid transformation. First of all, we define the set of orthogonal matrices

$$\mathcal{O}_p = \{ \mathbf{Q} \in \mathbb{R}^{p \times p} : \mathbf{Q}^T \mathbf{Q} = \mathbf{I}_p \}. \quad (36)$$

Then, given $\mathbf{Q} \in \mathcal{O}_p$ and $\boldsymbol{\mu}_0 \in \mathbb{R}^p$, the rigid transformation ψ of a set of data $\{\mathbf{x}_i\}_{i=1}^n$ is defined as

$$\psi(\{\mathbf{x}_i\}_{i=1}^n) = \{ \mathbf{Q}^T \mathbf{x}_i + \boldsymbol{\mu}_0 \}_{i=1}^n. \quad (37)$$

These rigid transformations define isometries on \mathbb{R}^p since

$$\|\psi(\mathbf{x}_i) - \psi(\mathbf{x}_j)\|_2 = \|\mathbf{x}_i - \mathbf{x}_j\|_2, \quad (38)$$

$\forall \mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^p$. These are important in machine learning problems since they transform data without changing distances. An important property of the regularized NLL (25) is that the estimated textures of the model are invariant under rigid transformations of the data; see Proposition 8. 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 VII leverage this property and show robust performances when data undergo a rigid transformation during the testing phase.

Proposition 8 (Minima of $\mathcal{L}_{\mathcal{R}_\kappa}$ and rigid transformations): Let \mathcal{R}_κ be a regularization satisfying Assumption 1, and $\theta^* = (\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\tau})$ be a minimum of the regularized NLL (26) computed on data $\{\mathbf{x}_i\}_{i=1}^n$, i.e.

$$\theta^* \in \arg \min_{\theta \in \mathcal{M}_{p,n}} \mathcal{L}_{\mathcal{R}_\kappa}(\theta | \{\mathbf{x}_i\}_{i=1}^n),$$

then, given $\mathbf{Q} \in \mathcal{O}_p$ and $\boldsymbol{\mu}_0 \in \mathbb{R}^p$, a minimum of the regularized NLL computed on the transformed data $\psi(\{\mathbf{x}_i\}_{i=1}^n) = \{ \mathbf{Q}^T \mathbf{x}_i + \boldsymbol{\mu}_0 \}_{i=1}^n$ is $\phi(\theta^*) = (\mathbf{Q}^T \boldsymbol{\mu} + \boldsymbol{\mu}_0, \mathbf{Q}^T \boldsymbol{\Sigma} \mathbf{Q}, \boldsymbol{\tau})$, i.e.

$$\phi(\theta^*) \in \arg \min_{\theta \in \mathcal{M}_{p,n}} \mathcal{L}_{\mathcal{R}_\kappa}(\theta | \psi(\{\mathbf{x}_i\}_{i=1}^n)).$$

Proof: First of all, given $\mathbf{Q} \in \mathcal{O}_p$ and $\boldsymbol{\mu}_0 \in \mathbb{R}^p$, one can check that

$$\mathcal{L}(\phi(\theta) | \psi(\{\mathbf{x}_i\}_{i=1}^n)) = \mathcal{L}(\theta | \{\mathbf{x}_i\}_{i=1}^n),$$

where \mathcal{L} is the NLL defined in (5), $\theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\tau})$, $\phi(\theta) = (\mathbf{Q}^T \boldsymbol{\mu} + \boldsymbol{\mu}_0, \mathbf{Q}^T \boldsymbol{\Sigma} \mathbf{Q}, \boldsymbol{\tau})$ and ψ is defined in equation (37). Then, \mathcal{R}_κ satisfies Assumption 1 and thus only depends on the eigenvalues of the matrices $\tau_i \boldsymbol{\Sigma}$. This implies that $\mathcal{R}_\kappa(\phi(\theta)) = \mathcal{R}_\kappa(\theta)$ and hence we get that

$$\mathcal{L}_{\mathcal{R}_\kappa}(\phi(\theta) | \psi(\{\mathbf{x}_i\}_{i=1}^n)) = \mathcal{L}_{\mathcal{R}_\kappa}(\theta | \{\mathbf{x}_i\}_{i=1}^n).$$

This implies that if $\theta^* \in \arg \min_{\theta \in \mathcal{M}_{p,n}} \mathcal{L}_{\mathcal{R}_\kappa}(\theta | \{\mathbf{x}_i\}_{i=1}^n)$, then $\phi(\theta^*) \in \arg \min_{\theta \in \mathcal{M}_{p,n}} \mathcal{L}_{\mathcal{R}_\kappa}(\theta | \psi(\{\mathbf{x}_i\}_{i=1}^n))$, which concludes the proof. \square

VI. KL DIVERGENCE AND RIEMANNIAN CENTER OF MASS

In the previous section, we proposed to optimize the regularized NLL (26) of the NC-MSG (2). Once these parameters are estimated, they can be used as features for Riemannian classification/clustering algorithms [19], [20], [21], [22]. To do this classification/clustering, 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-MSGs (2). It benefits from a simple closed-form formula presented in Subsection VI-A. Secondly, simple classification algorithms, such as *K-means* or the *Nearest centroid 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 VI-B. Its computation is realized with Algorithm 1.

A. KL Divergence

Classification/clustering algorithms, such as *K-means* or the *Nearest centroid 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 (2). Indeed, the objective is to classify its parameters θ . In the context of measuring proximities between distributions admitting probability density functions, a classical divergence is the KL. The latter measures the similarity between two probability density functions. Definition 2 gives the general formula of the KL divergence.

Definition 2 (KL divergence): Given two probability density functions p and q defined on the sample space \mathcal{X} , the KL divergence is

$$\delta_{\text{KL}}(p, q) = \int_{\mathcal{X}} p(x) \log \left(\frac{p(x)}{q(x)} \right) dx.$$

Applied to NC-MSGs, the KL divergence is derived from the Gaussian one and is presented in Proposition 9. It benefits from a simple closed-form formula and therefore is of practical interest.

Proposition 9 (KL divergence): Given the r.v. $x = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ and two NC-MSGs of probability density functions $p_{\theta_1}(x) = \prod_{i=1}^n f(\mathbf{x}_i | (\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \tau_{1,i}))$ and $p_{\theta_2}(x) =$

$\prod_{i=1}^n f(\mathbf{x}_i | (\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2, \tau_{2,i}))$, the KL divergence is

$$\delta_{\text{KL}}(\theta_1, \theta_2) = \frac{1}{2} \left(\sum_{i=1}^n \frac{\tau_{1,i}}{\tau_{2,i}} \text{Tr}(\boldsymbol{\Sigma}_2^{-1} \boldsymbol{\Sigma}_1) + \sum_{i=1}^n \frac{1}{\tau_{2,i}} \Delta \boldsymbol{\mu}^T \boldsymbol{\Sigma}_2^{-1} \Delta \boldsymbol{\mu} + n \log \left(\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|} \right) - np \right),$$

with $\Delta \boldsymbol{\mu} = \boldsymbol{\mu}_2 - \boldsymbol{\mu}_1$.

Proof: The r.v. $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ can be vectorized into $\mathbf{x} = [\mathbf{x}_1^T, \dots, \mathbf{x}_n^T]^T \in \mathbb{R}^{np}$ which follows a multivariate Gaussian distribution of location the concatenation of the locations of $\mathbf{x}_1, \dots, \mathbf{x}_n$ and of block-diagonal covariance matrix whose elements are the covariance matrices of $\mathbf{x}_1, \dots, \mathbf{x}_n$. Thus, the KL divergence between the probability density functions p_{θ_1} and p_{θ_2} is the KL divergence between two multivariate Gaussian distributions whose covariance matrices are block diagonal. 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. \square

Finally, this KL divergence is non-symmetrical. We rely on the classical symmetrization to define the proposed divergence $\delta_{\mathcal{M}_{p,n}} : \mathcal{M}_{p,n} \times \mathcal{M}_{p,n} \rightarrow \mathbb{R}$,

$$\delta_{\mathcal{M}_{p,n}}(\theta_1, \theta_2) = \frac{1}{2} (\delta_{\text{KL}}(\theta_1, \theta_2) + \delta_{\text{KL}}(\theta_2, \theta_1)). \quad (39)$$

B. Center of Mass Computation

To implement simple machine learning algorithms such as *K-means* or the *Nearest centroid classifier* on $\mathcal{M}_{p,n}$, it remains to define an averaging algorithm. To do so, we leverage a classical definition of centers of mass which are minimizers of variances [23], [45]. 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

$$\underset{\theta \in \mathcal{M}_{p,n}}{\text{minimize}} \quad \frac{1}{M} \sum_{i=1}^M \delta_{\mathcal{M}_{p,n}}(\theta, \theta_i), \quad (40)$$

where $\delta_{\mathcal{M}_{p,n}}$ is the symmetrized KL divergence from Equation (39). To realize (40), Algorithm 1 can be employed.

VII. 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, the subsection VII-A studies the performance of Algorithm 1, in terms of speed of convergence on the cost functions (26) and (40) and in terms of estimation error on the cost function (22). Both studies are done through simulations. Algorithm 1 is shown to be fast. Indeed, it requires from 5 to 30 times fewer iterations to minimize costs functions (26) and (40) compared to other sophisticated optimization algorithms. This demonstrates the interest in the choice of the FIM to develop Riemannian optimization algorithms. Also, Algorithm 1 applied to the cost function (22) gives lower estimation errors than other

classical estimators such as the Tyler joint mean-scatter one and the Gaussian ones.

Second, an application on the crop classification dataset *Breizhcrops* [29] is presented in Subsection VII-B. This dataset consists of 600 000 time series to be classified into 9 classes. The application implements a *Nearest centroid classifier* on $\mathcal{M}_{p,n}$ using the divergence (39) and the Riemannian center of mass (40). Three results ensue. First, the proposed algorithms can be used on large-scale datasets. Second, the proposed regularization in Section V plays an important role in classification. Third, considering an NC-MSG (2) is interesting for time series especially when data undergo a rigid transformation (37).

Python code implementing the different experiments can be found at https://github.com/antoinecollas/optim_compound.

A. Simulation

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 $\boldsymbol{\Sigma} = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^T$. $\mathbf{U} \in \mathcal{O}_p$ is drawn from the uniform distribution on \mathcal{O}_p [46] using the module “scipy.stats” from the Scipy library [47]. Then, the elements on the diagonal of the diagonal matrix $\boldsymbol{\Lambda}$ are drawn from a χ_1^2 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 Algorithm 1 is studied on two cost functions: the regularized NLL (26) and the cost function (40) to compute the center of mass associated to the KL divergence of Proposition 9.

We begin with the minimization of the regularized NLL (26). $n = 150$ data $\mathbf{x}_i \in \mathbb{R}^{10}$ are drawn from a NC-MSG, i.e. $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \tau_i \boldsymbol{\Sigma})$. The parameter $\theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\tau})$ of this distribution is generated as explained in the introduction of this subsection with $\nu = 1$. Different parameters β in (26) are considered: $\beta \in \{0, 10^{-5}, 10^{-3}\}$. The chosen regularization is the L2 penalty from Table I. 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 6. The minimization is performed with three different algorithms.

- The plain conjugate gradient presented in [37]. It is a Riemannian conjugate gradient descent that uses a sum of three independent Riemannian metrics associated with the three parameters $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$, and $\boldsymbol{\tau}$. Thus, the corresponding Riemannian geometry is easier to derive but is not linked to the NC-MSG.
- The plain steepest descent. It is similar to the plain conjugate gradient. Still, it only uses the gradient as a direction of descent (and not a linear combination with the direction of descent of the previous step).
- The Algorithm 1 that leverages the information geometry of the NC-MSG presented in Section III-B.

The results of this experiment are presented in Figs. 1 and 3 in terms of iterations and computation time respectively. We

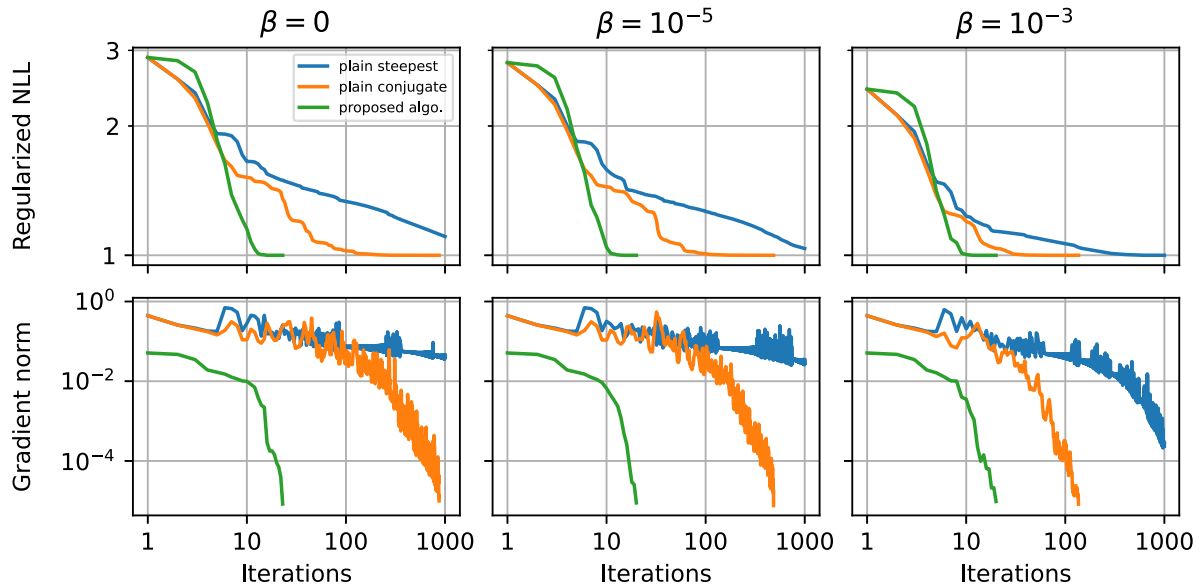


Fig. 1. Regularized NLL (26) and its gradient norm versus the iterations of three estimation algorithms. The chosen regularization is the L2 penalty (see Table I) and three different regularization intensities β are considered: 0 in the left column, 10^{-5} in the middle one, and 10^{-3} in the right one. Each estimation is performed on $n = 150$ samples in \mathbb{R}^{10} sampled from an NC-MSG. The regularized NLL are normalized so that their minimum value is 1.

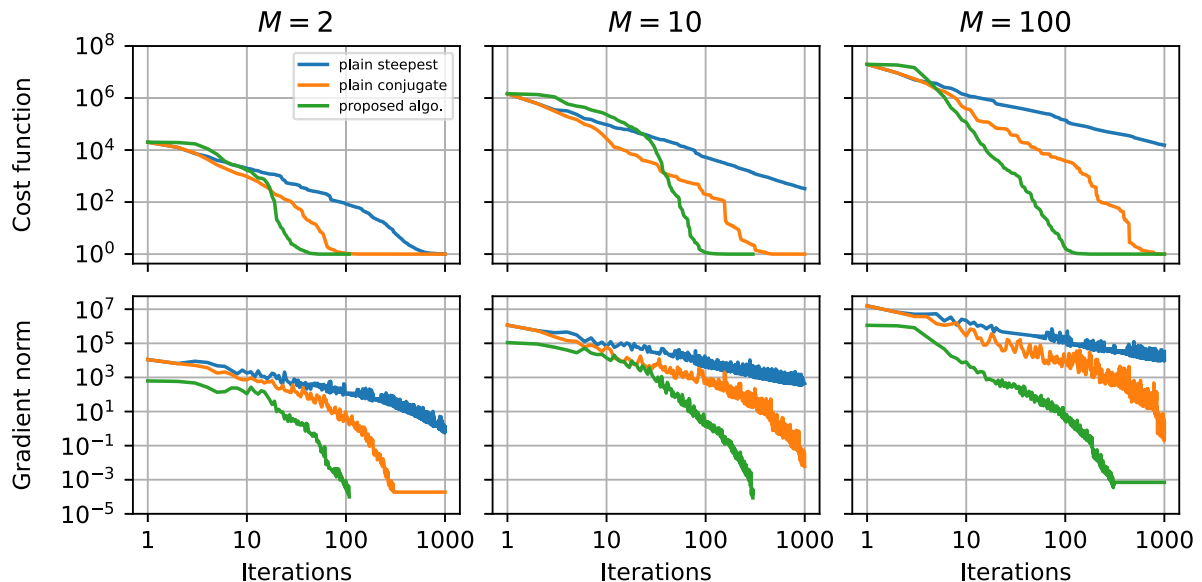


Fig. 2. Cost function (40) and its gradient norm versus the iterations of three estimation algorithms. The dimensions of the parameter space are $p = 10$ and $n = 150$. Three different numbers of points M are considered: 2 in the left column, 10 in the middle one, and 100 in the right one. The cost functions are normalized so that their minimum value is 1.

observe that Algorithm 1 is much faster than the two others regardless of the β parameter. Indeed, in the case $\beta \in \{0, 10^{-5}\}$, the Algorithm 1 is at least 100 times faster than the plain steepest descent and 10 times faster than the plain conjugate gradient. In the case of $\beta = 10^{-3}$, Algorithm 1 is at least 20 times faster than the plain steepest descent and 3 times faster than the plain conjugate gradient. Furthermore, we observe these results are valid either in the number of iterations or in computation time. Indeed, the three considered algorithms have iterations with similar computational costs in $\mathcal{O}(np^2 + p^3)$. Thus, a reduction in the number of iterations results in a reduction in computation time.

Then, a similar experiment is performed with the cost function (40) to compute the center of mass. $M \in \{2, 10, 100\}$ parameters θ are generated as described in the introduction of Subsection VII-A with $\nu = 1$. The minimization is performed with the same optimization algorithms as previously: the plain steepest descent, the plain conjugate gradient, and Algorithm 1. The results of this experiment are presented in Figs. 2 and 4 in terms of iterations and computation time, respectively.

We observe that Algorithm 1 is much faster than the two others regardless of M . Indeed, when $M = 2$, Algorithm 1 converges in 40 iterations whereas the plain conjugate gradient requires 300 iterations and the plain steepest descent still has not

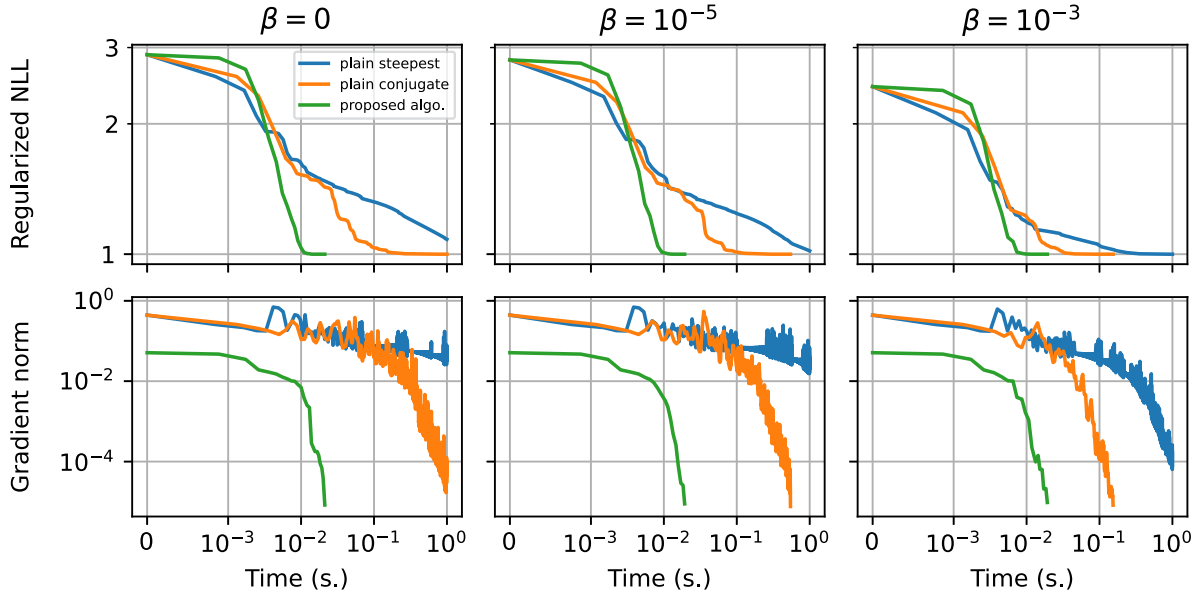


Fig. 3. Regularized NLL (26) and its gradient norm versus the computation time of three estimation algorithms. The chosen regularization is the L2 penalty (see Table I) and three different regularization intensities β are considered: 0 in the left column, 10^{-5} in the middle one, and 10^{-3} in the right one. Each estimation is performed on $n = 150$ samples in \mathbb{R}^{10} sampled from an NC-MSG. The regularized NLL are normalized so that their minimum value is 1.

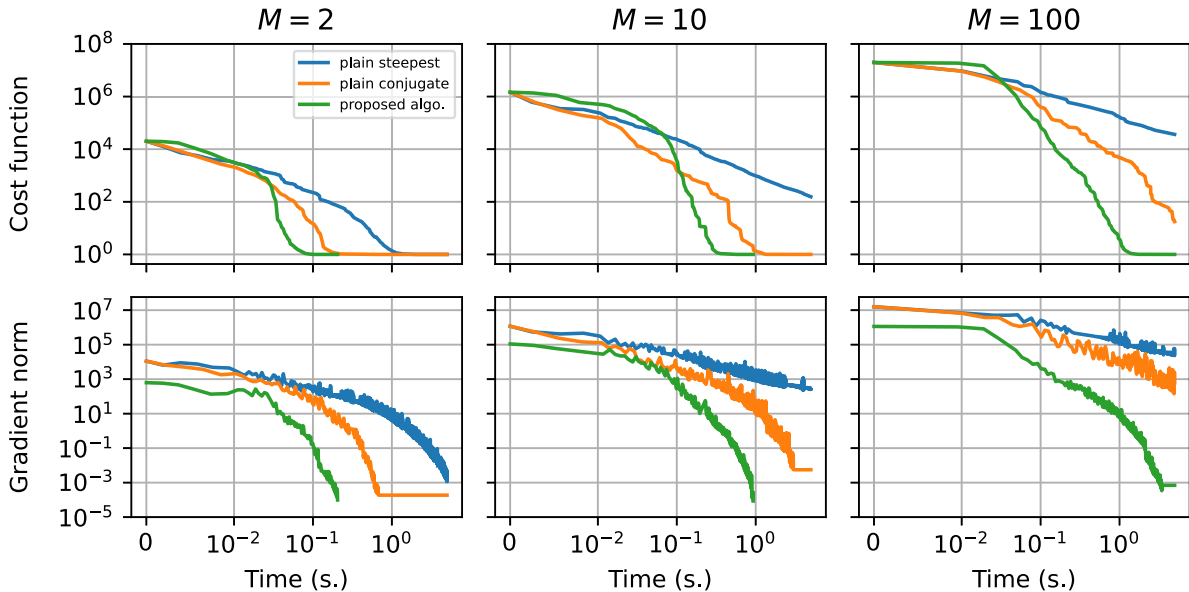


Fig. 4. Cost function (40) and its gradient norm versus the computation time of three estimation algorithms. The dimensions of the parameter space are $p = 10$ and $n = 150$. Three different numbers of points M are considered: 2 in the left column, 10 in the middle one, and 100 in the right one. The cost functions are normalized so that their minimum value is 1.

converged after 1000 iterations. When $M \in \{10, 100\}$, Algorithm 1 converges in less than 60 iterations which is 4 times faster than the plain conjugate gradient. It should be noted that the plain steepest descent has not converged after 1000 iterations in the cases $M \in \{100, 1000\}$. Once again, these results are valid either in the number of iterations or in computation time since the three considered algorithms have iterations with similar computational costs in $\mathcal{O}(M(n + p^3))$. Hence, reducing the number of iterations implies a reduction in computation time.

The estimation error made by Algorithm 1 applied on the NLL (5) is studied with numerical experiments on simulated

data. $n \in [20, 1000]$ data \mathbf{x}_i are sampled from the NC-MSG (2). The parameter $\theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\tau})$ of this distribution is generated as presented in the introduction of Subsection VII-A with $\nu = 0.1$ to have heterogeneous textures τ_i . The considered estimators for this numerical experiment are the following:

- Gaussian estimators: the sample mean $\boldsymbol{\mu}^G = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$ and the SCM $\boldsymbol{\Sigma}^G = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu}^G)(\mathbf{x}_i - \boldsymbol{\mu}^G)^T$.
- Tyler's joint location-scatter matrix estimator [5] denoted $\boldsymbol{\mu}^{\text{Ty}}$ and $\boldsymbol{\Sigma}^{\text{Ty}}$.
- Tyler's M -estimator with location known [5]. The sampled data \mathbf{x}_i are centered with the true location $\boldsymbol{\mu}$, and then $\boldsymbol{\Sigma}$ is estimated. This estimator is denoted $\boldsymbol{\Sigma}^{\text{Ty}, \boldsymbol{\mu}}$.

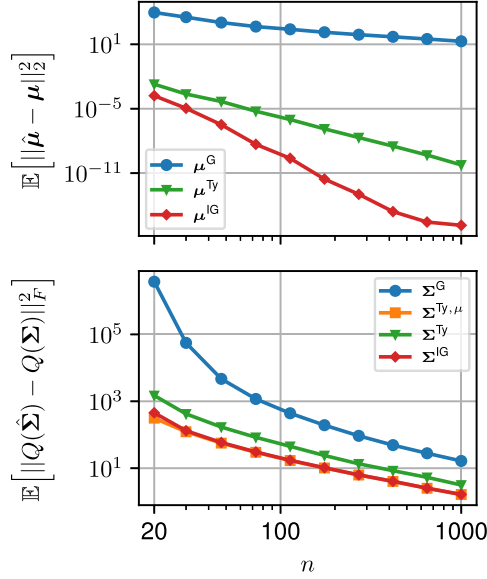


Fig. 5. MSE over 2000 simulated sets $\{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^{10}$ versus the number samples \mathbf{x}_i for the considered estimators $\hat{\boldsymbol{\mu}} \in \{\boldsymbol{\mu}^G, \boldsymbol{\mu}^{\text{Ty}}, \boldsymbol{\mu}^{\text{IG}}\}$ and $\hat{\boldsymbol{\Sigma}} \in \{\boldsymbol{\Sigma}^G, \boldsymbol{\Sigma}^{\text{Ty}, \boldsymbol{\mu}}, \boldsymbol{\Sigma}^{\text{Ty}}, \boldsymbol{\Sigma}^{\text{IG}}\}$. The proposed estimators $\boldsymbol{\mu}^{\text{IG}}$ and $\boldsymbol{\Sigma}^{\text{IG}}$ are computed as in (22) using Algorithm 1 and the Q function normalizes scatter matrices, i.e. $\forall \boldsymbol{\Sigma} \in \mathcal{S}_p^{++}$, $Q(\boldsymbol{\Sigma}) = |\boldsymbol{\Sigma}|^{-\frac{1}{p}} \boldsymbol{\Sigma}$.

- The proposed estimator denoted $\boldsymbol{\mu}^{\text{IG}}$ and $\boldsymbol{\Sigma}^{\text{IG}}$. Algorithm 1 minimizes the NLL (5). The initialization is the Gaussian maximum likelihood i.e. $\theta_{\text{init}} = (\boldsymbol{\mu}^G, \boldsymbol{\Sigma}^G, \mathbf{1}_n)$, where $\boldsymbol{\mu}^G = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$, $\boldsymbol{\Sigma}^G = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu}^G)(\mathbf{x}_i - \boldsymbol{\mu}^G)^T$ and $\mathbf{1}_n = (1, \dots, 1)^T$.

The estimation errors are measured with the Mean Squared Errors (MSE). These errors are computed as $\mathbb{E}[\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_2^2]$ and $\mathbb{E}[\|Q(\hat{\boldsymbol{\Sigma}}) - Q(\boldsymbol{\Sigma})\|_F^2]$, with $Q(\boldsymbol{\Sigma}) = |\boldsymbol{\Sigma}|^{-\frac{1}{p}} \boldsymbol{\Sigma}$, for the estimated location $\hat{\boldsymbol{\mu}}$ and the estimated scatter $\hat{\boldsymbol{\Sigma}}$ respectively, with 2000 Monte-Carlo. The MSE on the location and the scatter versus the number of samples \mathbf{x}_i are plotted in Fig. 5. First, we observe in both figures that the Gaussian estimators have a high MSE. This shows the interest in considering robust estimators such as Tyler’s joint location-scatter matrix estimator or the proposed one when the textures τ_i are heterogeneous. Then, the proposed estimators realize a much lower MSE than Tyler’s joint location-scatter 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 Tyler’s M -estimator for the scatter estimation. Indeed, when the location is known, Tyler’s M -estimator is the MLE of the NC-MSG (2). We observe that when enough samples are provided, the proposed estimator matches the MSE of Tyler’s M -estimator. Overall, this experimental subsection illustrates the good performance of the proposed estimator when data are sampled from a NC-MSG (2).

B. Application

In the previous subsection, the different theoretical results derived in Sections from III to VI showed several interests in synthetic data. We now focus on applying a *Nearest centroid*

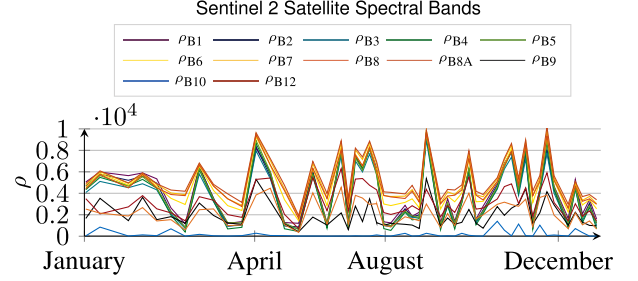


Fig. 6. Reflectances of a Sentinel-2 time series of meadows from the *Breizhcrocs* dataset. Figure courtesy [29].

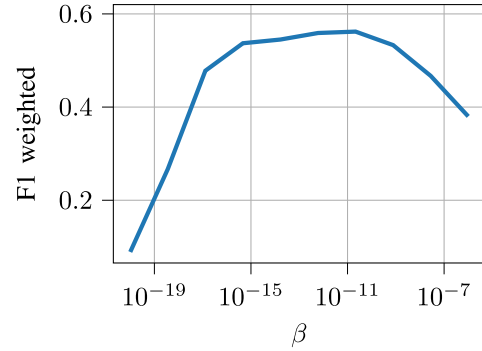


Fig. 7. “F1 weighted” metric achieved by the proposed *Nearest centroid classifier* on the *Breizhcrocs* dataset versus the parameter of regularization β in (25). The chosen regularization is the L2 penalty from Table 1.

classifier on $\mathcal{M}_{p,n}$ to real data using the estimation framework developed in Section V, the divergence and the Riemannian center of mass from Section VI as well as the optimization framework from Section IV. This classifier is compared with several other *Nearest centroid classifiers* associated with different estimators and divergences.

To do so, we consider the dataset *Breizhcrocs* [29]: a large-scale dataset of more than 600 000 crop time series from the Sentinel-2 satellite to classify. More specifically, for each crop $n = 45$ observations $\mathbf{x}_i \in \mathbb{R}^p$ are measured over time. Each \mathbf{x}_i contains reflectance measurements of $p = 13$ spectral bands. Then, these measurements are concatenated into one batch $\mathbf{X}_j = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$. Hence, we get one matrix \mathbf{X}_j per crop and each one belongs to an unknown class $y \in \llbracket 1, K \rrbracket$. These $K = 9$ classes represent crop types such as nuts, barley, or wheat and are heavily imbalanced, i.e. some classes are much more represented than others. An example of a time series of meadows is presented in Fig. 6. We apply a single preprocessing step: all the data are centered using the global mean. For simplicity, the matrix \mathbf{X}_j is noted \mathbf{X} in the following.

To classify these crops, we apply a *Nearest centroid classifier* on descriptors. Indeed, the use of statistical descriptors is a classical procedure in machine learning as they are often more discriminative than raw data (see e.g. [19], [20]). Hence, this classification algorithm works in three steps.

- 1) For each batch \mathbf{X} , a descriptor is computed, e.g. the parameter $\theta \in \mathcal{M}_{p,n}$ from the minimization of the regularized NLL (25).

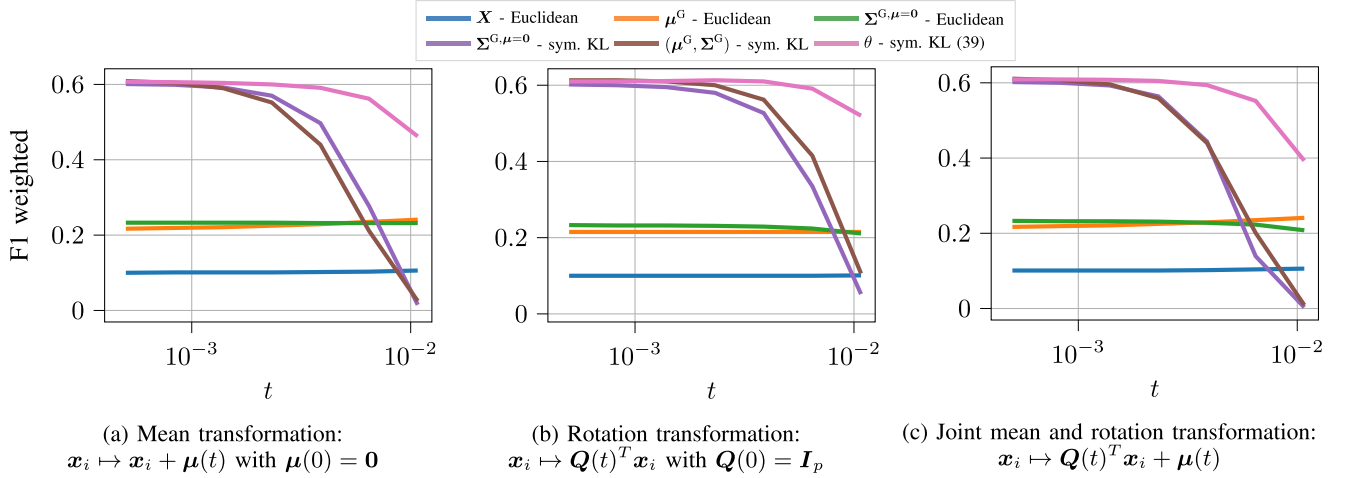


Fig. 8. “F1 weighted” metric versus the parameter t associated with three transformations applied to the test set of the *Breizhcrocs* dataset. The different *Nearest centroid classifiers* estimate the barycentres on the training data without transformations. Then, the classification is performed on 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 centroid classifiers* are compared: each is a combination of an estimator, a divergence, and its associated center of mass computation. The latter uses the Equations (26), (39) and (40) for the estimation, the divergence and the center of mass computation respectively. The regularization is the L2 penalty from Table I and β is fixed at 10^{-11} .

- 2) Then, on the training set, the center of mass of the descriptors of each 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 (40).
- 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 centroid 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 (39) between NC-MSGs. For each divergence, several *Nearest centroid classifiers* are derived using several estimators. These estimators correspond to different assumptions on the data.

Three *Nearest centroid classifiers* rely on the Euclidean distance between matrices. Given two matrices \mathbf{A} and \mathbf{B} of the same size, the Euclidean distance is $d(\mathbf{A}, \mathbf{B}) = \|\mathbf{A} - \mathbf{B}\|_F$. The center of mass of a given set $\{\mathbf{A}_i\}_{i=1}^M$ is the arithmetic mean $\frac{1}{M} \sum_{i=1}^M \mathbf{A}_i$ which is the solution of minimize $\frac{1}{M} \sum_{i=1}^M \|\mathbf{Y} - \mathbf{A}_i\|_F^2$. From this geometry, three *Nearest centroid classifiers* are derived using three estimators: the batch itself \mathbf{X} , the sample mean $\boldsymbol{\mu}^G$ and $\boldsymbol{\Sigma}^{G,\mu=0} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^T$. The last two estimators correspond to the assumption that data follow a Gaussian distribution (either with the same scatter matrix for all batches or the same location).

Two *Nearest centroid 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 $v_1 = (\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \in \mathcal{M}_p$ and $v_2 = (\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, this divergence is given by $\delta_{\mathcal{M}_p}(v_1, v_2) = \frac{1}{2} (\delta_{\text{KL}}(v_1, v_2) + \delta_{\text{KL}}(v_2, v_1))$ where $\delta_{\text{KL}}(v_1, v_2) = \frac{1}{2} (\text{Tr}(\boldsymbol{\Sigma}_2^{-1} \boldsymbol{\Sigma}_1) + \Delta \boldsymbol{\mu}^T \boldsymbol{\Sigma}_2^{-1} \Delta \boldsymbol{\mu} + \log(\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|}) - p)$. The center of mass of $\{v_i\}_{i=1}^M$ is the solution of minimize $v \in \mathcal{M}_p \sum_{i=1}^M \delta_{\mathcal{M}_p}(v, v_i)$.

Then, two *Nearest centroid classifiers* are derived using two estimators: $\boldsymbol{\Sigma}^{G,\mu=0}$ and the MLE of the Gaussian distribution $(\boldsymbol{\mu}^G, \boldsymbol{\Sigma}^G)$.

Finally, the proposed *Nearest centroid classifier* on $\mathcal{M}_{p,n}$ relies on the symmetrized KL divergence (39) between NC-MSGs. The center of mass is computed as explained in the subsection VI-B and the estimation is described in Section V with the L2 penalty for the regularization. For initialization, we used the arithmetic mean, i.e. given a set of parameters $\{\theta_i \triangleq (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \boldsymbol{\tau}_i)\}_{i=1}^M$, with $\theta_{\text{init}} = (\boldsymbol{\mu}_{\text{mean}}, \boldsymbol{\Sigma}_{\text{mean}}, N(\boldsymbol{\tau}_{\text{mean}}))$, where $\boldsymbol{\mu}_{\text{mean}} = \frac{1}{M} \sum_{i=1}^M \boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_{\text{mean}} = \frac{1}{M} \sum_{i=1}^M \boldsymbol{\Sigma}_i$, $\boldsymbol{\tau}_{\text{mean}} = \frac{1}{M} \sum_{i=1}^M \boldsymbol{\tau}_i$ and N is the normalization function: $\forall \mathbf{x} = (x_i)_{1 \leq i \leq n} \in (\mathbb{R}_*^+)^n$, $N(\mathbf{x}) = (\prod_{i=1}^n x_i)^{-\frac{1}{n}} \mathbf{x}$.

The data are divided into two sets: a training set and a test set with 485 649 and 122 614 batches respectively [29]. Among the six *Nearest centroid classifiers*, only the one on $\mathcal{M}_{p,n}$ has a hyperparameter which the parameter β of the regularized NLL (25). Several values of β are tested on a training set and a validation set, and both are subsets of the original training set. The performance is measured with the “F1 weighted” metric used in [29] and is plotted in Fig. 7. The value of β with the highest “F1 weighted” metric is 10^{-11} . Hence, we use this value in the rest of the article. Then, we propose an experiment to illustrate Proposition 8 on the invariance of the estimation of textures under rigid transformations. Indeed, we train the six *Nearest centroid classifiers* on a subset of the original training set and apply them to the full test set with a rigid transformation. Thus, the more a *Nearest centroid classifier* is robust to these rigid transformations, the better the “F1 weighted” metric. Given $t \in [0, 1]$, three different rigid transformations are performed: transformation of the mean $\mathbf{x}_i \mapsto \mathbf{x}_i + \boldsymbol{\mu}(t)$ with $\boldsymbol{\mu}(t) = t \mathbf{a}$ for a given $\mathbf{a} \in \mathbb{R}^p$, rotation transformation $\mathbf{x}_i \mapsto \mathbf{Q}(t)^T \mathbf{x}_i$ with $\mathbf{Q}(t) = \exp(t\xi)$ for a given skew-symmetric $\xi \in \mathbb{R}^{p \times p}$ (hence

$\mathbf{Q}(t) \in \mathcal{O}_p$), and the joint mean and rotation transformation $\mathbf{x}_i \mapsto \mathbf{Q}(t)^T \mathbf{x}_i + \boldsymbol{\mu}(t)$. It should be noted that at $t = 0$, the data are left unchanged. The results are presented in Fig. 8.

The conclusions of these experiments are fourfold. First, the proposed *Nearest centroid classifier* applies to large-scale datasets such as the *Breizhcrops* dataset. Second, the regularization proposed in Section V is important to get good classification performance. Indeed, we observe from Fig. 7 that if β is too small, then the “F1 weighted” metric becomes very low. Also, if β is too large, then the “F1 weighted” metric also becomes very low. Third, using KL divergences and their associated centers of mass to classify estimators gives much better performance than the classical Euclidean distance. Indeed, even when data do not undergo rigid transformations, *Nearest centroid classifiers* based on KL divergences outperform Euclidean *Nearest centroid classifiers* in Fig. 8. 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 to the data. Indeed, in Fig. 8, we observe a large performance improvement when data are considered distributed from a NC-MSG and undergo rigid transformations.

VIII. CONCLUSION

In this article, we proposed a Riemannian gradient descent algorithm based on the Fisher-Rao information geometry of the NC-MSG. This algorithm is leveraged for two problems: parameter estimation and computation of centers of mass. The estimation problem of the NC-MSG is not straightforward. Indeed, a major issue is that the existence of a solution to the NLL minimization problem is not guaranteed. To overcome this issue, we proposed a class of regularized NLLs that make the trade-off between a white Gaussian distribution and the NC-MSG. These functions are guaranteed to have a minimum, and this result holds without conditions on the samples. Furthermore, we derived the KL divergence between NC-MSGs which enabled us to define the centers of mass of NC-MSGs as minimization problems. The latter is solved using the proposed Riemannian gradient descent. Simulations have shown that the proposed Riemannian gradient descent is fast on both minimization problems. Also, a *Nearest centroid classifier* based on the KL divergence has been implemented. It has been applied on the large-scaled dataset *Breizhcrops* and showed robustness to transformations of the test set.

REFERENCES

- [1] F. Kai-Tai and Z. Yao-Ting, *Generalized Multivariate Analysis*, vol. 19. Berlin, Germany: Springer, 1990.
- [2] R. A. Maronna, “Robust M-estimators of multivariate location and scatter,” *Ann. Statist.*, vol. 4, no. 1, pp. 51–67, 1976.
- [3] E. Ollila, D. E. Tyler, V. Koivunen, and H. V. Poor, “Complex elliptically symmetric distributions: Survey, new results and applications,” *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 5597–5625, Nov. 2012.
- [4] A. Wiesel, “Unified framework to regularized covariance estimation in scaled Gaussian models,” *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 29–38, Jan. 2012.
- [5] D. E. Tyler, “A distribution-free M-estimator of multivariate scatter,” *Ann. Statist.*, vol. 15, no. 1, pp. 234–251, 1987.
- [6] 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,” *IEEE Trans. Signal Process.*, vol. 50, no. 8, pp. 1908–1915, Aug. 2002.
- [7] F. Pascal, P. Forster, J.-P. Ovarlez, and P. Larzabal, “Performance analysis of covariance matrix estimates in impulsive noise,” *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2206–2217, Jun. 2008.
- [8] G. Frahm and U. Jaekel, “Tyler’s M-estimator, random matrix theory, and generalized elliptical distributions with applications to finance,” in *Random Matrix Theory, Generalized Elliptical Distributions With Appl. Finance*, Oct. 2008.
- [9] T. Zhang, X. Cheng, and A. Singer, “Marčenko–Pastur law for Tyler’s M-estimator,” *J. Multivariate Anal.*, vol. 149, pp. 114–123, 2016.
- [10] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ, USA: Princeton Univ. Press, 2008.
- [11] N. Boumal, *An Introduction to Optimization on Smooth Manifolds*. Cambridge, U.K.: Cambridge Univ. Press, Mar. 2023.
- [12] F. Nielsen, “The many faces of information geometry,” *Not. Amer. Math. Soc.*, vol. 69, pp. 36–45, 2022.
- [13] F. Pascal, Y. Chitour, and Y. Quek, “Generalized robust shrinkage estimator and its application to STAP detection problem,” *IEEE Trans. Signal Process.*, vol. 62, no. 21, pp. 5640–5651, Nov. 2014.
- [14] Y. Sun, P. Babu, and D. P. Palomar, “Regularized Tyler’s scatter estimator: Existence, uniqueness, and algorithms,” *IEEE Trans. Signal Process.*, vol. 62, no. 19, pp. 5143–5156, Oct. 2014.
- [15] E. Ollila and D. E. Tyler, “Regularized M-estimators of scatter matrix,” *IEEE Trans. Signal Process.*, vol. 62, no. 22, pp. 6059–6070, Nov. 2014.
- [16] Y. Sun, P. Babu, and D. P. Palomar, “Regularized robust estimation of mean and covariance matrix under heavy-tailed distributions,” *IEEE Trans. Signal Process.*, vol. 63, no. 12, pp. 3096–3109, Jun. 2015.
- [17] A. Breloy, E. Ollila, and F. Pascal, “Spectral shrinkage of Tyler’s M-estimator of covariance matrix,” in *Proc. IEEE 8th Int. Workshop Comput. Adv. Multi-Sensor Adaptive Process.*, 2019, pp. 535–538.
- [18] M. Yi and D. E. Tyler, “Shrinking the covariance matrix using convex penalties on the matrix-log transformation,” *J. Comput. Graphical Statist.*, vol. 30, no. 2, pp. 442–451, 2020.
- [19] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, “Multiclass brain-computer interface classification by Riemannian geometry,” *IEEE Trans. Biomed. Eng.*, vol. 59, no. 4, pp. 920–928, Apr. 2012.
- [20] O. Tuzel, F. Porikli, and P. Meer, “Human detection via classification on Riemannian manifolds,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.
- [21] O. Tuzel, F. Porikli, and P. Meer, “Pedestrian detection via classification on Riemannian manifolds,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1713–1727, Oct. 2008.
- [22] P. Formont, J.-P. Ovarlez, and F. Pascal, “On the use of matrix information geometry for polarimetric SAR image classification,” in *Matrix Information Geometry*. Berlin, Germany: Springer, 2013, pp. 257–276.
- [23] H. Karcher, “Riemannian center of mass and mollifier smoothing,” *Commun. Pure Appl. Math.*, vol. 30, no. 5, pp. 509–541, 1977.
- [24] M. Arnaudon, F. Barbaresco, and L. Yang, *Medians and Means in Riemannian Geometry: Existence, Uniqueness and Computation*. Berlin, Germany: Springer 2013.
- [25] S. Chevallier, E. K. Kalunga, Q. Barthélemy, and E. Monacelli, “Review of Riemannian distances and divergences, applied to SSVEP-based BCI,” *Neuroinformatics*, vol. 19, no. 1, pp. 93–106, 2021.
- [26] M. Calvo and J. M. Oller, “An explicit solution of information geodesic equations for the multivariate normal model,” *Statist. Risk Model.*, vol. 9, no. 1/2, pp. 119–138, 1991.
- [27] M. Tang, Y. Rong, J. Zhou, and X. R. Li, “Information geometric approach to multisensor estimation fusion,” *IEEE Trans. Signal Process.*, vol. 67, no. 2, pp. 279–292, Jan. 2019.
- [28] 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 *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 5697–5701.
- [29] M. Rußwurm, C. Pelletier, M. Zollner, S. Lefèvre, and M. Körner, “Breizhcrops: A time series dataset for crop type mapping,” *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 43, pp. 1545–1551, 2020.
- [30] E. Ollila, D. E. Tyler, V. Koivunen, and H. V. Poor, “Compound-Gaussian clutter modeling with an inverse Gaussian texture distribution,” *IEEE Signal Process. Lett.*, vol. 19, no. 12, pp. 876–879, Dec. 2012.

- [31] 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," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 34–48, Jan. 2008.
- [32] A. Breloy, G. Ginolhac, A. Renaux, and F. Bouchard, "Intrinsic Cramér–Rao bounds for scatter and shape matrices estimation in CES distributions," *IEEE Signal Process. Lett.*, vol. 26, no. 2, pp. 262–266, Feb. 2019.
- [33] F. Bouchard, A. Mian, J. Zhou, S. Said, G. Ginolhac, and Y. Berthoumieu, "Riemannian geometry for compound Gaussian distributions: Application to recursive change detection," *Signal Process.*, vol. 176, 2020, Art. no. 107716.
- [34] A. Wiesel and T. Zhang, "Structured Robust Covariance Estimation, *Now Foundations and Trends*," *Signal Process.*, vol. 8, pp. 127–216, 2015.
- [35] Y. Sun, P. Babu, and D. P. Palomar, "Robust estimation of structured covariance matrix for heavy-tailed elliptical distributions," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3576–3590, Jul. 2016.
- [36] B. Meriaux, C. Ren, M. N. El Korso, A. Breloy, and P. Forster, "Robust estimation of structured scatter matrices in (MIS) matched models," *Signal Process.*, vol. 165, pp. 163–174, 2019.
- [37] 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 *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 5160–5164.
- [38] S. Amari, *Information Geometry and Its Applications*, 1st ed. New York, NY, USA: Springer, 2016.
- [39] L. T. Skovgaard, "A Riemannian geometry of the multivariate normal model," *Scand. J. Statist.*, vol. 11, no. 4, pp. 211–223, 1984.
- [40] M. Pilté and F. Barbaresco, "Tracking quality monitoring based on information geometry and geodesic shooting," in *Proc. IEEE 17th Int. Radar Symp.*, 2016, pp. 1–6.
- [41] D. Maclaurin, D. Duvenaud, and R. P. Adams, "Autograd: Effortless gradients in pure numpy," in *Proc. AutoML Workshop ICML*, 2015, vol. 238, no. 5.
- [42] J. Bradbury et al., "JAX: Composable transformations of Python NumPy programs," 2018. [Online]. Available: <http://github.com/google/jax>
- [43] A. Wiesel, "Geodesic convexity and covariance estimation," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6182–6189, Dec. 2012.
- [44] E. Ollila, D. P. Palomar, and F. Pascal, "Shrinking the eigenvalues of M-estimators of covariance matrix," *IEEE Trans. Signal Process.*, vol. 69, pp. 256–269, 2021.
- [45] M. Moakher, "A differential geometric approach to the geometric mean of symmetric positive-definite matrices," *SIAM J. Matrix Anal. Appl.*, vol. 26, no. 3, pp. 735–747, 2005.
- [46] F. Mezzadri, "How to generate random matrices from the classical compact groups," *Notices Amer. Math. Soc.*, vol. 54, pp. 592–604, 2007.
- [47] P. Virtanen et al., "SciPy 1.0: Fundamental algorithms for scientific computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.
- [48] S. Smith, "Covariance, subspace, and intrinsic Cramér–Rao bounds," *IEEE Trans. Signal Process.*, vol. 53, pp. 1610–1630, 2005.
- [49] F. Bouchard, A. Breloy, G. Ginolhac, A. Renaux, and F. Pascal, "A Riemannian framework for low-rank structured elliptical models," *IEEE Trans. Signal Process.*, vol. 69, pp. 1185–1199, 2021.



Antoine Collas received the master's degree in computer science and engineering from UTC, Compiègne, France, in 2019 and the Ph.D. diploma in signal and image processing from the University of Paris-Saclay, Gif-sur-Yvette, France. The Ph.D. was conducted in the French-Singaporean SONDRALaboratory, CentraleSupélec, Gif-sur-Yvette. He is currently a Postdoctoral Researcher with the Mind team, Inria Saclay, Palaiseau, France. His research interests include statistical signal processing, riemannian geometry, and machine learning.

Arnaud Breloy (Member, IEEE) received the master's degree in signal and image processing from the University of Aix-Marseille, Marseille, France, in 2012, the graduate degree from Ecole Centrale Marseille, Marseille, in 2013, and the Ph.D. degree in signal processing from the University of Paris-Saclay, Gif-sur-Yvette, France, in 2015. The thesis was done in collaboration between SATIE (ENS Cachan) and SONDRALaboratory (CentraleSupélec) laboratories. Since 2016, he has been an Associate Professor with the LEMELaboratory, University Paris Nanterre, Nanterre, France. His research interests include statistical signal processing, dimension reduction, and information geometry.

Chengfang Ren (Member, IEEE) received the Agrégation in applied physics, the M.Sc. degree in electrical engineering from Ecole Normale Supérieure de Cachan, Cachan, France, in 2011 and 2012, respectively, and the Ph.D. degree from Paris-Sud XI University, Orsay, France, in 2015. From 2015 to 2016, he was a Postdoctoral Research Scientist with GIPSA Laboratory, Saint Martin d'Hères, France. He is currently an Assistant Professor with the SONDRALaboratory, French-Singaporean Laboratory, CentraleSupélec, Gif-sur-Yvette, France. His research interests include statistical signal processing and machine and deep learning for radar applications.



Guillaume Ginolhac (Senior Member, IEEE) received the master's degree in electrical engineer and the Ph.D. degree in signal processing from Grenoble-INP University, Grenoble, France, in 1997 and 2001, respectively. He received the Research Directorship Habilitation thesis in signal processing from the ENS Cachan Institute, Cachan, France, in 2011. From 2002 to 2012, he was an Associate Professor with SATIE Laboratory, Gif-sur-Yvette, France, and University Paris Nanterre, Nanterre, France. Since 2012, he has been a Full Professor with LISTIC Laboratory, University Savoie Mont-Blanc, Annecy, France. Since 2018, he has been the Head of the Learning, Fusion and Remote Sensing Group, LISTIC. His research interests include estimation and detection theory for statistical signal processing, applications to array processing, and radar remote sensing.



Jean-Philippe Ovarlez (Member, IEEE) was born in Denain, France, in 1963. He received the engineering degree jointly from Ecole Supérieure d'Electronique Automatique et Informatique (ESIEA), Paris, France, the Diplôme d'Etudes Approfondies degree in signal processing from the University of Paris XI, Orsay, France, in 1987, and the Ph.D. degree in physics from the University of Paris VI, Paris, in 1992. In 2011, he obtained a Research Directorship Habilitation thesis in signal processing from the University of Paris-Sud, Orsay, and his qualification for the University Professor position. His research interests include statistical signal processing for radar and SAR applications such as time-frequency, imaging, detection, and parameter estimation. In 1992, he joined the Electromagnetic and Radar Division of the French Aerospace Lab (ONERA), Palaiseau, France, where he is currently the Research Director and a Member of the Scientific Committee of ONERA Physics Branch. Since 2008, he has been attached part-time to Centrale-Supélec SONDRALab, in charge of signal processing activities supervision. In 2015, he became a Member of the Special Area Team in Theoretical and Methodological Trends in Signal Processing, EURASIP, and treasurer of the IEEE GRSS French Chapter in 2016.