Random matrix theory improved Fréchet mean of symmetric positive definite matrices

Florent Bouchard [1] **Ammar Mian** [2] **Malik Tiomoko** [3] **Guillaume Ginolhac** [2] **Frédéric Pascal** [1]

Abstract

In this study, we consider the realm of covariance matrices in machine learning, particularly focusing on computing Fréchet means on the manifold of symmetric positive definite matrices, commonly referred to as Karcher or geometric means. Such means are leveraged in numerous machine learning tasks. Relying on advanced statistical tools, we introduce a random matrix theory based method that estimates Fréchet means, which is particularly beneficial when dealing with low sample support and a high number of matrices to average. Our experimental evaluation, involving both synthetic and real-world EEG and hyperspectral datasets, shows that we largely outperform state-of-the-art methods.

1. Introduction

Covariance matrices are of significant interest in machine learning, especially in scenarios with a limited number of labeled data or when dealing with high intra-class variability, as seen in EEG (Barachant et al., 2011) and remote sensing (Rußwurm et al., 2020). Numerous machine learning algorithms have been developed when features are covariance matrices, and therefore symmetrical positive-definite matrices (SPD). A common and notable algorithm in this realm is the well-established nearest centroid. SPD matrices find their use in deep learning networks (Huang & Van Gool, 2017; Brooks et al., 2019), metric learning (Zadeh et al., 2016; Harandi et al., 2017), domain adaptation (Kobler et al., 2022), privacy protection (Reimherr et al., 2021). A pivotal component in most machine learning algorithms that utilize SPD matrices is the computation of a class barycenter. For SPD matrices, this barycenter is known as the Fréchet mean (or Karcher mean) (Bhatia, 2015). This mean is used, for example, for nearest centroid (Tuzel et al., 2008), pooling in SPD deep learning networks (Brooks et al., 2019) and metric learning (Zadeh et al., 2016). The optimal solution is not available analytically necessitating the use of iterative algorithms often based on deriving a Riemannian gradient (Boumal, 2023). These algorithms are grounded in Riemannian geometry, since matrices belong to specific manifolds depending on their specific properties (fair SPD, low rank, *etc.*) and the chosen metric. The geometry is often the classical one given for SPD matrices, but alternatives geometries are available to perform this algorithm such as Bures-Wassertein (Han et al., 2021), log-Euclidean (Utpala et al., 2023) and even for a more general manifold (Lou et al., 2020).

These algorithms generally perform effectively, yet there are instances where the solution may be numerically unfeasible, particularly with the presence of a singular matrix. In this case, the most common solution is to regularize each of the covariance matrices. There is a plethora of work in this field. The most common regularization technique involves shrinking the covariance estimate towards the identity matrix, introducing a parameter upon which the new estimate hinges. Numerous methods have been proposed to optimally estimate this parameter according to a chosen criterion. A seminal contribution in this domain is by (Ledoit & Wolf, 2004), where the mean square error (MSE) between the true covariance and the regularized covariance is used. The optimal parameter is finally calculated on the basis of statistical consistency considerations. Improvements have been proposed in (Ledoit & Wolf, 2015) and (Ledoit & Wolf, 2020). Extensions to non-Gaussian data have also been proposed (Ollila & Tyler, 2014; Pascal et al., 2014).

In (Tiomoko et al., 2019), a novel approach was introduced, utilizing a distance-based criterion. This method draws upon the innovative distance presented in (Couillet et al., 2019), which offers a consistent estimation of the true distance between two matrices. This new estimate is derived from the tools of random matrix theory, which enables us to study the statistical behavior of the eigenvalues and eigenvectors of random matrices in a high-dimensional regime (when the size of the data p and the number of samples n grow at the same rate). While this new estimator demonstrates promising potential in terms of estimation, it is not without

[*]Equal contribution [1]Université Paris Saclay, CNRS, CentraleSupélec, L2S [2]Université Savoie Mont Blanc, LISTIC [3]Huawei Paris Research Center. Correspondence to: Florent Bouchard <florent.bouchard@cnrs.fr>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

its practical challenges, such as the selection of initial values and the definition of an appropriate stopping criterion. A critical issue is its non-compliance with certain conditions set forth in (Tiomoko et al., 2019) concerning the independence of the two matrices in the distance. Furthermore, similar to traditional regularization methods, this approach only regularizes the eigenvalues, leaving the eigenspaces unaltered. As indicated by empirical results, this form of regularization alone may not suffice to deliver optimal performance in classification or clustering tasks.

We recognize the significance of tailoring regularization strategies to specific applications. For example, in (Kammoun et al., 2018), the criterion is not the MSE or a distance, but maximizing the probability of detection. Given the primary goal of detection, this tailored approach yields substantially better outcomes compared to conventional regularization techniques. So, based on the results of (Couillet et al., 2019; Tiomoko et al., 2019), we propose a new regularization strategy directly related to classification algorithms. In particular, we are interested in barycenters, which are for instance central to nearest centroid and K-means algorithms. Specifically, our contributions are the followings:

- First, we improve the RMT based covariance estimator proposed in (Tiomoko et al., 2019).
- Second, the main contribution is to propose a new RMT corrected Fréchet mean of SPD matrices by exploiting the improved distance from (Couillet et al., 2019).
- Third, we adapt learning algorithms, *i.e.*, K-means and nearest centroid classifier, to our new RMT mean.
- Finally, the interest of our approach is proved on both simulated and real EEG and hyperspectral data.

To ensure reproducibility, the code for the experiments discussed is accessible at `https://github.com/AmmarMian/icml-rmt-2024`.

2. Preliminaries

2.1. Random matrix theory

Random matrix theory (RMT) is a tremendous tool when it comes to studying the statistical behaviour of random matrices when the number of features p and the number of samples n grow at the same rate toward infinity, *i.e.*, as $p, n \to \infty$, $p/n \to c > 0$. In particular, from the seminal works (Wishart, 1928; Marchenko & Pastur, 1967; Silverstein & Bai, 1995), we know that the eigenvectors and eigenvalues of the sample covariance matrix (SCM) are not consistent in the large dimensional regime. This lead researchers to regularize the SCM, more specifically its eigenvalues, in order to obtain consistent estimators; see *e.g.* (Ledoit & Wolf, 2015; 2018). Recently, with the rise of machine learning, distances between covariance matrices have attracted attention; see *e.g.* (Couillet et al., 2019; Couillet & Liao, 2022). In the same spirit as for the study of covariance matrices, it has been shown that the distances between SPD matrices are not consistent and that it is then possible to regularize them to obtain improved distances which are then consistent in the high-dimensional regime.

2.1.1. Covariance estimation

Let $\boldsymbol{X} \in \mathbb{R}^{p \times n}$ with true covariance $\boldsymbol{C}$ and SCM $\hat{\boldsymbol{C}} = \frac{1}{n}\boldsymbol{X}\boldsymbol{X}^T$. The most famous – and probably the simplest – way to regularize the SCM $\hat{\boldsymbol{C}}$ consists in the linear shrinkage: $\hat{\boldsymbol{C}}_{\text{LW}} = \rho \boldsymbol{I}_p + \sqrt{1-\rho^2}\hat{\boldsymbol{C}}$ (Ledoit & Wolf, 2004). Parameter $\rho > 0$ is chosen so that it minimizes the expected $\ell 2$ distance $\mathbb{E}[\|\boldsymbol{C} - \hat{\boldsymbol{C}}_{\text{LW}}\|_2]$ asymptotically. To estimate ρ consistently, basic results from RMT are used. However, in this setting, the eigenvalues are then biased. Another solution is to obtain a consistent estimate of the true eigenvalues $\lambda_i(\boldsymbol{C})$. A method to estimate these eigenvalues was first proposed in (El Karoui, 2008), with little success as the optimization process was very unstable. This was solved in (Ledoit & Wolf, 2015) and (Ledoit & Wolf, 2018), where the $\ell 2$ distance and a Stein loss are leveraged to estimate the $\lambda_i(\boldsymbol{C})$'s from the $\lambda_i(\hat{\boldsymbol{C}})$'s with the so-called QuEST method. The major limitation is that, even though QuEST is quite accurate, it is computationnally very expensive, which makes it complicated to employ in real scenarios. Recently, (Ledoit & Wolf, 2020) proposed an analytical non-linear shrinkage of the $\lambda_i(\hat{\boldsymbol{C}})$'s, *i.e.*, functions ϕ_i are learnt such that the $\lambda_i(\boldsymbol{C})$'s are estimated through $\phi_i(\lambda_i(\hat{\boldsymbol{C}}))$. To determine the ϕ_i's, RMT, oracle non-linear shrinkage function and kernels are exploited. They are chosen to minimize the true variance. This method features the accuracy of QuEST while being numerically very efficient.

2.1.2. Distance estimation

Covariance matrices are increasingly exploited in machine learning algorithms such as Quadratic Discriminant Analysis, Metric Learning, Nearest Centroid, *etc.* Often, in such scenarios, covariance matrices are mainly leveraged to compute some kind of distance. In this paper, we focus on distances between covariance matrices. Unfortunately, as shown in (Couillet et al., 2019), these are not consistent in the large dimensionality regime. Some efforts have thus been dedicated to finding some good estimators.

In (Couillet et al., 2019; Pereira et al., 2023), RMT corrected estimators of the squared distance between the true covariance matrices of some data are derived. They considered two different cases. In the first one, random data $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ with true covariance $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ and SCMs $\hat{\boldsymbol{C}}_1$ and $\hat{\boldsymbol{C}}_2$ are considered. A consistent estimator $\tilde{\delta}^2(\hat{\boldsymbol{C}}_1, \hat{\boldsymbol{C}}_2)$ of

the squared distance $\delta^2(\boldsymbol{C}_1, \boldsymbol{C}_2)$ are derived. In the other case, only one matrix is random. We have $\boldsymbol{X}$ with covariance $\boldsymbol{C}$ and SCM $\hat{\boldsymbol{C}}$ and a deterministic SPD matrix $\boldsymbol{R}$. A consistent estimator $\hat{\delta}^2(\boldsymbol{R}, \hat{\boldsymbol{C}})$ of the squared distance $\delta^2(\boldsymbol{R}, \boldsymbol{C})$ is provided. In both cases, estimators for a wide range of distances are obtained in closed form. The main limitation of these RMT squared distance estimators is that they are valid only when data $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are independent (respectively that $\boldsymbol{R}$ is not constructed with $\boldsymbol{X}$).

We focus on the squared Fisher distance (Skovgaard, 1984), which is, for all $\boldsymbol{C}_1$ and $\boldsymbol{C}_2 \in \mathcal{S}_p^{++}$ (manifold of SPD matrices),

$$\begin{aligned} \delta^2(\boldsymbol{C}_1, \boldsymbol{C}_2) &= \tfrac{1}{2p} \| \operatorname{logm}(\boldsymbol{C}_1^{-1}\boldsymbol{C}_2) \|_2^2 \\ &= \tfrac{1}{2p} \textstyle\sum_{i=1}^p \log^2(\lambda_i(\boldsymbol{C}_1^{-1}\boldsymbol{C}_2)), \end{aligned} \tag{1}$$

where $\operatorname{logm}(\cdot)$ denotes the matrix logarithm. In the present work, we exploit the estimator $\hat{\delta}^2$ between some random matrix and a deterministic matrix in $\mathcal{S}_p^{++}$ derived in (Couillet et al., 2019). It is provided in Theorem 2.1.

Theorem 2.1 (RMT corrected squared Fisher distance from (Couillet et al., 2019))**.** *Given $\boldsymbol{X} \in \mathbb{R}^{p \times n}$ $(p > n)$ with SCM $\hat{\boldsymbol{C}}$ and a deterministic $\boldsymbol{R} \in \mathcal{S}_p^{++}$, the RMT correction of the squared Fisher distance (1) is*

$$\begin{aligned} \hat{\delta}^2(\boldsymbol{R}, \hat{\boldsymbol{C}}) &= \frac{1}{2p} \sum_{i=1}^p \log^2(\lambda_i) + \frac{1}{p} \sum_{i=1}^p \log(\lambda_i) \\ &- (\boldsymbol{\lambda} - \boldsymbol{\zeta})^T [\frac{1}{p} \boldsymbol{Q} \boldsymbol{1}_p + \frac{1-c}{c} \boldsymbol{q}] - \frac{1-c}{2c} \log^2(1-c), \end{aligned} \tag{2}$$

where $c = p/n < 1$; $\boldsymbol{\lambda}$ and $\boldsymbol{\zeta}$ contain the eigenvalues of $\boldsymbol{R}^{-1}\hat{\boldsymbol{C}}$ and $\boldsymbol{\Lambda} - \frac{\sqrt{\boldsymbol{\lambda}}\sqrt{\boldsymbol{\lambda}}^T}{n}$, with $\boldsymbol{\Lambda} = \operatorname{diag}(\boldsymbol{\lambda})$; $\boldsymbol{q} \in \mathbb{R}^p$ such that $q_i = \frac{\log(\lambda_i)}{\lambda_i}$; and $\boldsymbol{Q} \in \mathbb{R}^{p \times p}$ is the matrix such that

$$Q_{ij} = \begin{cases} \frac{\lambda_i \log\left(\frac{\lambda_i}{\lambda_j}\right) - (\lambda_i - \lambda_j)}{(\lambda_i - \lambda_j)^2}, & i \neq j \\ \frac{1}{2\lambda_i}, & i = j \end{cases}.$$

2.2. Riemannian optimization on $\mathcal{S}_p^{++}$

Riemannian optimization (Absil et al., 2009; Boumal, 2023) provide generic methods to solve constrained optimization problems over any smooth manifold. In the present work, we are interested in optimization on the manifold of SPD matrices $\mathcal{S}_p^{++}$ and we are limiting ourselves to the Riemannian gradient descent algorithm. Let $f : \mathcal{S}_p^{++} \to \mathbb{R}$ be an objective function. The goal is to solve the optimization problem

$$\operatorname*{argmin}_{\boldsymbol{R} \in \mathcal{S}_p^{++}} \quad f(\boldsymbol{R}).$$

To do so, the differential structure of $\mathcal{S}_p^{++}$ is exploited. Since $\mathcal{S}_p^{++}$ is open in the space of symmetric matrices $\mathcal{S}_p$, the tangent space at any point $\boldsymbol{R} \in \mathcal{S}_p^{++}$ can be identified to $\mathcal{S}_p$. The next step is to equip $\mathcal{S}_p^{++}$ with a Riemannian metric. The choice that appears natural in our case is the Fisher information metric of the normal distribution, which yields (1). It is, for all $\boldsymbol{R} \in \mathcal{S}_p^{++}$, $\boldsymbol{\xi}, \boldsymbol{\eta} \in \mathcal{S}_p$,

$$\langle \boldsymbol{\xi}, \boldsymbol{\eta} \rangle_{\boldsymbol{R}} = \operatorname{tr}(\boldsymbol{R}^{-1}\boldsymbol{\xi}\boldsymbol{R}^{-1}\boldsymbol{\eta}). \tag{3}$$

It allows to define the Riemannian gradient $\nabla f(\boldsymbol{R})$ of f at $\boldsymbol{R} \in \mathcal{S}_p^{++}$ as the only matrix in $\mathcal{S}_p$ such that, for all $\boldsymbol{\xi} \in \mathcal{S}_p$,

$$\langle \nabla f(\boldsymbol{R}), \boldsymbol{\xi} \rangle_{\boldsymbol{R}} = \mathrm{d}\, f(\boldsymbol{R})[\boldsymbol{\xi}], \tag{4}$$

where $\mathrm{d}\, f(\boldsymbol{R})[\boldsymbol{\xi}]$ denotes the directional derivative of f at $\boldsymbol{R}$ in the direction $\boldsymbol{\xi}$. The Riemannian gradient provides a descent direction of the cost function f in the tangent space at $\boldsymbol{R}$. From there, we need to obtain a new point on $\mathcal{S}_p^{++}$. This is achieved by a retraction R, which maps every tangent vector at any point onto the manifold. In our opinion, the optimal retraction choice on $\mathcal{S}_p^{++}$ is the second-order approximation of the Riemannian exponential mapping (generalization of a straight line on a manifold) defined in (Jeuris et al., 2012), for all $\boldsymbol{R} \in \mathcal{S}_p^{++}$ and $\boldsymbol{\xi} \in \mathcal{S}_p$, as

$$R_{\boldsymbol{R}}(\boldsymbol{\xi}) = \boldsymbol{R} + \boldsymbol{\xi} + \frac{1}{2}\boldsymbol{\xi}\boldsymbol{R}^{-1}\boldsymbol{\xi}. \tag{5}$$

All the tools to apply the Riemannian gradient descent algorithm in order to optimize the cost function f have now been introduced. Given an initial guess $\boldsymbol{R}_0 \in \mathcal{S}_p^{++}$, the sequence of iterates $\{\boldsymbol{R}_\ell\}$ produced by the gradient descent is given through the recurrence

$$\boldsymbol{R}_{\ell+1} = R_{\boldsymbol{R}_\ell}(-t_\ell \nabla f(\boldsymbol{R}_\ell)), \tag{6}$$

where $t_\ell > 0$ is the stepsize, which can be computed through a linesearch; see *e.g.*, (Absil et al., 2009).

3. Closely related work

In this paper, we explore covariance and Fréchet mean estimation by leveraging (2). In (Tiomoko et al., 2019), the problem of estimating covariance by exploiting (2) has already been considered. Indeed, authors are interested in the optimization problem

$$\operatorname*{argmin}_{\boldsymbol{R} \in \mathcal{S}_p^{++}} \quad \hat{\delta}^2(\boldsymbol{R}, \hat{\boldsymbol{C}}), \tag{7}$$

and also consider Riemannian optimization to solve it. As previously explained, for (2) to provide an accurate approximation of $\delta^2(\boldsymbol{R}, \boldsymbol{C})$, $\boldsymbol{R}$ must be sufficiently independent from $\boldsymbol{X}$. When trying to solve (7), this is a big issue. The gradient obviously depends on $\boldsymbol{X}$. It inevitably induces some dependency between $\boldsymbol{R}$ and $\boldsymbol{X}$ along iterations. As a consequence, $\hat{\delta}^2(\boldsymbol{R}, \hat{\boldsymbol{C}})$ becomes irrelevant at some point and lead to an inappropriate solution.

Concerning covariance estimation, the big difference between their approach and ours lies in how this issue is handled. In (Tiomoko et al., 2019), since $\hat{\delta}^2(\boldsymbol{R}, \hat{\boldsymbol{C}})$ becomes negative when it is no longer informative, they considered optimizing $\boldsymbol{R} \mapsto (\hat{\delta}^2(\boldsymbol{R}, \hat{\boldsymbol{C}}))^2$. As it did not appear to be sufficient, they also limited their search to the eigenvalues of the true covariance matrix, *i.e.*, they assumed $\boldsymbol{R} = \boldsymbol{U}\boldsymbol{\Delta}\boldsymbol{U}^T$, where $\boldsymbol{U}$ contain the eigenvectors of the sample covariance $\hat{\boldsymbol{C}}$ and $\boldsymbol{\Delta}$ contain the sought eigenvalues.

In our work, we employ a very different strategy. Indeed, we choose to keep $\boldsymbol{R} \mapsto \hat{\delta}^2(\boldsymbol{R}, \hat{\boldsymbol{C}})$ as the actual cost function and our search space remains $\mathcal{S}_p^{++}$. Instead of changing these, we wisely define a new stopping criterion. Further notice that we derive the gradient in a very different way and end up with a formula that has a different form. Finally, they do not consider at all the Fréchet mean estimation problem, which is clearly the main contribution of our paper.

4. RMT improved covariance estimation

In this section, we improve the RMT based covariance estimation proposed in (Tiomoko et al., 2019). This latter is based on the squared distance estimator (2). The main stake of this section is to be able to disrupt as much as possible the dependency on $\boldsymbol{X}$ that is created along the optimization process which leads to mitigate results in (Tiomoko et al., 2019). To do so, in Section 4.1, we clean up the method proposed in (Tiomoko et al., 2019), which relies on (2). More specifically, we don't take the square of the squared distance estimator, perform optimization on $\mathcal{S}_p^{++}$, and propose a properly adapted stopping criterion. Finally, in Section 4.2, simulations are performed in order to compare the proposed approach to baseline methods and concluding remarks are provided.

4.1. Covariance estimator algorithm

We first consider the optimization problem (7), which leverages the squared distance estimator (2). In this scenario, the covariance estimator $\hat{\boldsymbol{C}}_{\text{dist}}$ is obtained by minimizing $f : \boldsymbol{R} \mapsto \hat{\delta}^2(\boldsymbol{R}, \hat{\boldsymbol{C}})$, which approximates $\boldsymbol{R} \mapsto \delta^2(\boldsymbol{R}, \boldsymbol{C})$, where $\boldsymbol{C}$ is the true covariance of $\boldsymbol{X}$. To solve the optimization problem, we resort to Riemannian optimization on $\mathcal{S}_p^{++}$ with the tools presented in Section 2.2. To be able to implement the Riemannian gradient descent, all we need is the Riemannian gradient of $f : \boldsymbol{R} \mapsto \hat{\delta}^2(\boldsymbol{R}, \hat{\boldsymbol{C}})$.

The objective f is a function of the eigenvalues of $\boldsymbol{R}^{-1}\hat{\boldsymbol{C}}$, *i.e.*, $f(\boldsymbol{R}) = g(\boldsymbol{\Lambda})$, where $\boldsymbol{\Lambda} \in \mathcal{D}_p^{++}$ (manifold of positive definite diagonal matrices) contain the eigenvalues of $\boldsymbol{R}^{-1}\hat{\boldsymbol{C}}$ (or equivalently $\boldsymbol{R}^{-1/2}\hat{\boldsymbol{C}}\boldsymbol{R}^{-1/2}$ to keep a symmetric matrix). First, in Proposition 4.1, the Riemannian gradient $\nabla f(\boldsymbol{R})$ of f in $\mathcal{S}_p^{++}$ is given as a function of the Riemannian gradient $\nabla g(\boldsymbol{\Lambda})$ of g in $\mathcal{D}_p^{++}$, also equipped with metric (3). As for $\mathcal{S}_p^{++}$, $\nabla g(\boldsymbol{\Lambda})$ is the only element of the space of diagonal matrices $\mathcal{D}_p$ such that, for all $\boldsymbol{\xi} \in \mathcal{D}_p$, $\mathrm{d}\, g(\boldsymbol{\Lambda})[\boldsymbol{\xi}] = \operatorname{tr}(\boldsymbol{\Lambda}^{-1}\nabla g(\boldsymbol{\Lambda})\boldsymbol{\Lambda}^{-1}\boldsymbol{\xi})$.

Proposition 4.1. *Let $\hat{\boldsymbol{C}} \in \mathcal{S}_p^{++}$ and $f : \mathcal{S}_p^{++} \to \mathbb{R}$ such that for all $\boldsymbol{R} \in \mathcal{S}_p^{++}$, $f(\boldsymbol{R}) = g(\boldsymbol{\Lambda})$, where $g : \mathcal{D}_p^{++} \to \mathbb{R}$ and $\boldsymbol{\Lambda}$ is obtained through the eigenvalue decomposition $\boldsymbol{R}^{-1/2}\hat{\boldsymbol{C}}\boldsymbol{R}^{-1/2} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$. It follows that*

$$\nabla f(\boldsymbol{R}) = -\boldsymbol{R}^{1/2}\boldsymbol{U}\boldsymbol{\Lambda}^{-1}\nabla g(\boldsymbol{\Lambda})\boldsymbol{U}^T\boldsymbol{R}^{1/2},$$

where $\nabla g(\boldsymbol{\Lambda})$ is the Riemannian gradient of g at $\boldsymbol{\Lambda}$ in $\mathcal{D}_p^{++}$.

Proof. See Appendix A. □

It now remains to compute the Riemannian gradient $\nabla g(\boldsymbol{\Lambda})$ of g at $\boldsymbol{\Lambda} \in \mathcal{D}_p^{++}$, where g corresponds to the RMT corrected squared Fisher distance (2). It is provided in Proposition 4.2.

Proposition 4.2. *Let $g : \mathcal{D}_p^{++} \to \mathbb{R}$ the function such that $g(\boldsymbol{\Lambda}) = \hat{\delta}^2(\boldsymbol{R}, \hat{\boldsymbol{C}})$, with $\hat{\delta}^2$ defined in (2) and $\boldsymbol{\Lambda}$ the eigenvalues of $\boldsymbol{R}^{-1/2}\hat{\boldsymbol{C}}\boldsymbol{R}^{-1/2}$. It follows that*

$$\begin{aligned}\nabla g(\boldsymbol{\Lambda}) = \frac{1}{p}[\operatorname{logm}(\boldsymbol{\Lambda})+\boldsymbol{I}_p]\boldsymbol{\Lambda}-\boldsymbol{\Lambda}^2[\boldsymbol{\Delta}+\operatorname{diag}(\boldsymbol{A}\boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^T)] \\ -\frac{1}{p}\boldsymbol{\Lambda}^2\operatorname{diag}(\boldsymbol{B}\boldsymbol{1}_p(\boldsymbol{\lambda}-\boldsymbol{\zeta})^T+\boldsymbol{1}_p(\boldsymbol{\lambda}-\boldsymbol{\zeta})^T\boldsymbol{C}) \\ -\frac{1-c}{c}(\boldsymbol{I}_p-\operatorname{logm}(\boldsymbol{\Lambda}))(\boldsymbol{\Lambda}-\operatorname{diag}(\boldsymbol{\zeta})),\end{aligned}$$

where $\boldsymbol{\zeta}$ and $\boldsymbol{V}$ are the eigenvalues and eigenvectors of $\boldsymbol{\Lambda} - \frac{\sqrt{\boldsymbol{\lambda}}\sqrt{\boldsymbol{\lambda}}^T}{n}$; $\boldsymbol{\Delta} = \operatorname{diag}(\frac{1}{p}\boldsymbol{Q}\boldsymbol{1}_p + \frac{(1-c)}{c}\boldsymbol{q})$, with $\boldsymbol{Q}$ and $\boldsymbol{q}$ defined in (2); and $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are the matrices such that

$$\boldsymbol{A}_{ij} = \begin{cases} -\frac{1}{n}\sqrt{\frac{\lambda_j}{\lambda_i}}, & i \neq j \\ 1-\frac{1}{n}, & i = j \end{cases},$$

$$\boldsymbol{B}_{ij} = \begin{cases} -\frac{(\lambda_i+\lambda_j)\log(\frac{\lambda_i}{\lambda_j})}{(\lambda_i-\lambda_j)^3} - \frac{2}{(\lambda_i-\lambda_j)^2}, & i \neq j \\ -\frac{1}{\lambda_i^2}, & i = j \end{cases},$$

$$\boldsymbol{C}_{ij} = \begin{cases} \frac{1}{\lambda_j(\lambda_i-\lambda_j)} + \frac{2\lambda_i\log(\frac{\lambda_i}{\lambda_j})}{(\lambda_i-\lambda_j)^3} - \frac{2}{(\lambda_i-\lambda_j)^2}, & i \neq j \\ \frac{1}{2\lambda_i^2}, & i = j \end{cases}.$$

Proof. See Appendix A. □

Injecting Proposition 4.2 in Proposition 4.1 yields the Riemannian gradient of f. This is all that is needed to perform the Riemannian gradient descent (6) on $\mathcal{S}_p^{++}$ in order to solve (7).

For covariance estimation, the interest of solving (7) lies in the fact that $\hat{\delta}^2(\boldsymbol{R}, \hat{\boldsymbol{C}})$ provides an accurate estimation of $\delta^2(\boldsymbol{R}, \boldsymbol{C})$. Unfortunately, $\hat{\delta}^2(\boldsymbol{R}, \hat{\boldsymbol{C}})$ does not actually

Algorithm 1 Covariance based on RMT corrected distance

Input: data $\boldsymbol{X} \in \mathbb{R}^{p\times n}$, initial guess $\boldsymbol{R}_0 \in \mathcal{S}_p^{++}$, tolerances $\alpha > 0$, $\varepsilon > 0$, maximum iterations $\ell_{\max}$
Compute SCM $\hat{\boldsymbol{C}} = \frac{1}{n}\boldsymbol{X}\boldsymbol{X}^T$
Set $\ell = 0$
repeat
  Compute gradient $\nabla f(\boldsymbol{R}_\ell)$ (Prop. 4.1 and 4.2)
  Compute stepsize t_ℓ with linesearch
  $\boldsymbol{R}_{\ell+1} = R_{\boldsymbol{R}_\ell}(-t_\ell \nabla f(\boldsymbol{R}_\ell))$, with R defined in (5)
  $\ell = \ell + 1$
until $f(\boldsymbol{R}_\ell) < -\alpha/p$ **or** $\delta^2(\boldsymbol{R}_\ell, \boldsymbol{R}_{\ell-1}) < \varepsilon$ **or** $\ell > \ell_{\max}$
Return: $\hat{\boldsymbol{C}}_{\text{dist}} = \boldsymbol{R}_\ell$

approximate $\delta^2(\boldsymbol{R}, \boldsymbol{C})$ for any $\boldsymbol{R} \in \mathcal{S}_p^{++}$. Indeed, if $\boldsymbol{R}$ is too related to $\boldsymbol{X}$ (*e.g.*, $\boldsymbol{R} = \hat{\boldsymbol{C}}$), then $\hat{\delta}^2(\boldsymbol{R}, \hat{\boldsymbol{C}})$ is no longer informative. In fact, it can even take negative values. To handle this, (Tiomoko et al., 2019) chose to rather perform optimization on the square of the RMT squared distance estimator (2). In this paper, we argue that this is not necessary and that wisely choosing the stopping criterion is enough. Indeed, starting from an adequate initialization (*i.e.*, one that is sufficiently independent from $\boldsymbol{X}$), our idea is to pursue optimization while $\hat{\delta}^2(\boldsymbol{R}, \hat{\boldsymbol{C}})$ is relevant and to stop as we reach the limit. From a statistical point of view, when $\boldsymbol{R}$ is not too related to $\boldsymbol{X}$, one expects $\hat{\delta}^2(\boldsymbol{R}, \hat{\boldsymbol{C}}) \geq O(-1/p)$. Thus, our new stopping criterion consists in checking that we have $f(\boldsymbol{R}) = \hat{\delta}^2(\boldsymbol{R}, \hat{\boldsymbol{C}}) \geq -\alpha/p$, and to stop as soon as this is no longer true. Some cross-validation on synthetic data for various p and n lead us to believe that choosing $\alpha = 10$ is the best option. The method to estimate covariance by leveraging the RMT corrected squared distance is presented in Algorithm 1[1].

Remark 4.3. Concerning initialization, we need to select one that is sufficiently unrelated to $\boldsymbol{X}$. The simplest choice is $\boldsymbol{I}_p$. The SCM $\hat{\boldsymbol{C}}$ is of course not an option. The non-linear shrinkage estimator $\hat{\boldsymbol{C}}_{\text{LW-NL}}$ from (Ledoit & Wolf, 2020) also usually does not work. Interestingly, the linear shrinkage estimator $\hat{\boldsymbol{C}}_{\text{LW}}$ (Ledoit & Wolf, 2004) appears to usually be the strongest option we considered.

4.2. Simulations summary and concluding remarks

Detailed simulations on covariance estimation are provided in Appendix B. Due to space limitations, only a summary and some concluding remarks are provided here. In our simulations, we randomly generate a covariance matrix. We then simulate some data that are used to estimate their covariance. Various methods are considered: the SCM $\hat{\boldsymbol{C}}$, the linear Ledoit-Wolf estimator $\hat{\boldsymbol{C}}_{\text{LW}}$ (Ledoit & Wolf, 2004), the non-linear Ledoit-Wolf estimator $\hat{\boldsymbol{C}}_{\text{LW-NL}}$ (Ledoit & Wolf, 2020), and our RMT distance based method $\hat{\boldsymbol{C}}_{\text{dist}}$ from Algorithm 1.

The best performance is obtained with $\hat{\boldsymbol{C}}_{\text{LW-NL}}$. Our estimator $\hat{\boldsymbol{C}}_{\text{dist}}$ improves upon $\hat{\boldsymbol{C}}$ and $\hat{\boldsymbol{C}}_{\text{LW}}$ at low sample support. Considering that it is also more expensive (others are analytical), it does not seem advantageous and exploiting (2) might not be suited for covariance estimation. Notice however that in some rare cases at low sample support, $\hat{\boldsymbol{C}}$, $\hat{\boldsymbol{C}}_{\text{LW}}$ and $\hat{\boldsymbol{C}}_{\text{LW-NL}}$ behave poorly while our estimator performs well. We believe that this occurs when the SCM does not provide good eigenvectors.

5. RMT corrected Fréchet mean on $\mathcal{S}_p^{++}$

This section contains the most interesting contribution of this paper. We propose an original RMT based method to estimate the Fréchet mean (also known as Karcher or geometric mean) $\boldsymbol{G} \in \mathcal{S}_p^{++}$ of a set of K covariance matrices $\{\boldsymbol{C}_k\}$ in $\mathcal{S}_p^{++}$ when only some data $\{\boldsymbol{X}_k\}$ in $\mathbb{R}^{p\times n}$ are known. Notice that this corresponds to the setting that is always encountered in practice when one aims to exploit one or several Fréchet means of some covariance matrices in order to perform a learning task. Usually, getting a Fréchet mean is achieved with a two steps procedure: (*i*) covariance matrices are estimated from the data and (*ii*) their mean is computed with an iterative method such as (Fletcher & Joshi, 2004; Jeuris et al., 2012). The obtained Fréchet means are then exploited for classification or clustering, for instance in Nearest Centroid or K-Means algorithms; see *e.g.*, (Tuzel et al., 2008; Barachant et al., 2011).

In this work, we rather develop a one step method that directly estimate the mean $\boldsymbol{G}$ from observations $\{\boldsymbol{X}_k\}$ without trying to obtain their covariance matrices. As for our attempt on improving covariance in Section 4, our model heavily relies on the RMT corrected squared Fisher distance (2). In Section 5.1, the optimization problem that we consider along with the algorithm proposed to solve it are presented. In Section 5.2, our RMT mean is leveraged to define original Nearest Centroid and K-Means. Finally, in Section 5.3, our method is compared with the usual two steps procedure for various covariance estimators on simulated data. Concluding remarks are also provided.

5.1. RMT mean algorithm

Let a set of K raw data matrices $\{\boldsymbol{X}_k\}$ in $\mathbb{R}^{p\times n}$ with SCMs $\{\hat{\boldsymbol{C}}_k\}$. To obtain our RMT based Fréchet $\hat{\boldsymbol{G}}_{\text{RMT}}$ on $\mathcal{S}_p^{++}$, we simply replace the squared Fisher distance δ^2 defined in (1) with its RMT corrected counterpart (2) in the definition of the Fréchet mean. It follows that $\hat{\boldsymbol{G}}_{\text{RMT}}$ is solution to the

[1] Notice that the linesearch (Absil et al., 2009; Boumal, 2023) is slightly modified. In addition to the Armijo condition, we add the condition $f(R_{\boldsymbol{R}_\ell}(-t_\ell \nabla f(\boldsymbol{R}_\ell)) \geq -\alpha/p$ to the backtracking procedure on t_ℓ.

optimization problem

$$\underset{\boldsymbol{R}\in\mathcal{S}_p^{++}}{\text{argmin}} \quad h(\boldsymbol{R}) = \frac{1}{K}\sum_{k=1}^{K}\hat{\delta}^2(\boldsymbol{R},\hat{\boldsymbol{C}}_k). \tag{8}$$

The objective function $h : \boldsymbol{R} \mapsto \frac{1}{K}\sum_k \hat{\delta}^2(\boldsymbol{R},\hat{\boldsymbol{C}}_k)$ aims to approximate the cost function one would get if the true covariance matrices $\{\boldsymbol{C}_k\}$ were known, *i.e.*, $\boldsymbol{R} \mapsto \frac{1}{K}\sum_k \delta^2(\boldsymbol{R},\boldsymbol{C}_k)$. Hence, our hope is to significantly improve the estimation of the true mean $\boldsymbol{G}$ as compared to two steps procedures that compute the Fréchet mean of some covariance estimators.

Remark 5.1. When n is large enough, the usual cost function $\boldsymbol{R} \mapsto \frac{1}{K}\sum_k \delta^2(\boldsymbol{R},\hat{\boldsymbol{C}}_k)$ well approximates $\boldsymbol{R} \mapsto \frac{1}{K}\sum_k \delta^2(\boldsymbol{R},\boldsymbol{C}_k)$ since the SCM $\hat{\boldsymbol{C}}_k$ asymptotically converges to the true covariance $\boldsymbol{C}_k$. It is no longer true for a small n. In comparison, our proposed cost function appears advantageous for a wider range of number of samples n.

As for covariance estimation from Section 4, it is crucial to determine whether our cost function is truly informative. Given k, recall that $\hat{\delta}^2(\boldsymbol{R},\hat{\boldsymbol{C}})_k$ well approximates $\delta^2(\boldsymbol{R},\boldsymbol{C}_k)$ only if $\boldsymbol{R}$ is sufficiently independent from $\boldsymbol{X}_k$. Again, while optimizing h, some dependency on $\boldsymbol{X}_k$ is introduced. However, this time, the dependency on $\boldsymbol{X}_k$ is counterbalanced by the ones on the other data matrices $\{\boldsymbol{X}_{k'}\}_{k'\neq k}$. Since data matrices are independent from one another, overall, we expect $\boldsymbol{R}$ to remain sufficiently independent from each $\boldsymbol{X}_k$ as soon as K is large enough[2].

To solve (8), we again resort to a Riemannian gradient descent on $\mathcal{S}_p^{++}$. It is thus needed to compute the gradient of h. Writing $h : \boldsymbol{R} \mapsto \frac{1}{K}\sum_k f_k(\boldsymbol{R})$, with $f_k : \boldsymbol{R} \mapsto \hat{\delta}^2(\boldsymbol{R},\hat{\boldsymbol{C}}_k)$, one has

$$\nabla h(\boldsymbol{R}) = \frac{1}{K}\sum_{k=1}^{K}\nabla f_k(\boldsymbol{R}), \tag{9}$$

where $\nabla f_k(\boldsymbol{R})$ is obtained by combining Propositions 4.1 and 4.2. With the tools of Section 2.2, it is enough to implement the Riemannian gradient descent. Our proposed method is summarized in Algorithm 2.

Remark 5.2. The complexity of an iteration of Algorithm 2 is of the same order of magnitude as an iteration of the Riemannian gradient descent for the usual Fréchet mean on $\mathcal{S}_p^{++}$ (*i.e.*, with (1)). The difference between the two lies in gradients computations. Even though (9) appears way more complicated, it is not that much more expensive. Concerning costly operations, in both cases, we have to perform a Cholesky decomposition and its inverse (to compute $\boldsymbol{R}^{1/2}$ and $\boldsymbol{R}^{-1/2}$), and K eigenvalue decompositions (of $\boldsymbol{R}^{-1/2}\hat{\boldsymbol{C}}_k\boldsymbol{R}^{-1/2}$). To get (9), we further need K eigenvalue decompositions (of $\boldsymbol{\Lambda} - \frac{\boldsymbol{\lambda}\boldsymbol{\lambda}^T}{n}$). The rest only involve less expensive operations (matrix multiplications, *etc.*).

[2]In practice, it appears true even for small values of K. Indeed, in our simulations (Section 5.3), even for $K = 2$, we improve upon the SCM associated with the usual Fréchet mean on $\mathcal{S}_p^{++}$.

Algorithm 2 RMT corrected Fréchet mean on $\mathcal{S}_p^{++}$

Input: data $\{\boldsymbol{X}_k\}_{k=1}^K$ in $\mathbb{R}^{p\times n}$, initial guess $\boldsymbol{R}_0 \in \mathcal{S}_p^{++}$, tolerance $\varepsilon > 0$, maximum iterations $\ell_{\max}$
for k **in** $[\![1, K]\!]$ **do**
  Compute SCM $\hat{\boldsymbol{C}}_k = \frac{1}{n}\boldsymbol{X}_k\boldsymbol{X}_k^T$
end for
Set $\ell = 0$
repeat
  Compute gradient $\nabla h(\boldsymbol{R}_\ell)$ with (9)
  Compute stepsize t_ℓ with linesearch
  $\boldsymbol{R}_{\ell+1} = R_{\boldsymbol{R}_\ell}(-t_\ell\nabla h(\boldsymbol{R}_\ell))$, with R defined in (5)
  $\ell = \ell + 1$
until $\delta^2(\boldsymbol{R}_\ell, \boldsymbol{R}_{\ell-1}) < \varepsilon$ **or** $\ell > \ell_{\max}$
Return: $\hat{\boldsymbol{G}}_{\text{RMT}} = \boldsymbol{R}_\ell$

5.2. Nearest Centroid and K-Means based on RMT

To exploit the RMT corrected Fréchet mean on $\mathcal{S}_p^{++}$ in learning, we adapt the acclaimed Nearest Centroid classifying and K-Means clustering methods. Both algorithms rely on the RMT Fréchet mean $\hat{\boldsymbol{G}}_{\text{RMT}}$ and on the corrected squared Fisher distance $\hat{\delta}^2$.

In the supervised Nearest Centroid setting, provided in Algorithm 3, we have a training set $\{\boldsymbol{X}_k, y_k\}_{k=1}^K$, where each $\boldsymbol{X}_k \in \mathbb{R}^{p\times n}$ belongs to a class y_k in $[\![1, Z]\!]$. In the fitting phase, the RMT Fréchet means $\{\hat{\boldsymbol{G}}_{\text{RMT}}^{(z)}\}$ of every class $z \in [\![1, Z]\!]$ are learnt by solving

$$\hat{\boldsymbol{G}}_{\text{RMT}}^{(z)} = \underset{\boldsymbol{R}\in\mathcal{S}_p^{++}}{\text{argmin}} \quad \frac{1}{K_z}\sum_{y_k\in\mathcal{A}_z}\hat{\delta}^2(\boldsymbol{R},\hat{\boldsymbol{C}}_k), \tag{10}$$

where $\mathcal{A}_z = \{y_k : k \in [\![1, K]\!] \text{ and } y_k = z\}$ and K_z is the cardinal of $\mathcal{A}_z$. They are obtained with Algorithm 2. Then, in the prediction phase, given some unlabeled data $\boldsymbol{X} \in \mathbb{R}^{p\times n}$ with SCM $\hat{\boldsymbol{C}}$, the decision rule is

$$y = \underset{z\in[\![1,Z]\!]}{\text{argmin}} \quad \{\hat{\delta}^2(\hat{\boldsymbol{G}}_{\text{RMT}}^{(z)},\hat{\boldsymbol{C}})\}_{z=1}^Z. \tag{11}$$

The Nearest Centroid classifier can be adapted to the unsupervised K-Means clustering scenario, detailed in Algorithm 4. In this setting, one has a set of K data samples $\{\boldsymbol{X}_k\}$ in $\mathbb{R}^{p\times n}$. Given a certain number of classes Z, the goal is to assign a label $y_k \in [\![1, Z]\!]$ to each $\boldsymbol{X}_k$. This is achieved iteratively. Each iteration ℓ consists of two steps. An assignment step, where, given Z means $\{\hat{\boldsymbol{G}}_{\text{RMT}}^{(z)}(\ell)\}$, a label $y_k(\ell)$ is assigned to each $\boldsymbol{X}_k$ leveraging rule (11). A mean update step, where every $\hat{\boldsymbol{G}}_{\text{RMT}}^{(z)}(\ell)$ is recomputed

Algorithm 3 Nearest Centroid classifier based on RMT

Fitting phase

Input: data $\{\boldsymbol{X}_k\}_{k=1}^K$ in $\mathbb{R}^{p\times n}$, labels $\{y_k\}_{k=1}^K$ in $[\![1, Z]\!]$
for z **in** $[\![1, Z]\!]$ **do**
  Compute $\hat{\boldsymbol{G}}_{\text{RMT}}^{(z)}$ from $\{\boldsymbol{X}_k : y_k = z\}$ with Algo. 2
end for
Return: $\{\hat{\boldsymbol{G}}_{\text{RMT}}^{(z)}\}_{z=1}^Z$

Prediction phase

Input: unlabeled data $\boldsymbol{X} \in \mathbb{R}^{p\times n}$
Compute SCM $\hat{\boldsymbol{C}} = \frac{1}{n}\boldsymbol{X}\boldsymbol{X}^T$
for z **in** $[\![1, Z]\!]$ **do**
  Compute $\hat{\delta}^2(\hat{\boldsymbol{G}}_{\text{RMT}}^{(z)}, \hat{\boldsymbol{C}})$ with (2)
end for
Compute y with (11)
Return: label $y \in [\![1, Z]\!]$

Algorithm 4 K-Means clustering based on RMT

Input: data $\{\boldsymbol{X}_k\}_{k=1}^K$ in $\mathbb{R}^{p\times n}$, number of classes Z, tolerance $\alpha > 0$, maximum iterations $\ell_{\max}$, number of different initializations M
for m **in** $[\![1, M]\!]$ **do**
  Randomly choose $\{k_z\}_{z=1}^Z$ and set $\hat{\boldsymbol{G}}_{\text{RMT}}^{(z)}(0) = \hat{\boldsymbol{C}}_{k_z}$
  Compute $\{y_k(0)\}$ with (11)
  Set $\ell = 0$
  repeat
    $\ell = \ell + 1$
    Compute $\{\hat{\boldsymbol{G}}_{\text{RMT}}^{(z)}(\ell)\}$ from $\{\boldsymbol{X}_k : y_k(\ell - 1) = z\}$ with Algo. 2
    Compute $\{y_k(\ell)\}$ with (11)
  until $\frac{1}{K}\sum_k \|y_k(\ell) - y_k(\ell-1)\|_2 < \alpha$ **or** $\ell > \ell_{\max}$
  Compute inertia $\mathcal{I}(m)$ for initialization m with (12)
end for
Compute $m_{\max} = \text{argmax}_{m\in[\![1,M]\!]} \{\mathcal{I}(m)\}_{m=1}^M$.
Return: $\{y_k\}_{k=1}^K$ associated with $m_{\max}$.

from $\{\boldsymbol{X}_k : y_k(\ell) = z\}$. This is repeated until we reach some equilibrium. It is well known that the results of this procedure are very sensitive to the initialization of centroids. As prescribed in (Arthur & Vassilvitskii, 2007) in the Euclidean case, we consider using several initializations and keep results from the one maximizing the criterion

$$\mathcal{I}(\{\mathbf{X}_k, y_k\}, \{\hat{\boldsymbol{G}}_{\text{RMT}}^{(z)}\}) = \sum_{k=1}^K \hat{\delta}^2(\hat{\boldsymbol{G}}_{\text{RMT}}^{(y_i)}, \boldsymbol{X}_k). \qquad (12)$$

5.3. Simulations

This section contains simulations conducted to evaluate the performance of our proposed RMT based method as compared to state-of-the-art algorithms. The experimental setup is as follows. A center $\boldsymbol{G} = \boldsymbol{U}\boldsymbol{\Delta}\boldsymbol{U}^T \in \mathcal{S}_p^{++}$ is generated, where $\boldsymbol{U}$ is uniformly drawn on $\mathcal{O}_p$ (orthogonal group), and $\boldsymbol{\Delta}$ is randomly drawn on $\mathcal{D}_p^{++}$. Maximal and minimal diagonal entries of $\boldsymbol{\Delta}$ are set to $\sqrt{a}$ and $1/\sqrt{a}$, where $a = 100$ is the condition number. Remaining non-zero elements are uniformly drawn in-between. Then, K matrices $\{\boldsymbol{C}_k\}$ whose Fréchet mean is $\boldsymbol{G}$ are randomly generated. To do so, given k, $\frac{p(p+1)}{2}$ values are drawn from $\mathcal{N}(0, \sigma^2)$, with $\sigma^2 = 0.1$. These are used to canonically construct $\boldsymbol{S}_k \in \mathcal{S}_p$. A set of K centered symmetric matrices $\{\boldsymbol{\xi}_k\}$ is obtained by canceling the mean of the $\boldsymbol{S}_k$'s, *i.e.*, $\boldsymbol{\xi}_k = \boldsymbol{S}_k - \frac{1}{K}\sum_{k'} \boldsymbol{S}_{k'}$. Hence, $\frac{1}{K}\sum_k \boldsymbol{\xi}_k = \mathbf{0}$. Finally, $\boldsymbol{C}_k = \boldsymbol{G}^{1/2}\,\text{expm}(\boldsymbol{\xi}_k)\boldsymbol{G}^{1/2}$, where $\text{expm}(\cdot)$ denotes the matrix exponential. After that, we generate K matrices $\boldsymbol{X}_k$ in $\mathbb{R}^{p\times n}$ such that each column of $\boldsymbol{X}_k$ is drawn from $\mathcal{N}(\mathbf{0}, \boldsymbol{C}_k)$.

To estimate $\boldsymbol{G}$ from $\{\boldsymbol{X}_k\}$, several methods are considered. First, two steps methods are employed. They consist in estimating covariance matrices and then their usual Fréchet mean. The mean resulting from the SCM estimator is denoted $\hat{\boldsymbol{G}}_{\text{SCM}}$. The ones obtained after employing the linear and non-linear Ledoit-Wolf estimators are denoted $\hat{\boldsymbol{G}}_{\text{LW}}$ and $\hat{\boldsymbol{G}}_{\text{LW-NL}}$, respectively. These are compared to our proposed RMT based mean $\hat{\boldsymbol{G}}_{\text{RMT}}$ obtained with Algorithm 2. To measure performance, we use the squared Fisher distance (1) between the true mean and its estimator.

Results are presented in Figure 1. These indicate a distinct advantage of our proposed RMT-based method over the others across all examined scenarios. One can observe that when the number of samples n grows, $\hat{\boldsymbol{G}}_{\text{SCM}}$ and $\hat{\boldsymbol{G}}_{\text{LW-NL}}$ slowly catch up with $\hat{\boldsymbol{G}}_{\text{RMT}}$. When n is fixed (moderately low) and the number of matrices K increases, $\hat{\boldsymbol{G}}_{\text{RMT}}$ is the only one that strongly improves. In conclusion, our RMT-based method demonstrates superior performance, especially when the sample size n is moderately limited and the number of matrices K is large.

6. Real data learning experiments

To assess the practical relevance of the proposed RMT-based mean estimation method, two real-world scenarios are considered: (*i*) electroencephalography (EEG) classification using Nearest Centroid classifiers and (*ii*) clustering of hyperspectral images using K-Means algorithms. Various strategies were implemented for mean computation and distance. Specifically, for mean estimation, we consider two step strategies where we first estimate covariances and then compute their generic Fréchet mean associated with (1). As before, we consider the SCM, linear Ledoit-Wolf (LW) and non-linear Ledoit-Wolf (LW-NL) estimators. These methods were then benchmarked against our proposed RMT-

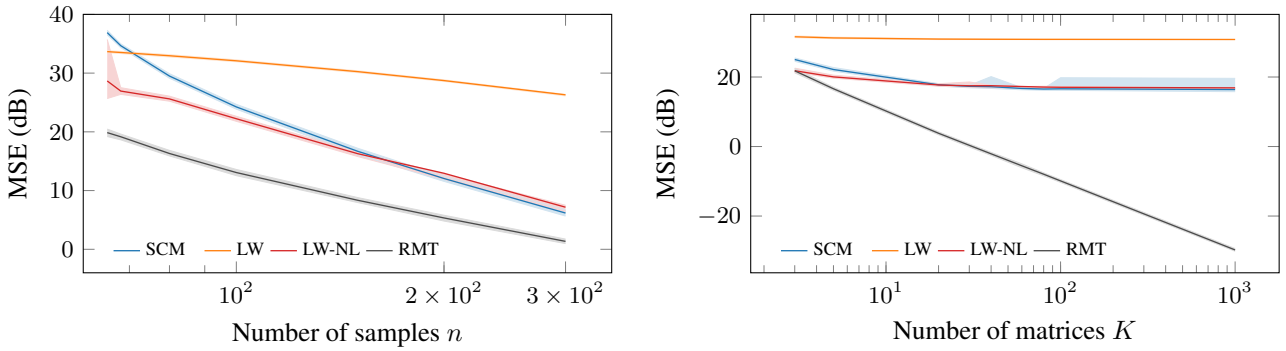

Figure 1. Mean square error (MSE) over 1000 trials of the estimated Fréchet mean towards the true mean matrix with respect to the number of samples n (left) and number of matrices K (right). Parameters are $p = 64$, $K = 10$ on the left and $n = 128$ on the right. Lines correspond to the medians while filled areas correspond to the 5th and 95th quantiles.

	SCM	LW	LW-NL	RMT
GrosseWentrup09	0.632 ± 0.0867	0.624 ± 0.0829	$\times$	$\mathbf{0.638 \pm 0.0917}$
Schirmeister17	0.597 ± 0.139	0.483 ± 0.0958	0.561 ± 0.120	$\mathbf{0.603 \pm 0.120}$
Cho17	0.615 ± 0.158	0.609 ± 0.136	0.601 ± 0.131	$\mathbf{0.622 \pm 0.158}$
Lee19	$\mathbf{0.666 \pm 0.138}$	0.642 ± 0.130	0.626 ± 0.126	0.66 ± 0.137

Table 1. Classification results on EEG motor imaging data.

based Nearest Centroid and K-Means algorithms, as detailed in Section 5.2. The development and evaluation of these methods were conducted in Python. Specifically, SCM and LW implementations were sourced from the scikit-learn library (Pedregosa et al., 2018), while LW-NL comes from scikit-RMT[3]. The conventional Fréchet means, standard Nearest Centroid and K-Means algorithms were taken from the pyRiemann library (Barachant et al., 2023).

6.1. EEG data

We initiated our analysis by assessing the Nearest Centroid classifier's efficacy on EEG data, specifically focusing on motor imagery datasets accessible via the MOABB platform (Aristimunha et al., 2023). In this context, subjects participate in experiments where they are instructed to mentally simulate various movements, encompassing actions like the motion of the left or right hand, feet, tongue, among others. The following datasets are used: GrosseWentrup2009, where $Z = 2$, $p = 128$, signals resampled to 100Hz; Schirmeister2017, where $Z = 4$, $p = 128$, signals resampled to 100Hz; Cho2017, where $Z = 2$, $p = 64$, signals resampled to 128Hz, trials taken from 1s to 3s; Lee2019, where $Z = 2$, $p = 62$, signals resampled to 100Hz, trials taken from 2s to 3s.

The outcomes are summarized in Table 1. An analysis of the results reveals that the SCM and RMT methods demonstrate comparable levels of performance across all datasets, with RMT achieving marginal enhancements in three out of the four datasets. Conversely, the accuracy rates for both LW and LW-NL are notably lower. Specifically, in the case of GrosseWentrup2009, the LW-NL method encountered issues, failing to produce SPD matrices as required, rendering it non-functional. This observation underscores the superior reliability of the RMT method as a regularization technique for these datasets. However, given the minimal performance gap between RMT and SCM, the incremental benefit of RMT may not justify the additional complexity for this particular application.

6.2. Hyperspectral data

Our second experiment with real data delves into the clustering of hyperspectral remote sensing datasets, including Indian Pines, Salinas, Pavia, and KSC[4]. These datasets are inherently diverse, characterized by a unique number of bands and classes. They also feature annotated ground truths. Certain zones labeled as "undefined" are considered unreliable and hence are omitted from the accuracy calculations of the clustering methods. Nevertheless, these zones are included during the clustering phase to ensure realistic evaluation.

Data preprocessing involves three main steps: normalizing data by subtracting the image's global mean, employing Principal Component Analysis (PCA) to select a set number of channels p as per prior research (Collas et al., 2021),

[3] https://scikit-rmt.readthedocs.io/

[4] Available at https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes.

			SCM		LW		RMT	
	p	n	acc	mIoU	acc	mIoU	acc	mIoU
Indian pines	5	5×5	0.385	0.278	0.302	0.204	**0.454**	**0.367**
	16	5×5	0.357	0.229	0.316	0.215	**0.413**	**0.284**
	24	7×7	0.377	0.253	0.359	0.248	**0.453**	**0.285**
Salinas	5	5×5	0.542	0.382	0.402	0.252	**0.777**	**0.631**
	10	7×7	0.525	0.34	0.449	0.303	**0.746**	**0.532**
	16	11×11	0.497	0.317	0.404	0.244	**0.632**	**0.461**
Pavia	5	5×5	0.629	0.378	0.615	0.319	**0.819**	**0.549**
KSC	5	5×5	0.263	0.167	0.247	0.169	**0.377**	**0.222**

Table 2. Clustering results for hyperspectral data. For Indian pines, we did 10 initializations and 5 for the other datasets.

and using a sliding window with overlap for data sampling around each pixel. We excluded the LW-NL method due to numerical instability. The K-means algorithm, capped at 100 iterations with early stopping at a 10^{-4} tolerance, concludes with a linear assignment optimization to align the clustered image with ground truth, optimizing classification accuracy.

The results, detailed in Table 2, evaluate classification accuracy and mean intersection over union. Our RMT method consistently outperforms SCM and LW across all datasets and varying feature/sample sizes. In our opinion, this success can be attributed to the high number of matrices per class in these datasets, resonating with our simulation insights: in such contexts, the RMT-corrected mean significantly enhances accuracy. In essence, for data scenarios with extensive matrices per class, the RMT approach proves highly effective.

6.3. Discussion

The first thing that we notice on these real data learning experiments is the is the lower performance of the methods associated with the linear and non-linear Ledoit-Wolf estimators. For LW, this is in accordance with our simulations. It is likely that this is because the true covariance matrices are in fact quite far from the identity matrix. So, linearly shrinking toward the identity does not improve estimation in this case. For LW-NL, this is more surprising. On these real data, LW-NL has proved itself numerically unstable and unable to consistently provide SPD matrices.

Concerning our RMT-based method, we observe that:

- When the number of samples n is sufficiently big as compared to the dimension p, our proposed RMT mean becomes equivalent to the usual Fisher mean. In this case, one can expect to obtain very similar results with our method as compared to the method associated with the SCM. This is indeed what we observe with EEG data, for which a rather large n is available (and needed to capture the information contained in the data).
- When n is comparable to p, then our performance depends on the number of matrices K available to compute the mean:
  - when K is very small, the issue is that there is a clear statistical dependence introduced between the estimated mean and the data at hand. In this case, the RMT distance estimator can become irrelevant and our mean is not very satisfying. This is the worst case scenario for our method and its main limitation. However, in our simulations, a small number of matrices K is enough for our method to be competitive.
  - when K becomes big, this is where our estimator performs very well. In fact, on simulated data, our RMT estimator appears to converge to the true mean as K grows while other methods do not. For the real hyperspectral data that we consider, we are exactly in this scenario and the performance we obtain corresponds to our simulations, *i.e.*, our method performs really well as compared to other methods.

7. Conclusions and perspectives

The first part of this paper presents a refined regularized covariance estimator, building upon the corrected squared distance outlined in (Couillet et al., 2019). While this work aligns closely with (Tiomoko et al., 2019), it introduces subtle yet noteworthy enhancements, including a more comprehensive treatment of matrix independence and a new stopping criterion rooted in statistical principles. The primary contribution, however, is the development of a novel Fréchet mean algorithm tailored for random matrices under conditions of low sample support, utilizing a Riemannian gradient on the SPD matrix manifold. When applied to Nearest Centroid classifiers and K-means clustering, this new method demonstrates great potential. It appears very advantageous when dealing with a large number of matrices per class, offering a big improvement over traditional methods in this case.

Acknowledgement

This research was supported by DATAIA Convergence Institute as part of the "Programme d'Investissement d'Avenir", (ANR-17-CONV-0003) operated by Laboratoire des Signaux et Systèmes.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Absil, P.-A., Mahony, R., and Sepulchre, R. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

Aristimunha, B., Carrara, I., Guetschel, P., Sedlar, S., Rodrigues, P., Sosulski, J., Narayanan, D., Bjareholt, E., Quentin, B., Schirrmeister, R. T., Kalunga, E., Darmet, L., Gregoire, C., Abdul Hussain, A., Gatti, R., Goncharenko, V., Thielen, J., Moreau, T., Roy, Y., Jayaram, V., Barachant, A., and Chevallier, S. Mother of all BCI Benchmarks, 2023. URL https://github.com/NeuroTechX/moabb.

Arthur, D. and Vassilvitskii, S. K-means++: The advantages of careful seeding. In *Soda*, volume 7, pp. 1027–1035, 2007.

Barachant, A., Bonnet, S., Congedo, M., and Jutten, C. Multiclass brain–computer interface classification by Riemannian geometry. *IEEE Transactions on Biomedical Engineering*, 59(4):920–928, 2011.

Barachant, A., Barthélemy, Q., Gramfort, A., KING, J.-R., Rodrigues, P. L. C., Dave, Olivetti, E., Goncharenko, V., maxdolle, vom Berg, G. W., G.Reguig, Yamamoto, M. S., Artim436, Beasley, B., Bjäreholt, E., Clisson, P., Höchenberger, R., jliersch, Sassenhagen, J., mccorsi, mhurte, stevenmortier, and stonebig. pyriemann/pyriemann: v0.5, June 2023. URL https://doi.org/10.5281/zenodo.8059038.

Bhatia, R. *Positive Definite Matrices*. Princeton University Press, USA, 2015. ISBN 0691168253.

Boumal, N. *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023.

Brooks, D., Schwander, O., Barbaresco, F., Schneider, J.-Y., and Cord, M. Riemannian batch normalization for SPD neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Collas, A., Bouchard, F., Breloy, A., Ginolhac, G., Ren, C., and Ovarlez, J.-P. Probabilistic pca from heteroscedastic signals: Geometric framework and application to clustering. *IEEE Transactions on Signal Processing*, 69: 6546–6560, 2021. doi: 10.1109/TSP.2021.3130997.

Couillet, R. and Liao, Z. *Random matrix methods for machine learning*. Cambridge University Press, 2022.

Couillet, R., Tiomoko, M., Zozor, S., and Moisan, E. Random matrix-improved estimation of covariance matrix distances. *Journal of Multivariate Analysis*, 174:104531, 2019.

El Karoui, N. Spectrum estimation for large dimensional covariance matrices using random matrix theory. *Annals of Statistics*, 36(6):2757–2790, December 2008.

Fletcher, P. T. and Joshi, S. Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors. In *International Workshop on Mathematical Methods in Medical and Biomedical Image Analysis*, pp. 87–98. Springer, 2004.

Han, A., Mishra, B., Jawanpuria, P. K., and Gao, J. On riemannian optimization over positive definite matrices with the bures-wasserstein geometry. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 8940–8953. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/4b04b0dcd2ade339a3d7ce13252a29d4-Paper.pdf.

Harandi, M., Salzmann, M., and Hartley, R. Joint dimensionality reduction and metric learning: A geometric take. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1404–1413. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/harandi17a.html.

Huang, Z. and Van Gool, L. A riemannian network for spd matrix learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Jeuris, B., Vandebril, R., and Vandereycken, B. A survey and comparison of contemporary algorithms for computing the matrix geometric mean. *Electronic Transactions on Numerical Analysis*, 39:379–402, 2012.

Kammoun, A., Couillet, R., Pascal, F., and Alouini, M.-S. Optimal design of the adaptive normalized matched filter detector using regularized tyler estimators. *IEEE Transactions on Aerospace and Electronic Systems*, 54 (2):755–769, 2018. doi: 10.1109/TAES.2017.2766538.

Kobler, R. J., ichiro Hirayama, J., Zhao, Q., and Kawanabe, M. Spd domain-specific batch normalization to crack interpretable unsupervised domain adaptation in eeg. In *Neurips*, 2022.

Ledoit, O. and Wolf, M. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of multivariate analysis*, 88(2):365–411, 2004.

Ledoit, O. and Wolf, M. Spectrum estimation: A unified framework for covariance matrix estimation and PCA in large dimensions. *Journal of Multivariate Analysis*, 139: 360–384, 2015.

Ledoit, O. and Wolf, M. Optimal estimation of a large-dimensional covariance matrix under Stein's loss. *Bernoulli*, 24(4B):3791–3832, 2018.

Ledoit, O. and Wolf, M. Analytical nonlinear shrinkage of large-dimensional covariance matrices. *The Annals of Statistics*, 48(5):3043 – 3065, 2020.

Lou, A., Katsman, I., Jiang, Q., Belongie, S., Lim, S.-N., and De Sa, C. Differentiating through the fréchet mean. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Marchenko, V. A. and Pastur, L. A. Distribution of eigenvalues for some sets of random matrices. *Matematicheskii Sbornik*, 114(4):507–536, 1967.

Ollila, E. and Tyler, D. E. Regularized m-estimators of scatter matrix. *IEEE Transactions on Signal Processing*, 62(22):6059–6070, 2014.

Pascal, F., Chitour, Y., and Quek, Y. Generalized robust shrinkage estimator and its application to stap detection problem. *IEEE Transactions on Signal Processing*, 62 (21):5640–5651, 2014. doi: 10.1109/TSP.2014.2355779.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Édouard Duchesnay. Scikit-learn: Machine learning in python, 2018.

Pereira, R., Mestre, X., and Gregoratti, D. Consistent estimators of a new class of covariance matrix distances in the large dimensional regime. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Reimherr, M. L., Bharath, K., and Soto, C. Differential privacy over riemannian manifolds. In *Neural Information Processing Systems*, 2021.

Rußwurm, M., Pelletier, C., Zollner, M., Lefèvre, S., and Körner, M. Breizhcrops: A time series dataset for crop type mapping. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences ISPRS (2020)*, 2020.

Silverstein, J. W. and Bai, Z. On the empirical distribution of eigenvalues of a class of large dimensional random matrices. *Journal of Multivariate analysis*, 54(2):175–192, 1995.

Skovgaard, L. T. A Riemannian geometry of the multivariate normal model. *Scandinavian journal of statistics*, pp. 211–223, 1984.

Tiomoko, M., Couillet, R., Bouchard, F., and Ginolhac, G. Random matrix improved covariance estimation for a large class of metrics. In *International Conference on Machine Learning*, pp. 6254–6263. PMLR, 2019.

Tuzel, O., Porikli, F., and Meer, P. Pedestrian detection via classification on Riemannian manifolds. *IEEE transactions on pattern analysis and machine intelligence*, 30 (10):1713–1727, 2008.

Utpala, S., Vepakomma, P., and Miolane, N. Differentially private fréchet mean on the manifold of symmetric positive definite (SPD) matrices with log-euclidean metric. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Wishart, J. The generalised product moment distribution in samples from a normal multivariate population. *Biometrika*, pp. 32–52, 1928.

Zadeh, P., Hosseini, R., and Sra, S. Geometric mean metric learning. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2464–2471, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/zadeh16.html.

A. Proofs of Propositions 4.1 and 4.2

A.1. Proof of Proposition 4.1

Let $f(\boldsymbol{R}) = g(\boldsymbol{\Lambda})$, where we have the eigenvalue decomposition $\boldsymbol{R}^{-1/2}\hat{\boldsymbol{C}}\boldsymbol{R}^{-1/2} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$. By definition, we have

$$\mathrm{d}\, f(\boldsymbol{R}) = \mathrm{tr}(\boldsymbol{R}^{-1}\nabla f(\boldsymbol{R})\boldsymbol{R}^{-1}\,\mathrm{d}\,\boldsymbol{R}) = \mathrm{d}\, g(\boldsymbol{\Lambda}) = \mathrm{tr}(\boldsymbol{\Lambda}^{-1}\nabla g(\boldsymbol{\Lambda})\boldsymbol{\Lambda}^{-1}\,\mathrm{d}\,\boldsymbol{\Lambda}).$$

$\mathrm{d}\,\boldsymbol{\Lambda}$ corresponds to the differential of the eigenvalues of $\boldsymbol{R}^{-1/2}\hat{\boldsymbol{C}}\boldsymbol{R}^{-1/2}$, which is known to be equal to $\mathrm{d}\,\boldsymbol{\Lambda} = \mathrm{diag}(\boldsymbol{U}^T\,\mathrm{d}(\boldsymbol{R}^{-1/2}\hat{\boldsymbol{C}}\boldsymbol{R}^{-1/2})\boldsymbol{U})$. For any matrix $\boldsymbol{N}$ and diagonal matrix $\boldsymbol{\Delta}$, we have $\mathrm{tr}(\boldsymbol{\Delta}\,\mathrm{diag}(\boldsymbol{N})) = \mathrm{tr}(\boldsymbol{\Delta}\boldsymbol{N})$. Hence, since $\boldsymbol{\Lambda}^{-1}\nabla g(\boldsymbol{\Lambda})\boldsymbol{\Lambda}^{-1}$ is diagonal, we obtain

$$\mathrm{tr}(\boldsymbol{\Lambda}^{-1}\nabla g(\boldsymbol{\Lambda})\boldsymbol{\Lambda}^{-1}\,\mathrm{diag}(\boldsymbol{U}^T\,\mathrm{d}(\boldsymbol{R}^{-1/2}\hat{\boldsymbol{C}}\boldsymbol{R}^{-1/2})\boldsymbol{U})) = \mathrm{tr}(\boldsymbol{\Lambda}^{-1}\nabla g(\boldsymbol{\Lambda})\boldsymbol{\Lambda}^{-1}\boldsymbol{U}^T\,\mathrm{d}(\boldsymbol{R}^{-1/2}\hat{\boldsymbol{C}}\boldsymbol{R}^{-1/2})\boldsymbol{U}).$$

We further have $\mathrm{d}(\boldsymbol{R}^{-1/2}\hat{\boldsymbol{C}}\boldsymbol{R}^{-1/2}) = \mathrm{d}(\boldsymbol{R}^{-1/2})\hat{\boldsymbol{C}}\boldsymbol{R}^{-1/2} + \boldsymbol{R}^{-1/2}\hat{\boldsymbol{C}}\,\mathrm{d}(\boldsymbol{R}^{-1/2})$, where $\mathrm{d}(\boldsymbol{R}^{-1/2})$ is the unique symmetric solution to the equation $\mathrm{d}(\boldsymbol{R}^{-1/2})\boldsymbol{R}^{-1/2} + \boldsymbol{R}^{-1/2}\,\mathrm{d}(\boldsymbol{R}^{-1/2}) = \mathrm{d}(\boldsymbol{R}^{-1}) = -\boldsymbol{R}^{-1}\,\mathrm{d}\,\boldsymbol{R}\boldsymbol{R}^{-1}$. Let $\boldsymbol{M} = \boldsymbol{R}^{-1/2}\hat{\boldsymbol{C}}\boldsymbol{R}^{-1/2}$ and $\boldsymbol{D} = \boldsymbol{\Lambda}^{-1}\nabla g(\boldsymbol{\Lambda})$. Since $\boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{I}_p$, we have $\boldsymbol{U}\boldsymbol{\Lambda}^{-1}\nabla g(\boldsymbol{\Lambda})\boldsymbol{\Lambda}^{-1}\boldsymbol{U}^T = \boldsymbol{M}^{-1}\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T\boldsymbol{M}^{-1}$. From there, we obtain

$$\mathrm{d}\, g(\boldsymbol{\Lambda}) = \mathrm{tr}(\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T(\mathrm{d}(\boldsymbol{R}^{-1/2})\boldsymbol{R}^{1/2} + \boldsymbol{R}^{1/2}\,\mathrm{d}(\boldsymbol{R}^{-1/2}))).$$

Leveraging $\mathrm{d}(\boldsymbol{R}^{-1/2})\boldsymbol{R}^{-1/2} + \boldsymbol{R}^{-1/2}\,\mathrm{d}(\boldsymbol{R}^{-1/2}) = -\boldsymbol{R}^{-1}\,\mathrm{d}\,\boldsymbol{R}\boldsymbol{R}^{-1}$, one gets

$$\mathrm{d}\, g(\boldsymbol{\Lambda}) = -\,\mathrm{tr}(\boldsymbol{R}^{-1/2}\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T\boldsymbol{R}^{-1/2}\,\mathrm{d}\,\boldsymbol{R}) = \mathrm{d}\, f(\boldsymbol{R}) = \mathrm{tr}(\boldsymbol{R}^{-1}\nabla f(\boldsymbol{R})\boldsymbol{R}^{-1}\,\mathrm{d}\,\boldsymbol{R}).$$

The result is finally obtained by identification.

A.2. Proof of Proposition 4.2

To get the gradient of g, the directional derivative $\mathrm{d}\, g(\boldsymbol{\Lambda})$ at $\boldsymbol{\Lambda}$ is computed. Let the eigenvalue decomposition $\boldsymbol{\Lambda} - \frac{\sqrt{\boldsymbol{\lambda}}\sqrt{\boldsymbol{\lambda}}^T}{n} = \boldsymbol{V}\,\mathrm{diag}(\boldsymbol{\zeta})\boldsymbol{V}^T$, where $\boldsymbol{\lambda} = \mathrm{diag}(\boldsymbol{\Lambda})$. The differential $\mathrm{d}\,\boldsymbol{\zeta}$ of the eigenvalues $\boldsymbol{\zeta}$ is $\mathrm{d}\,\boldsymbol{\zeta} = \mathrm{diag}(\boldsymbol{V}^T\,\mathrm{d}(\boldsymbol{\Lambda} - \frac{\sqrt{\boldsymbol{\lambda}}\sqrt{\boldsymbol{\lambda}}^T}{n})\boldsymbol{V})$. Differentiating each term of g yields

$$\mathrm{d}\, g(\boldsymbol{\Lambda}) = \frac{1}{2p}\,\mathrm{d}(\mathrm{tr}(\log^2(\boldsymbol{\Lambda}))) + \frac{1}{p}\,\mathrm{d}(\log|\boldsymbol{\Lambda}|) - (\mathrm{d}\,\boldsymbol{\lambda} - \mathrm{d}\,\boldsymbol{\zeta})^T\left(\frac{1}{p}\boldsymbol{Q}\boldsymbol{1}_p + \frac{1-c}{c}\boldsymbol{q}\right) - (\boldsymbol{\lambda} - \boldsymbol{\zeta})^T\left(\frac{1}{p}\,\mathrm{d}\,\boldsymbol{Q}\boldsymbol{1}_p + \frac{1-c}{c}\,\mathrm{d}\,\boldsymbol{q}\right).$$

By leveraging classical results, we obtain

$$\frac{1}{2p}\,\mathrm{d}(\mathrm{tr}(\log^2(\boldsymbol{\Lambda}))) + \frac{1}{p}\,\mathrm{d}(\log|\boldsymbol{\Lambda}|) = \frac{1}{p}\,\mathrm{tr}([\log(\boldsymbol{\Lambda}) + \boldsymbol{I}_p]\boldsymbol{\Lambda}^{-1}\,\mathrm{d}\,\boldsymbol{\Lambda}).$$

In the following, $\div$ corresponds to the element-wise division, $\odot$ denotes the Hadamard (element-wise) product, and $\cdot^{\odot\cdot}$ corresponds to the element-wise power function. From (2), $\boldsymbol{q} = \mathrm{diag}(\log(\boldsymbol{\Lambda})\boldsymbol{\Lambda}^{-1}) = \frac{\log\boldsymbol{\lambda}}{\boldsymbol{\lambda}}$, and we obtain $\mathrm{d}\,\boldsymbol{q} = \mathrm{diag}((\boldsymbol{I}_p - \log\boldsymbol{\Lambda})\boldsymbol{\Lambda}^{-2}\,\mathrm{d}\,\boldsymbol{\Lambda}) = \frac{\boldsymbol{1}_p - \log\boldsymbol{\lambda}}{\boldsymbol{\lambda}^{\odot 2}} \odot \mathrm{d}\,\boldsymbol{\lambda}$. One can also rewrite $\boldsymbol{Q}$ as

$$\boldsymbol{Q} = \frac{[(\boldsymbol{\lambda}\odot\log\boldsymbol{\lambda})\cdot\boldsymbol{1}_p^T - \boldsymbol{\lambda}\cdot\log\boldsymbol{\lambda}^T] - [\boldsymbol{\lambda}\cdot\boldsymbol{1}_p^T - \boldsymbol{1}_p\cdot\boldsymbol{\lambda}^T] + \boldsymbol{I}_p}{[\boldsymbol{\lambda}\cdot\boldsymbol{1}_p^T - \boldsymbol{1}_p\cdot\boldsymbol{\lambda}^T]^{\odot 2} + 2\boldsymbol{\Lambda}}.$$

Differentiating this yields

$$\begin{aligned}\mathrm{d}\,\boldsymbol{Q} = &\frac{\mathrm{d}\,\boldsymbol{\Lambda}[\log\boldsymbol{\lambda}\cdot\boldsymbol{1}_p^T - \boldsymbol{1}_p\cdot\log\boldsymbol{\lambda}^T] + [\boldsymbol{\lambda}\cdot\boldsymbol{1}_p^T - \boldsymbol{1}_p\cdot\boldsymbol{\lambda}^T]\odot(\boldsymbol{1}_p\cdot\frac{\boldsymbol{1}_p}{\boldsymbol{\lambda}}^T)\,\mathrm{d}\,\boldsymbol{\Lambda}}{[\boldsymbol{\lambda}\cdot\boldsymbol{1}_p^T - \boldsymbol{1}_p\cdot\boldsymbol{\lambda}^T]^{\odot 2} + 2\boldsymbol{\Lambda}}\\ &- \frac{(2\,\mathrm{d}\,\boldsymbol{\Lambda}(\boldsymbol{1}_p\cdot\boldsymbol{1}_p^T) + [\boldsymbol{I}_p - 2(\boldsymbol{1}_p\cdot\boldsymbol{1}_p^T)]\,\mathrm{d}\,\boldsymbol{\Lambda})\odot([(\boldsymbol{\lambda}\odot\log\boldsymbol{\lambda})\cdot\boldsymbol{1}_p^T - \boldsymbol{\lambda}\cdot\log\boldsymbol{\lambda}^T] - [\boldsymbol{\lambda}\cdot\boldsymbol{1}_p^T - \boldsymbol{1}_p\cdot\boldsymbol{\lambda}^T] + \boldsymbol{I}_p)}{[\boldsymbol{\lambda}\cdot\boldsymbol{1}_p^T - \boldsymbol{1}_p\cdot\boldsymbol{\lambda}^T]^{\odot 3} + 2\boldsymbol{\Lambda}^2},\end{aligned}$$

where we use $(\mathrm{d}\,\boldsymbol{\lambda}\odot\boldsymbol{a})\cdot\boldsymbol{b}^T = \mathrm{d}\,\boldsymbol{\Lambda}(\boldsymbol{a}\cdot\boldsymbol{b}^T)$ and $\boldsymbol{a}\cdot(\mathrm{d}\,\boldsymbol{\lambda}\odot\boldsymbol{b})^T = (\boldsymbol{a}\cdot\boldsymbol{b}^T)\,\mathrm{d}\,\boldsymbol{\Lambda}$. Notice that in the equation above, when the diagonal part of the numerator is equal to zero, then the diagonal part of the denominator can be replaced with anything

different from zero. We usually choose $\boldsymbol{I}_p$. From there, calculations allow to obtain $\mathrm{d}\,\boldsymbol{Q} = \mathrm{d}\,\boldsymbol{\Lambda}\boldsymbol{B} + \boldsymbol{C}\,\mathrm{d}\,\boldsymbol{\Lambda}$ with $\boldsymbol{B}$ and $\boldsymbol{C}$ defined in Proposition 4.2.

Further calculations yield

$$-(\mathrm{d}\,\boldsymbol{\lambda} - \mathrm{d}\,\boldsymbol{\zeta})^T \left(\frac{1}{p}\boldsymbol{Q}\mathbf{1}_p + \frac{1-c}{c}\boldsymbol{q}\right) = -\operatorname{tr}(\boldsymbol{\Delta}\,\mathrm{d}\,\boldsymbol{\Lambda}) - \operatorname{tr}(\operatorname{diag}(\boldsymbol{A}\boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{A}^T)\,\mathrm{d}\,\boldsymbol{\Lambda}),$$

where $\boldsymbol{A}$ and $\boldsymbol{\Delta}$ are defined in proposition 4.2. We also have

$$-\frac{1}{p}(\boldsymbol{\lambda} - \boldsymbol{\zeta})^T\,\mathrm{d}\,\boldsymbol{Q}\mathbf{1}_p = -\frac{1}{p}\operatorname{tr}(\operatorname{diag}(\boldsymbol{B}\mathbf{1}_p(\boldsymbol{\lambda} - \boldsymbol{\zeta})^T + \mathbf{1}_p(\boldsymbol{\lambda} - \boldsymbol{\zeta})^T\boldsymbol{C})\,\mathrm{d}\,\boldsymbol{\Lambda}),$$

and

$$-\frac{1-c}{c}(\boldsymbol{\lambda} - \boldsymbol{\zeta})^T\,\mathrm{d}\,\boldsymbol{q} = -\frac{1-c}{c}\operatorname{tr}(\boldsymbol{\Lambda}^{-2}(\boldsymbol{I}_p - \log(\boldsymbol{\Lambda}))(\boldsymbol{\Lambda} - \operatorname{diag}(\zeta))\,\mathrm{d}\,\boldsymbol{\Lambda}).$$

The result is obtained by combining all above equation and identification with $\operatorname{tr}(\boldsymbol{\Lambda}^{-2}\nabla g(\boldsymbol{\Lambda})\,\mathrm{d}\,\boldsymbol{\Lambda}) = \mathrm{d}\,g(\boldsymbol{\Lambda})$.

B. Simulations for covariance estimation of Section 4

The experimental setting is as follows: some random covariance $\boldsymbol{C} = \boldsymbol{U}\boldsymbol{\Delta}\boldsymbol{U}^T \in \mathcal{S}_p^{++}$ ($p = 64$) is generated, where $\boldsymbol{U}$ is uniformly drawn on $\mathcal{O}_p$ (orthogonal group), and $\boldsymbol{\Delta}$ is randomly drawn on $\mathcal{D}_p^{++}$. Maximal and minimal diagonal entries of $\boldsymbol{\Delta}$ are set to $\sqrt{a}$ and $1/\sqrt{a}$, where $a = 100$ is the condition number. Remaining non-zero elements are uniformly drawn in-between. From there, matrices $\boldsymbol{X} \in \mathbb{R}^{p\times n}$ are simulated. Each column vector of $\boldsymbol{X}$ is independently drawn from $\mathcal{N}(\mathbf{0}, \boldsymbol{C})$. The effect of the number of samples n is studied. We perform 1000 Monte Carlo simulations.

To estimate $\boldsymbol{C}$ from $\boldsymbol{X}$, we consider the following methods: (*i*) the SCM estimator $\hat{\boldsymbol{C}}$; (*ii*) the linear Ledoit-Wolf estimator $\hat{\boldsymbol{C}}_{\text{LW}}$ (Ledoit & Wolf, 2004) (*ii*) the non-linear Ledoit-Wolf estimator $\hat{\boldsymbol{C}}_{\text{LW-NL}}$ (Ledoit & Wolf, 2020) and (*iii*) our RMT distance based method $\hat{\boldsymbol{C}}_{\text{dist}}$ from Algorithm 1. To measure performance, we evaluate the squared Fisher distance (1) between $\boldsymbol{C}$ and the estimators.

Results are given in Figure 2. We observe that the best performance is obtained by $\hat{\boldsymbol{C}}_{\text{LW-NL}}$. Our estimator $\hat{\boldsymbol{C}}_{\text{dist}}$ improves upon $\hat{\boldsymbol{C}}$ and $\hat{\boldsymbol{C}}_{\text{LW}}$ at low sample support. From these results, it does not appear appealing. It is also computationally significantly more expensive than other estimators, which are analytically known. Thus, exploiting (2) might generally not be suited for covariance estimation. To conclude on a positive note, notice that, while conducting our simulations, we encountered some rare cases at low sample support where $\hat{\boldsymbol{C}}$, $\hat{\boldsymbol{C}}_{\text{LW}}$ and $\hat{\boldsymbol{C}}_{\text{LW-NL}}$ behave poorly (especially $\hat{\boldsymbol{C}}_{\text{LW-NL}}$), while $\hat{\boldsymbol{C}}_{\text{dist}}$ performed well. We believe that this occurs when the SCM does not provide good eigenvectors.

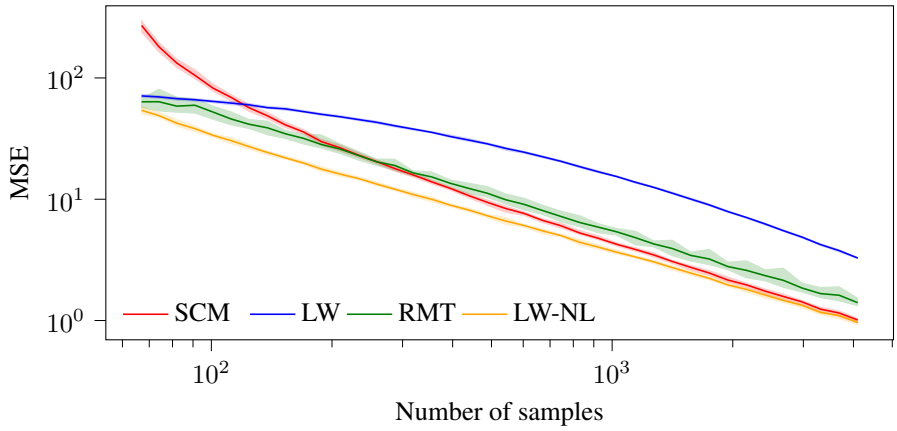

Figure 2. MSE of the estimated covariance. Parameters are $p = 64$, $\ell_{\max} = 100$, $\epsilon = 10^{-6}$, $\alpha = 10$. Plot done over 1000 trials. The line corresponds to the median and the filled area corresponds to the 5-th and 95-th quantiles over the trials.