Please cite the Published Version

Fixed Point Algorithms for Estimating Power Means of Positive Definite Matrices

Marco Congedo, Alexandre Barachant, and Ehsan Kharati Koopaei

Abstract—**Estimating means of data points lying on the Riemannian manifold of symmetric positive-definite (SPD) matrices has proved of great utility in applications requiring interpolation, extrapolation, smoothing, signal detection, and classification. The power means of SPD matrices with exponent p in the interval [–1, 1] interpolate in between the Harmonic mean (p = –1) and the Arithmetic mean (p = 1), while the Geometric (Cartan/Karcher) mean, which is the one currently employed in most applications, corresponds to their limit evaluated at 0. In this paper, we treat the problem of estimating power means along the continuum p ∈ (–1, 1) given noisy observed measurement. We provide a general fixed point algorithm (MPM) and we show that its convergence rate for p = ±0.5 deteriorates very little with the number and dimension of points given as input. Along the whole continuum, MPM is also robust with respect to the dispersion of the points on the manifold (noise), much more than the gradient descent algorithm usually employed to estimate the geometric mean. Thus, MPM is an efficient algorithm for the whole family of power means, including the geometric mean, which by MPM can be approximated with a desired precision by interpolating two solutions obtained with a small ±p value. We also present an approximated version of the MPM algorithm with very low computational complexity for the special case p = ±½. Finally, we show the appeal of power means through the classification of brain–computer interface event-related potentials data.**

Index Terms—**Brain-computer interface, geometric mean, high dimension, power means, Riemannian manifold, symmetric positive-definite matrix.**

I. INTRODUCTION

THE study of means (centers of mass) for a set of symmetric positive definite (SPD) matrices has recently attracted much attention, driven by practical problems in radar data processing, image and speech processing, computer vision, shape and movement analysis, medical imaging (especially diffusion magnetic resonance imaging and brain-computer interface), sensor networks, elasticity, numerical analysis and machine learning (e.g., [1]–[13]). In many applications the observed data can be conveniently summarized by SPD matrices, for example, some form of their covariance matrix in the time, frequency or time-frequency domain, or autocorrelation matrices. In others, SPD matrices arise naturally as kernels, tensors (or slice of), density matrices, elements of a search space, etc. Averaging such SPD matrices is a ubiquitous task. In signal processing we find it in a wide variety of data-driven algorithms allowing spatial filters, blind source separation, beamformers and inverse solutions. While robust estimation of covariance matrices and related quantities is a long-standing topic of research, only recently an information/differential geometry perspective has been considered [14]–[22].

Once observations are represented as SPD matrices, they may be treated as points on a smooth Riemannian manifold in which the fundamental geometrical notion of *distance* between two points and the *center of mass* among a number of points are naturally defined [14]. In turn, these notions allow useful operations such as interpolation, smoothing, filtering, approximation, averaging, signal detection and classification. In classification problems a simple Riemannian classifier based on a minimum distance to mean (MDM) procedure [3] has been tested with success on electroencephalographic data, in several kinds of brain-computer interfaces [3]–[6] and in the analysis of sleep stages [10], [11], as well as on motion capture data for the classification of body movements [13]. A similar method has been used for clustering in the context of video-based face and scene recognition [7] and in radar detection [2]. These examples demonstrate that simple machine learning algorithms, which are known to allow poor performance using the Euclidean metric, can be easily translated into equivalent Riemannian classifiers using an appropriate metric, obtaining excellent performance.

Among the several means one may define from an information geometry point of view, so far the geometric mean (sometimes referred to as Karcher, Cartan or Fréchet mean) has been the most studied and the most used in practical applications. It is the natural definition of mean when the Fisher-Rao metric is applied to multivariate Gaussian distributions [20], [21], but also arises naturally from a pure geometrical and algebraic perspective without making assumptions on the data distribution [14]. It so happens that the geometric mean satisfies a number of desirable invariances, including congruence invariance, self-duality, joint homogeneity and the determinant identity [23]. The simultaneous verification of all these properties is hard to

M. Congedo is with the Grenoble Images Parole Signal Automatique Laboratory, Centre National de la Recherche Scientifique, University of Grenoble Alpes and Grenoble Institute of Technology, Grenoble 38000, France (e-mail: marco.congedo@gmail.com).

A. Barachant is with the Burke Medical Research Institute, White Plains, NY 10605 USA (e-mail: e-alexandre.barachant@gmail.com).

E. K. Koopaei is with the Department of Statistical Science, University of Padova, Padova 35122, Italy (e-mail: eh.kh.ko@gmail.com).

find for means based on other metrics, such as the arithmetic, harmonic and log-Euclidean mean, thus the geometric mean of SPD matrices is not just important in practice, but a fundamental mathematical object per se.

For positive numbers the arithmetic, geometric and harmonic mean are all members of the family of power means, also known as Hölder or generalized mean. Given a set of K positive numbers $\{x_1, \dots x_K\}$ and K associated weights $\{w_1, \dots, w_K\}$ satisfying $\Sigma_k w_k = 1$, the w-weighted power mean of order p, g, of $\{x_1, \dots x_K\}$ is

$$g = \left(\sum_k w_k x_k^p \right)^{1/p}. \qquad (1)$$

Power means interpolate continuously in between the limit $p \to -\infty$ (the minimum of the set) and the limit $p \to +\infty$ (the maximum of the set), passing by $p = -1$ (harmonic mean) and $p = 1$ (arithmetic mean), while the limit $p \to 0$ from both sides allows the geometric mean. This generality of power means is appealing from a signal processing perspective; in a typical engineering scenario the sensor measurement is affected by additive noise and varying p one can find an optimal mean depending on the signal-to-noise-ratio (SNR), as we will show.

Recently [24] extended the concept of power means of positive numbers to SPD matrices for the continuum $p \in [-1, 1]$, with the case $p = -1$ being the matrix harmonic mean, $p = 1$ the matrix arithmetic mean and the limit to zero from both sides allowing the matrix geometric mean we have discussed (see also [25]–[27]). So far power means of SPD matrices have not been applied in signal processing. Also, only a "naive" fixed-point algorithm has been proposed for their estimation [24] and its convergence behavior is unsatisfactory. In this article we report a fixed-point algorithm for computing power means of SPD matrices along the interval $p \in (-1, 1)\backslash\{0\}$. This algorithm has been recently presented in [28] and therein we have named it MPM (multiplicative power means). We then demonstrate a procedure to use MPM for approximating the geometric mean with a desired precision. By means of simulation we show that the MPM displays better convergence properties as compared to alternatives used for the geometric mean, with equal or lesser computational complexity. We also show that it offers a better estimation of the geometric mean as compared to the standard gradient descent algorithm. Then, we show the advantage of considering the whole family of power means, instead of the sole geometric mean as it is customary, in classification problems, by analyzing a data set of 38 subjects related to brain-computer interface event-related potentials. Finally, in the appendix we describe an approximation of the MPM algorithm with very low computational complexity. This approximation is applicable only for values of $p = \pm\frac{1}{2}$ and is meant for applications when the computational power and/or battery life is of importance, like in mobile devices.

A. The Manifold of Symmetric Positive-Definite Matrices

In differential geometry, a smooth manifold is a topological space that is locally similar to the Euclidean space and has a
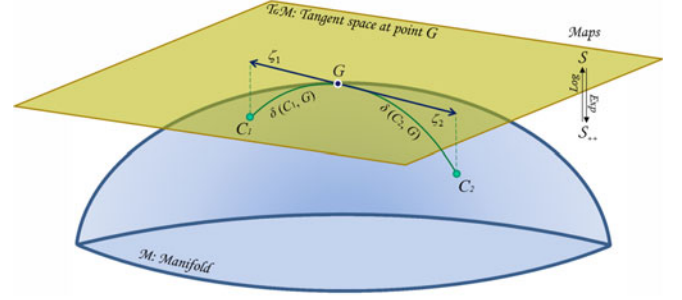

Fig. 1. Schematic representation of the SPD manifold, the geometric mean G of two points and the tangent space at G. Consider two points (e.g., two covariance matrices) C_1 and C_2 on $\mathcal{M}$. The geometric mean of these points is the midpoint on the geodesic connecting C_1 and C_2, i.e., it minimizes the sum of the two squared distances $\delta^2(C_1, G) + \delta^2(C_2, G)$. Now construct the tangent space $T_G \mathcal{M}$ at G. There exists one and only one tangent vector ζ_1 (respectively ζ_2) departing from G and arriving at the projection of C_1 (respectively C_2) from the manifold onto the tangent space; we see that the geodesics on $\mathcal{M}$ through G are transformed into straight lines in the tangent space and that therein distances are mapped logarithmically; the map from the manifold (symmetric positive definite matrices S_{++}) to the tangent space (symmetric matrices S) is of logarithmic nature. The inverse map from the tangent space to the manifold is of exponential nature. See [14] for details on these maps and [3] for their use in classification problems.

globally defined differential structure. A smooth *Riemannian manifold* $\mathcal{M}$ is equipped with an inner product on the tangent space defined at each point and varying smoothly from point to point. The tangent space $T_G \mathcal{M}$ at point G is the vector space containing the tangent vectors to all curves on $\mathcal{M}$ passing through G. For the manifold $\mathcal{M}$ of SPD matrices S_{++}, this is the space S of symmetric matrices (Fig. 1). For any two tangent vectors ζ_1 and ζ_2, we are concerned here with the inner product given by the Fisher-Rao metric at any base-point G [14]:

$$\langle \zeta_1, \zeta_2 \rangle_G = tr\left(G^{-1}\zeta_1 G^{-1}\zeta_2\right). \qquad (2)$$

B. The Geodesic

The SPD manifold has non-positive curvature and is complete; for any two points C_1 and C_2 on $\mathcal{M}$, a unique path on $\mathcal{M}$ of minimal length (at constant velocity) connecting the two points always exists. The path is named the *geodesic* and the points along it have analytical expressions given by

$$C_1 \#_t C_2 = C_1^{1/2}\left(C_1^{-1/2}C_2 C_1^{-1/2}\right)^t C_1^{1/2}, t \in [0, 1], \qquad (3)$$

with t the arc-length parameter [14]. With $t = 0$ we are at C_1, with $t = 1$ we are at C_2 and with $t = 1/2$ we are at the *geometric mean* (or center of mass) of the two points (Fig. 1). As a special case we note $I \#_t C = C^t$ and $C \#_t I = C^{1/2} C^{-t} C^{1/2} = C^{1-t}$. Geodesic equation (3) verifies $C_1 \#_t C_2 = C_2 \#_{1-t} C_1$ and $(C_1 \#_t C_2)^{-1} = C_1^{-1} \#_t C_2^{-1}$. The points along the geodesic can be understood as means of C_1 and C_2 weighted by t according to the Riemannian metric, in analogy with the weighted mean according to the Euclidean metric given by $(1 - t)C_1 + tC_2$, which still results in a SPD matrix, but greater than $C_1 \#_t C_2$ in the Loewner order sense [27].

C. The Distance

Given two matrices (points) C_1 and C_2 of dimension N·N on $\mathcal{M}$, their *Riemannian distance* is the length of the geodesic (3) connecting them (Fig. 1). It is given by [14]

$$\delta\left(C_1, C_2\right) = \left\|\text{Log}\left(C_1^{-1/2} C_2 C_1^{-1/2}\right)\right\|_F = \sqrt{\sum_{\text{n}} \log^2\lambda_{\text{n}}},$$

(4)

where $\lambda_1, \ldots, \lambda_{\text{N}}$ are the N eigenvalues of matrix $C_1^{-1/2} C_2 C_1^{-1/2}$ or of the similar matrix $C_1^{-1} C_2$. This distance has a remarkable number of properties, some of which are listed in [23]. Besides the obvious symmetry and positivity, particularly useful in signal processing are the following invariance properties, the first of which is true for any invertible matrix *B*:

$$Congruence \qquad \delta\left(BC_1B^T, \, BC_2B^T\right) = \delta\left(C_1, \, C_2\right), \quad (5)$$

$$Self-Duality \qquad \delta\left(C_1^{-1}, \, C_2^{-1}\right) = \delta\left(C_1, \, C_2\right), \quad (6)$$

D. Means of a Matrix Set: Variational Definition

Let $\boldsymbol{C} = \{C_1, \ldots, C_K\}$ be a set of K SPD matrices and $w = \{w_1, \ldots, w_K\}$ a set of associated K positive weights verifying $\Sigma_k w_k = 1$. Typically, in signal processing the elements of $\boldsymbol{C}$ are noisy data points (recordings, observations, etc.) or quantities derived thereof. Following Fréchet's variational approach, the center of mass G of set $\boldsymbol{C}$ given a distance function $d(\cdot, \cdot)$ is the point G minimizing the *variance* (*dispersion*) of points, that is, $\Sigma_k w_k d^2(G, \, C_k)$. This definition applies in general. For instance, the *w*-weighted *Arithmetic* and *Harmonic Mean* are defined, respectively, as

$$G_{\mathcal{A}}\left(w; \boldsymbol{C}\right) = \underset{G}{\arg\min} \sum_k w_k \left\|C_k - G\right\|_F^2 = \sum_k w_k C_k, \quad (7)$$

$$G_{\mathcal{H}}\left(w; \boldsymbol{C}\right) = \underset{G}{\arg\min} \sum_k w_k \left\|C_k^{-1} - G^{-1}\right\|_F^2$$

$$= \left(\sum_k w_k C_k^{-1}\right)^{-1}. \quad (8)$$

E. The Geometric Mean of a Matrix Set

In $\mathcal{M}$ the *w*-weighted *Geometric Mean* $G_{\mathcal{G}}(w; \boldsymbol{C})$ is the point realizing the minimum of $\Sigma_k w_k \delta^2(C_k, G)$, where the Riemannian distance function $\delta(\cdot, \cdot)$ acting on $\mathcal{M}$ has been defined in (4). The geometric mean is the unique point G on $\mathcal{M}$ satisfying non-linear matrix equation [18]

$$\sum_k w_k \text{Log}\left(G^{-1/2} C_k G^{-1/2}\right) = 0. \quad (9)$$

In general, it has closed-form solution only for K = 2, in which case it is indeed equal to $C_1 \#_{1/2} C_2$ (indicated shortly as $C_1 \# C_2$)-see (3) and Fig. 1 - furthermore, for K = 2 it is the unique solution to Riccati equation $(C_1 \# C_2) C_2^{-1} (C_1 \# C_2) = C_1$ [2] and is also equal to $B^{-1} D_1^{1/2} D_2^{1/2} B^{-T}$ for whatever joint diagonalizer B of C_1 and C_2, i.e., for whatever matrix B satisfying $BC_1B^T = D_1$ and $BC_2B^T = D_2$, with D_1, D_2 invertible diagonal matrices [23]. The geometric mean enjoys all

10 properties of means postulated in the seminal work [29]. It also enjoys the congruence invariance and self-duality, inherited directly from the corresponding properties (5) and (6) of its associated distance function:

$$G_{\mathcal{G}}\left(w; \left\{BC_1B^T, \ldots, BC_KB^T\right\}\right)$$

$$= BG_{\mathcal{G}}\left(w; \left\{C_1, \ldots, C_K\right\}\right) B^T, \quad (10)$$

$$G_{\mathcal{G}}^{-1}\left(w; \left\{C_1^{-1}, \ldots, C_K^{-1}\right\}\right) = G_{\mathcal{G}}\left(w; \left\{C_1, \ldots, C_K\right\}\right). \quad (11)$$

F. Power Means

Given again $\boldsymbol{C} = \{C_1, \ldots, C_K\}$ and $w = \{w_1, \ldots, w_K\}$ with $\Sigma_k w_k = 1$, we can generalize to SPD matrices the power mean of real numbers in the continuum $\text{p} \in [-1, 1]$ by the one-parameter family of matrix power means $G_{\mathcal{P}}(w; \boldsymbol{C}; \text{p})$ given by the unique SPD solution G of non-linear matrix equation [24]–[26]

$$G = \sum_k \left(w_k G \#_{\text{p}} C_k\right), \quad (12)$$

where for any pair (G, C_k) in $\mathcal{M}$, $G \#_{\text{p}} C_k$ with $\text{p} \in [0, 1]$ is the mean of G and C_k weighted by p (3). Since $G \#_{\text{p}} C_k = C_k \#_{1-\text{p}} G$ (see Section I.B) we see that a power mean is the arithmetic mean of the input matrices dragged along the geodesic toward the desired mean by an arc-length equal to $1 - \text{p}$. When the input matrices C_k all pair-wise commute, it has been proved in [24] (Property 1, p. 1502) that their power mean is

$$G = \left(\sum_k w_k C_k^p\right)^{1/\text{p}}, \quad (13)$$

which is the straightforward extension of (1) to SPD matrices. As usual, such straightforward extensions work well in commuting algebra, but not in general, thus a general solution to (12) must be found by iterative algorithms. In the sequel, we will be using the following definition of power means G_{p} covering the whole interval $\text{p} \in [-1, 1]$:

$$\begin{cases} G_{\mathcal{P}}(w; \boldsymbol{C}; \text{p} = 1) & = G_{\mathcal{A}}\left(w; \boldsymbol{C}\right), \\ G_{\mathcal{P}}(w; \boldsymbol{C}; \text{p} \in (0, 1)) & = \sum_k w_k \left(G_{\mathcal{P}} \#_{\text{p}} C_k\right), \\ G_{\mathcal{P}}(w; \boldsymbol{C}; \text{p} = 0) & = G_{\mathcal{G}}\left(w, \boldsymbol{C}\right), \\ G_{\mathcal{P}}(w; \boldsymbol{C}; \text{p} \in (-1, 0)) & = G_{\mathcal{P}}^{-1}\left(w; \boldsymbol{C}^{-1}; -\text{p}\right), \\ G_{\mathcal{P}}(w; \boldsymbol{C}; \text{p} = -1) & = G_{\mathcal{H}}\left(w; \boldsymbol{C}\right) \end{cases} \quad (14)$$

where $\boldsymbol{C}^{-1} = \{C_1^{-1}, \ldots, C_K^{-1}, \}$ $G_{\mathcal{G}}(w; \boldsymbol{C})$ is the geometric mean of Section II.E and $G_{\mathcal{A}}(w; \boldsymbol{C})$, $G_{\mathcal{H}}(w; \boldsymbol{C})$ are the arithmetic mean (7) and the harmonic mean (8), respectively. $G_{\mathcal{P}}(w; \boldsymbol{C}; \text{p})$ is named the *w-weighted power mean of order p* [24]–[26]. As per (14), the pair of power means obtained at opposite values of p around zero are the *dual* of each other; for a negative value of p the mean is defined as the inverse of the mean for –p as applied on the inverted input matrices $\boldsymbol{C}^{-1}$. As for positive numbers, power means of SPD matrices so defined interpolate continuously in between the harmonic mean (p = −1), the geometric mean (p = 0) and the arithmetic mean

(p = 1). Thus, the power means family encompasses and generalizes all Pythagorean means we have encountered so far. All of them enjoy the congruence invariance as the geometric mean does (10), but their duality, expressed in the fourth line of (14), coincides with the self-duality property (11) only for p = 0. The numerous properties of the power means can be found in [24] and a recent extension of this already quite general mathematical object has been proposed in [27].

II. ALGORITHMS FOR POWER MEANS

A. Motivation

We sought a general algorithm for computing the w-weighted power mean of order p, with $p \in (-1, 1) \backslash \{0\}$. We are also interested in an effective algorithm for estimating the geometric mean, the third line in (14). The most popular algorithm for computing the geometric mean is a Riemannian gradient descent flow with fixed step size [30]–[31]. The convergence rate of this algorithm deteriorates rapidly as the SNR decreases (high dispersion of points on the manifold). The same is true for the method based on approximate joint diagonalization in [23]. Second order methods have complexity growing very fast with the size of the input matrices, thus they are little useful in practical applications [31]. The algorithm proposed in [32] has high complexity per iteration and slow convergence rate. For a review of available algorithms for estimating the geometric mean see [23], [31]. Our algorithm does not need to make use of Riemannian geometry optimization in the manifold of SPD matrices, with consequent conceptual and computational advantage. For instance, we will be able to derive a fast approximation based exclusively on triangular matrix algebra and on the Cholesky decomposition.

B. A General Multiplicative Fixed-Point Algorithm

Hereafter it will be convenient to lighten notation; let us denote the weighted power mean of order p as P, which by (14) is equal to $G_{\mathcal{P}}(w; \boldsymbol{C}; p)$ if $p \in (0, 1)$ or to $(G_{\mathcal{P}}(w; \boldsymbol{C}^{-1}; -p))^{-1}$ if $p \in (-1, 0)$. This way we will need to handle only one expression for whatever value of $p \in (-1, 1)\{0\}$, such as

$$P^* = G_{\mathcal{P}}(w; \boldsymbol{C}^*; |p|), \qquad (15)$$

where $|p| = abs(p)$ and we define the *dual operator* $* = sgn(p)$. Definition (15) is here introduced so as to state an algorithm with identical convergence behavior for all pairs of values $\pm p$ for $|p| \in (0, 1)$. Therefore we will show results only for p positive. As initialization we take the closed form solution of the mean in the case when all matrices in set $\boldsymbol{C}$ all pair-wise commute, given by (13).
Let us now turn to the iterations. We write out (15) from definition (12) and using (3) to obtain

$$P^* = P^{*/2} \left(\sum_{k} w_{k} \left(P^{-*/2} C_{k}^* P^{-*/2} \right)^{|p|} \right) P^{*/2}. \qquad (16)$$

In [24] the authors have shown that the map defined by $f(P^*) = G_{\mathcal{P}}(w; \boldsymbol{C}^*; |p|)$ is a strict contraction for the Thompson metric with the least contraction coefficient less than or equal

to 1-|p| and as such has a unique SPD fixed point. Numerical experiments show that iterating expression (16) as it is (hereafter referred to as "naive fixed-point") results in a rather slow convergence rate, maximal for $|p| = \frac{1}{2}$, but slower and slower as |p| gets closer to 0 or to 1. In order to hasten convergence we design a multiplicative algorithm as follows: post-multiplying both sides of (16) by $P^{-*/2}$ and taking the inverse at both sides we obtain

$$P^{-*/2} = H^{-1} P^{-*/2}, \qquad (17)$$

where

$$H = \sum_{k} w_{k} \left(P^{-*/2} C_{k}^* P^{-*/2} \right)^{|p|}. \qquad (18)$$

From (16) we see that upon convergence $H = I$. H here plays the role of the origin in the SPD manifold $\mathcal{M}$ for data linearly transformed by $P^{-*/2}$. In particular, the identity matrix I is the point of symmetry in $\mathcal{M}$ corresponding to 0 in the Euclidean space due to the logarithmic map; as $P^{-1/2}$ is a whitening matrix for the arithmetic mean (p = 1), so $P^{-*/2}$ is a whitening matrix for the whole family of power means. We wish to proceed by multiplicative updates according to (17). Rather than to P^* itself, we thus seek an algorithm converging to $P^{-*/2}$, which is its inverse square root for $* = 1$, i.e., when $p \in (0, 1]$ and its square root for $* = -1$, i.e., when $p \in [-1, 0)$. The numerical stability of fixed-point iterates (17) is ensured by the fact that H converges toward I. Moreover, using our update rule any update matrix with form $H^{-\varphi}$ in (17) is equivalent to H^{-1} upon convergence. We have observed that replacing H^{-1} by $H^{-\varphi}$ in the update rule (17) does not alter the convergence to the fixed point. Nonetheless, the value of exponent φ impacts the convergence rate. In practice, using an optimal value of φ leads to a significantly faster convergence as compared to the one achieved by setting $\varphi = 1$. This holds true for power means in the whole interval $p \in (-1, 1) \backslash \{0\}$. Therefore, we will use iterate

$$P^{-*/2} = H^{-\varphi} P^{-*/2}. \qquad (19)$$

Interestingly, optimal convergence speed is observed taking φ in an interval whose extremes vary proportionally to $|p|^{-1}$. An heuristic rule that has proven adequate in intensive experiments using both real and simulated data is

$$\varphi = \tfrac{1}{2} \varepsilon^{-1} / |p|, \varepsilon \in [1, 2], \qquad (20)$$

where ε is a constant eccentricity parameter for hyperbolas (20) (Fig. 2).
The exponent $-\varphi$ in (19) acts by *retracting* the jumps of the fixed point iterations: since the fixed point is reached at $H = I$, and φ is always positive in (20), $H^{-\varphi} = H \#_{-\varphi} I = I \#_{1+\varphi} H$ (see Section I.B) is a move over the geodesic from I to H (i.e., in the direction opposite to convergence) retracting H by a distance equal to φ times the distance between I and H (here φ is the arc-length parameter of Eq. (3)). The retraction is maximal for the unit hyperbola ($\varepsilon = 1$) and minimal for $\varepsilon = 2$. By increasing ε toward 2 we obtain faster convergence in general, up to a certain value, which according to our observations mainly depends on the signal to noise ratio. In this study we take ε as 4/3 and we
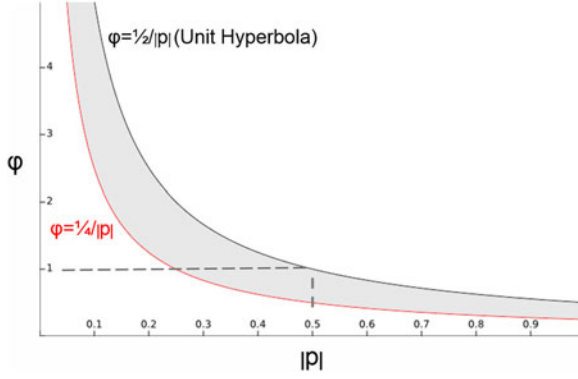
Fig. 2. The φ function of |p| (20) comprises a boomerang-shaped area enclosed by two hyperbolas: the upper limit is the unit hyperbola ($\varepsilon = 1$) and the other hyperbola obtained for $\varepsilon = 2$ is the lower limit. This area delimits an acceptable range of φ values for any given |p|.

keep it fixed in all analyses; this value has proven nearly optimal on the average of many combinations of SNR, input matrix size and dimension we have tested. The MPM algorithm in algebraic pseudo-code follows:

Algorithm: MPM (Multiplicative Power Means).

INPUT: $p \in (-1, 1) \backslash \{0\}$, K positive weights $w = w_1$, ..., w_K such that $\Sigma_k w_k = 1$ and K N·N SPD matrices $C^* = \{C_1^*, \ldots, C_K^*\}$, with $* = \text{sgn}(p)$.
OUTPUT: P, the w-weighted Power Mean of order p.
Initialize X as the principal square root inverse of (13) if $p \in (0, 1]$ or as its principal square root if $p \in [-1, 0)$.
Set ζ equal to a small floating precision number (e.g., 10^{-10})
Set $\varphi = 0.375/|p|$

REPEAT

$$H \leftarrow \sum_k \left[w_k (X C_k^* X^T)^{|p|} \right]$$

$$X \leftarrow H^{-\varphi} X$$

UNTIL $\quad \frac{1}{\sqrt{N}} \|H - I\|_F < \zeta$

RETURN $\quad P = \begin{cases} X^{-1} X^{-T} \ if \ p \in (0, 1] \\ X^T X \quad if \ p \in [-1, 0) \end{cases}$

C. Geometric Mean Approximation by Power Means

As an approximation of the geometric mean of Section I.E we consider the midpoint of geodesic (3) joining a pair of power means obtained by MPM at two small values $\pm p$ (in this article we will use $p = \pm 0.01$). Using this procedure we aim at improving current estimates of the geometric mean using the MPM algorithm.

D. A Fast Approximation

We are also interested in reducing the complexity per iteration of the MPM algorithm. This may be important when the computa-

tional complexity and/or the energy consumption are of interest, like for example in portable devices. An approximated solution is proposed for values of $p = \pm \frac{1}{2}$ and is detailed in the appendix.

III. STUDIES WITH SIMULATED DATA

A. Simulated Data Model

In many engineering applications, the matrix condition number of the SPD matrices summarizing the data (observations, recordings, . . .) tends to be positively correlated with the number of sensors. Also, the dispersion in the manifold of the matrices is proportional to the noise level. The following generative model for input data matrices $\{C_1, \ldots, C_K\}$ of size N·N is able to reproduce these properties:

$$C_k = U D_k U^T + \nu \left(V_k E_k V_k^T \right) + \alpha I, \tag{21}$$

where
- the ***signal part*** is given by $U D_k U^T$, where U is a matrix with elements drawn at random at each simulation from a uniform distribution in $[-1, 1]$ and then normalized so as to have columns with unit norm and D_k are K diagonal matrices with diagonal elements $d_{k,n}$ randomly drawn at each simulation from a chi-squared random variable divided by its degree of freedom and multiplied by $1/2^n$. So, the expectation of each element is $1/2^n$, where $n \in \{1, \ldots, N\}$ is the index of the N diagonal elements, thus forming elements of a well-known geometrical series absolutely converging to 1. The elements of the series represent the energy of N source processes, thus their sum is supposed finite (e.g., N brain dipole source processes with finite total energy).
- The ***uncorrelated noise part*** is given by αI, where I is the identity matrix and α here is taken as 10^{-6};
- The ***structured noise part*** is given by $V_k E_k V_k^T$, where the V_k matrices are generated as U above, the E_k matrices are generated as D_k above and ν is a constant controlling the SNR of the generated points (21) through

$$SNR = \frac{\text{tr} \left(\sum_k U D_k U^T \right)}{\nu \, \text{tr} \left(\sum_k \left(V_k E_k V_k^T + \alpha I \right) \right)}. \tag{22}$$

B. Simulation

In the ensuing simulations we study relevant outcome parameters as a function of the SNR, which is inversely proportional to noise level as per (22), as well as a function of the size (N) and number (K) of input matrices. We compare the gradient descent algorithm for estimating the geometric mean (GDGM: Section II.A), the naive fixed point algorithm for power means given in [24] (see (16) in Section II.B) and the MPM algorithm here presented, the latter for several values of p. In comparing the convergence rate of several algorithms the stopping criterion should be chosen identical for all of them: the relative error of matrix P with respect to a reference matrix P_{ref} is a dimensionless measure defined as [35]:

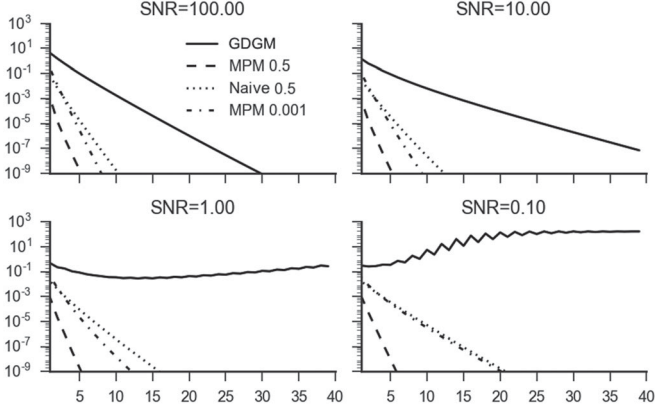$$\|P - P_{ref}\|_F^2 \Big/ \|P_{ref}\|_F^2. \tag{23}$$

Fig. 3. Typical convergence behavior (on abscissa the number of iterations and on the ordinate the convergence as defined in (24)) on simulated data for the gradient descent algorithm for estimating the geometric mean (GDGM), naive fixed point power mean with p = 0.5 and the MDM algorithm with p = {0.5, 0.001}, for N = 20 (dimension of input matrices), K = 100 (number of input matrices) and SNR = {100, 10, 1, 0.1} (22).

Fig. 4. Main effects average (bars) and sd (lines) number of iterations obtained across 50 repetitions for N = {10, 25, 50}, K = {10, 100, 500} and SNR = {100, 1, 0.01} for the MPM algorithm with p = {0.5, 0.25, 0.01}, the naive algorithm with p = {0.5, 0.01} and the gradient descent algorithm for estimating the geometric mean (GDGM).

As a stopping criterion, considering two successive iterations $P(i-1)$ and $P(i)$, we use

$$\frac{1}{N} \left\| P_{(i)}^{-1} P_{(i-1)} - I \right\|_F^2, \tag{24}$$

which magnitude does not depend on the size nor on the norm of the matrices.

We will also use simulated data to study the estimation of the geometric mean obtained by the gradient descent algorithm and by the procedure that uses the MPM algorithm as per Section II.C. We are interested in the relative error (23) of these estimations with respect to the 'true' geometric mean: according to our data generating model (22), the true geometric mean is the geometric mean of the signal part given by matrices $U D_k U^T$, where D_k, k = {1, ..., K} are diagonal matrices. Because of the congruence invariance of the geometric mean, the true geometric mean is $G_{\mathcal{G}}(w; \{U D_1 U^T, \ldots, U D_K U^T\}) = U G_{\mathcal{G}}(w; \{D_1, \ldots, D_K\}) U^T$ and has algebraic solution, since the geometric mean of diagonal matrices is their Log-Euclidean mean [1], i.e.,

$$G_{\mathcal{G}}(w; \{D_1, \ldots, D_K\}) = \mathrm{Exp} \sum_k w_k \mathrm{Log}(D_k). \tag{25}$$

C. Results

Figure 3 shows the typical convergence behavior for the gradient descent algorithm for computing the geometric mean (GDGM), the naive algorithm with p = 0.5 and the MPM algorithm (p = 0.5 and p = 0.001), for K = 100 input SPD matrices of dimension N = 20, and SNR = {100, 10, 1, 0.1}. This example illustrates the typical observed trend: the MPM algorithm is consistently faster as compared to both the naive and gradient descent algorithm. Moreover, it converges also in situations when the gradient descent and the naive algorithm do not (see also Fig. 4).

Figure 4 analyzes the convergence behavior of the naive fixed point, the MPM fixed point and GDGM. We show there the main
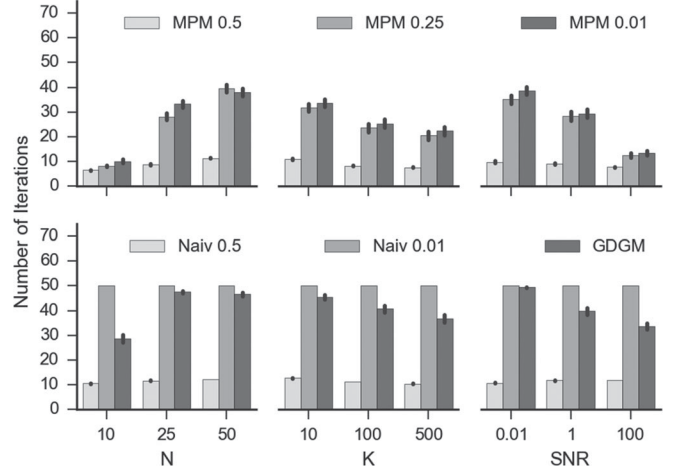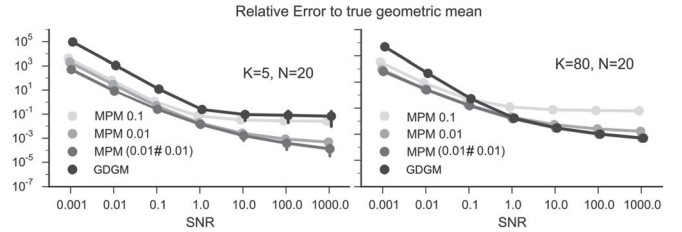

Fig. 5. Relative Error to the true geometric mean obtained with the GDGM algorithm, MPM with p = 0.1, MPM with p = 0.01 and as the midpoint of the geodesic joining the estimations obtained by MPM with p = ±0.01 (Section II.C). Left: N = 20, K = 5. Right: N = 20, K = 80. In both plots the horizontal axis is the SNR sampling the range $10^{-3}, \ldots, 10^3$.

effects (bars) and their standard deviation (sd: lines) across 50 simulations of N = {10, 25, 50}, K = {10, 100, 500} and SNR = {100, 1, 0.01} on the number of iterations. "Main effects" means that for each level of N, K and SNR the average and sd of the number of iterations are computed across all levels of the other two variables, as in a classical analysis of variance (ANOVA). We see that the number of iterations required by the MPM algorithm is always smaller as compared to the naive algorithm and that the naive algorithm converges very slow or does not converge at all for p = 0.01 (the maximum number of iterations allowed was fixed to 50 for all algorithms).

Figure 5 shows the relative error to the true geometric mean of the GDGM algorithm, MPM with p = 0.1, 0.01 and of the middle point of the geodesic joining the two MPM estimations obtained with p ± 0.01 (see Section II.C), for several SNR in the range SNR = $10^{-3}, \ldots, 10^3$, N = 20, and K = 5 (left) or K = 80 (right). We see that for negative SNR values (more noise than signal) all MPM-based estimations are closer to the true geometric mean as compared to the estimation offered by the gradient descent algorithm and that for all SNR values the midpoint of the geodesic joining the MPM estimations obtained with p ± 0.01 is as good as the best competitor, or better. Considering this
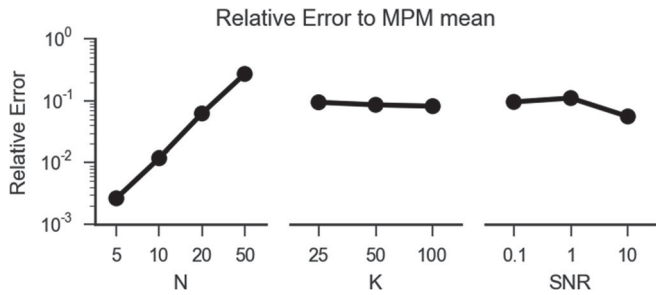
Fig. 6. Main effects across 50 simulations of N = {5, 10, 20, 50}, K = {25, 50, 100} and SNR = {0.1, 1, 101} on the Relative Error of the mean obtained by the ½MPM algorithm with respect to the mean obtained by the MPM algorithm with p = ½. Identical results are found taking p = −½.

and the convergence behavior of the MPM algorithm (Fig. 4), we conclude that the procedure based on MPM described on Section II.*G* is preferable for estimating the geometric mean. Figure 6 shows the relative error of the mean obtained by the ½MPM algorithm (see Appendix) to the mean obtained by the MPM algorithm for p = ½. For N small (N < 20) the approximation found by the ½MPM algorithm is good in absolute terms, but it deteriorates fast as N increases. In tolerating the error engendered by the approximation one should consider the noise level. In fact, any relative error can be considered satisfactory as long as the noise level is much higher than that and usually in practical applications the noise level is much higher than 10^{-1}.

IV. STUDIES WITH REAL DATA

A. Procedures

We tested the classification performance obtained by several power means on a real electroencephalography (EEG) data set acquired at the GIPSA-lab in Grenoble on 38 pairs of subjects participating to a Brain-Computer Interface (BCI) experiment. The BCI we used is the multi-subject *Brain Invaders* [38], which user-interface is similar to the joystick-controlled vintage video-game *Space Invaders* [39]. The BCI shows for several levels of the game 36 aliens on the screen and flash them in random patterns of 6 aliens [39]. The task of the participant is to destroy a TARGET alien only concentrating on it (i.e., without moving at all). The on-line classifier analyzes the event-related potentials (ERPs) produced during 1s after each flash and decides after every sequence of 12 flashes what alien is to be destroyed. The level continues until the TARGET alien is destroyed or 8 attempts have failed, after which a new level begins. For this analysis power means of special covariance matrices (see [5]) for the TARGET and NON-TARGET ERPs are estimated on a training set and the remaining trials are used for producing the area under the ROC curve (AUC). An AUC equal to 1 indicates perfect classification accuracy, while an AUC equal to 0.5 indicates random classification accuracy. We employed the Riemannian classifier described in [5], which uses only means of SPD matrices and distance function (4) in order to reach a decision. In the experiment, across subjects the average (sd) number of TARGET and NON-TARGET trials available was 109.9 (26.2)
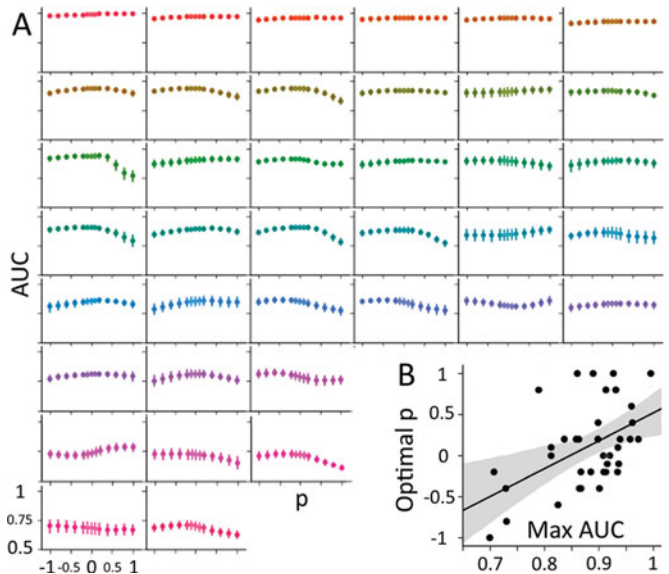

Fig. 7. A: from left to right and from top to bottom, AUC (disks) ± one standard deviation (vertical bars) obtained for 38 healthy subjects sorted by decreasing value of maximal AUC obtained across a sampling of power means in the interval p = [−1, . . . ,1]. B: scatter plot and regression line of the maximal AUC and the value of p allowing the maximal value. Each disk represents a subject.

and 549.48 (130.1), respectively. In order to keep the amount of data constant across subjects, only the first 80 TARGET and 400 NON-TARGET trials are used. AUC is evaluated by using a Monte Carlo cross-validation (MCCV) procedure averaging 10 random samples comprising 25% of the data selected as the test set and the remaining used as training set. EEG data were acquired by 16 scalp electrodes. Power means were tested at values of p = ±1, ±0.8, ±0.6, ±0.4, ±0.2, ±0.1, 0.

B. Results

The individual area under the ROC curve (AUC) for the brain-computer interface experiment on 38 subjects is shown in Fig. 7(A). The AUC as a function of p is a smooth curve. The value of p offering the maximum AUC appears to gravitate around zero. This illustrates a reason why the geometric mean is found useful in practice. However, the geometric mean (p = 0) is optimal only for three out of the 38 subjects and the optimal value of p is highly variable across individuals. This demonstrates that the use of power means instead of the sole geometric mean has potential to increase the accuracy. Finally, the Pearson correlation between the maximal value of AUC obtained and the value of p allowing such maximum is 0.49. A statistical test for the null hypothesis that this correlation is equal to zero against the alternative hypothesis that it is larger than zero gives a probability of type I error equal to 0.002. We therefore reject the null hypothesis and conclude that the higher the AUC, that is, the higher the SNR of the data, the higher the optimal value of p. This result matches our intuition; when the noise is higher than the signal, a power mean with negative p will suppress the noise more than the signal and vice versa.
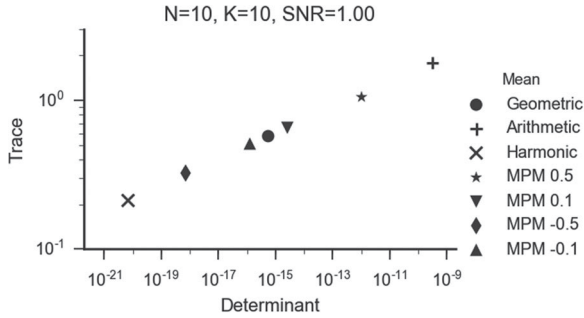
Fig. 8. *TraDe* plot obtained with N = 10, K = 10 and SNR = 1, for power means corresponding to p = 1 (Arithmetic), 0.5, 0.1, 0 (Geometric), −0.1, −0.5 and −1 (Harmonic). The relationship between the trace and the determinant of power means is log-log linear.

V. MEAN FIELDS

The family of power means is continuous and monotonic. Figure 8 is a *TraDe* plot (log-trace vs. log-determinant) for a sampling of power means along continuum p ∈ [−1, 1] illustrating the monotonicity of power means. We name a sampling of power means like those in Fig. 7 and Fig. 8 a Pythagorean *Mean Field*. Applications of mean fields include the possibility to evaluate the most appropriate choice of mean depending on its use and on the data at hand. Mean fields also allow robust extensions of current Riemannian classifiers, such as in [2]–[6], [10]–[13]. For instance, we may want to combine Riemannian classifiers applied to all the points of a mean field. The application of mean fields to real data will be the object of future investigations.

VI. CONCLUSIONS

Power means are generalized means interpolating continuously in the interval p ∈ [−1, 1], with p = 1 yielding the arithmetic mean, the limit of p→0 from both sides yielding the geometric mean and p = −1 yielding the harmonic mean. We have presented a new multiplicative algorithm for estimating power means of SPD matrices in the interval p ∈ (−1, 1)\{0}. A numerical analysis shows that its convergence rate is very fast and quasi-uniform for values of p close to ½ and –½, while for values of p close to 0 or ±1 it is still faster as compared to the gradient descent with fixed step-size used for estimating the geometric mean. Furthermore, it converges also in low SNR situations, whereas the gradient descent algorithm fails. The approximation to the geometric mean we have proposed in Section II.C gives better estimates of the geometric mean with respect to the gradient descent algorithm. We can therefore prefer MPM also for estimating the geometric mean. In conjunction with the procedure for p = 0 of Section II.C and expression (7) and (8) for p = 1 and p = −1, respectively, using the MPM algorithm we can now estimate a number of means sampling along the Pythagorean continuum p = [−1, 1].

The ½MPM algorithm offers a very efficient implementation to approximate the p = ±½ power mean. We have shown that the approximation is good in absolute terms for data matrices of small dimension (N < 20), but it may turn useful also in higher dimension for noisy data, whenever the noise level is

significantly higher than the error engendered by the approximation. However, while the MPM algorithm is computationally robust, ½MPM requires a careful implementation. In particular, Partlett's recurrence [33], [34] should be used for computing the inverse of triangular matrices and input matrices may need a normalization and/or regularization in order to allow proper Cholesky decompositions (see the Appendix).

Some works have focused on estimating geometric medians on Riemannian manifolds, extending the concept of median for random variables. This may provide a better estimation of the center of mass given a set of data points in the presence of outliers [40], [41]. Another current research direction is the definition of means of probability distributions rather than SPD matrices from a pure geometrical perspective, as we have done here. The geometric mean of Gaussian distributions and of mixtures of Gaussian distributions have been developed in [42], [43]. A one-parameter family of "p-means" for probability distributions, leading to the geometric mean and median as special cases, have been developed in [44], [45].

APPENDIX

We here develop a fast approximation to the MPM fixed point for the case p = ±½. The computational complexity of the MPM algorithm is dominated by the computation of matrix *H* at each iteration (see pseudo-code of the MPM algorithm in Section II.B). This requires 2K matrix multiplications and K eigenvalue-eigenvector (EVD) or Schur decompositions for evaluating the pth power of $XC_k^*X^T$. We here show how we can approximate *H* using 2K products of two triangular matrices and K matrix-triangular matrix multiplications. We wish an algorithm completely EVD-free and Schur decomposition-free, exploiting instead fast libraries for the Cholesky decomposition and triangular matrix operations.

First, we renounce to smart initialization, since it also requires EVD or Schur decompositions. Then, we factorize the products $XC_k^*X^T$ as products of triangular matrices and we consider a *truncated iterative approach* for approximating their square root; the approximation of the final solution will depend on the approximation engendered by this truncation. Finally, we will fix to 1.0 the φ parameter for the power $H^{-\varphi}$, that is, we will stick to fixed point definition (17) as it is; for |p| = ½ this value is acceptable in that it corresponds to the lower bound of the optimal interval (20) (see Fig. 2).

Our goal here is to find an approximation to $F_k = (XC_k^*X^T)^{1/2}$. Matrix *X* being an (inverse) square root of the sought mean *P*, we may take it lower triangular so as to verify $XPX^T = I$, hence $(X^TX)^{-1} = P$; what it matters is that F_k is taken as the principal square root of $XC_k^*X^T$, i.e., the unique symmetric one [14]. For computational reasons we will also maintain in memory an upper triangular matrix $Y^T = X^{-T}$. Before running the algorithm we will compute the Cholesky lower triangular factors L_k of all input matrices C_k^* and all the inverse transpose of these factors $R_k^T = L_k^{-T}$, which are upper triangular; these matrices are such that $L_kL_k^T = C_k^*$ and $R_k^TR_k = C_k^{*-1}$. As a consequence, matrices $M_k = XL_k$ are the Cholesky lower triangular factor of
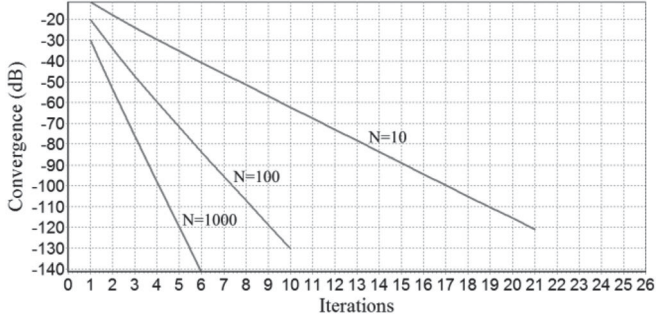
Fig. 9. Typical convergence behavior of the successive unit-norm normalization of the columns and rows of matrices with entry randomly drawn from a standard Gaussian distribution for N (matrix dimension) = 10, 100 and 1000. The convergence is the square root of the Euclidean mean square distance of the row and columns norm from unity after each iteration, in dB. The stopping criterion was set to convergence < −120 dB.

$XL_k L_k^T X^T = XC_k^* X^T$, i.e., $M_k M_k^T = XC_k^* X^T$. The approximation of F_k goes like this: we know that

$$F_k = \tfrac{1}{2}\left(U_k^T M_k + M_k^T U_k\right), \qquad (26)$$

where U_k is the orthogonal polar factor in the right polar decomposition $M_k = F_k U_k$ [35]. Since in our algorithm M_k can be computed easily by means of a triangular matrix multiplication, we can considerably speed up the algorithm approximating F_k by approximating U_k. We will obtain this by means of a suitable Newton's iteration converging to the polar factor U_k. We will use a well-established Newton's iteration known for its excellent numerical accuracy and quadratic convergence rate [35]–[37]. Another reason for choosing these iterations is that the first one turns out to be very efficient for us:

$$U_k = \tfrac{1}{2}\left(\sigma^{-1/2} M_k + \sigma^{1/2} M^{-T}\right) = \tfrac{1}{2}\left(\sigma^{-1/2} XL_k + \sigma^{1/2} Y^T R_k^T\right), \qquad (27)$$

where σ is a weight used for improving the convergence speed and $Y^T = X^{-T}$ can be computed fast by Parlett's recurrence [33], [34]. The optimal weight σ has been established to be a function of the maximum and minimum singular values of M_k [35]–[37]; in our case those can be found at no cost from the maximum and minimum entry on the diagonal of XL_k, which is triangular. In addition to the cited studies on polar orthogonal factor Newton's algorithms, we introduce here the following improved *a-posteriori* scaling: in order to make matrix U_k closer to orthogonal form after the first iteration, we require it to verify the following *diag-orthogonality condition*:

$$diag\left(U_k U_k^T\right) = diag\left(U_k^T U_k\right) = I, \qquad (28)$$

where the *diag* operator nullifies the off-diagonal elements of the argument. A finite sequence of the successive normalizations to unit norm of the columns and rows of U_k will converge to condition (28). The convergence rate is proportional of the size of U_k, as shown in Fig. 9 on random matrices. In this study we will use only two successive normalizations (error ≈ −20 dB for N = 10 as shown in Fig. 9). Notice that each normalization has only quadratic complexity on N, thus the diag-orthogonality scaling does not increase significantly the overall computational cost, however, it improves the final approximation. Finally, note

that although $F_k = \tfrac{1}{2}(U_k^T M_k + M_k^T U_k)$, in order to compute the average of the F_k over K only the $U_k^T M_k$ terms need to be averaged; we will symmetrize it once the mean over K has been computed. The resulting fast approximated algorithm follows:

Algorithm: Multiplicative ½ Power Mean (½MPM).

INPUT p $= \tfrac{1}{2}$ or p $= -\tfrac{1}{2}$. Given K positive weights $w = \{w_1, \ldots, w_K\}$ such that $\Sigma_k w_k = 1$ and K N·N SPD matrices $C^* = \{C_1^*, \ldots, C_K^*\}$, with $* = \mathrm{sgn}(p)$, input the K Cholesky lower triangular factors L_k and the K transpose inverse $R_k^T = L_k^{-T}$ of the input matrices.
OUTPUT P, an approximated w-weighted Power Mean of order p.
Note: H and U are square matrices. X, Z and L_k are lower triangular matrices. Y^T, Z^T and R_k^T are upper triangular matrices.
Initialize X to the identity lower triangular matrix and set $Y^T = X^T$.
Set ζ equal to a small floating precision number (e.g., 10^{-10})

REPEAT
    $H \leftarrow 0$
    For k: $= 1$ **to K do**
    Begin
        $M \leftarrow XL_k$
        $\sigma = \min(diag(M)) \cdot \max(diag(M))$
        $U = \sigma^{-1/2} M + \sigma^{1/2} Y^T R_k^T$
        Normalize to unit norm the columns, then the rows
        of U, two times in this order.
        $H \leftarrow H + w_k U^T M$
    End
    $H \leftarrow \tfrac{1}{2}(H + H^T)$
    Do Cholesky $(H) = ZZ^T$;
    $X \leftarrow Z^{-1} X$;
    $Y^T \leftarrow Z^T Y^T$
UNTIL $\tfrac{1}{\sqrt{N}}\|(H - I)\|_F < \varepsilon$

$$\textbf{\textit{RETURN }} P = \begin{cases} Y^T Y & if\ \mathrm{p} = \tfrac{1}{2} \\ X^T X & if\ \mathrm{p} = -\tfrac{1}{2} \end{cases}$$

REFERENCES

[1] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Geometric means in a novel vector space structure on symmetric positive-definite matrices," *SIAM. J. Matrix Anal. Appl.*, vol. 29, no. 1, pp. 328–347, 2007.
[2] M. Arnaudon, F. Barbaresco, and L. Yang, "Riemannian medians and means with applications to radar signal processing," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 4, pp. 595–604, Aug. 2013.
[3] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Multi-class brain computer interface classification by Riemannian geometry," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 4, pp. 920–928, Apr. 2012.
[4] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Classification of covariance matrices using a Riemannian-based kernel for BCI applications," *Neurocomputing*, vol. 112, pp. 172–178, 2013.
[5] M. Congedo, "EEG source analysis," HDR presented at the Doctoral School EDISCE, Grenoble Alpes Univ., Grenoble, France, 2013.
[6] E. K. Kalunga, S. Chevallier, Q. Barthélemy, K. Djouani, E. Monacelli, and Y. Hamam, "Online SSVEP-based BCI using Riemannian geometry," *Neurocomputing*, vol. 191, pp. 55–68, 2016.

[7] M. Faraki, M. T. Harandi, and F. Porikli, "More about VLAD: A leap from Euclidean to Riemannian manifolds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 2015, pp. 4951–4960.

[8] P. Fillard, V. Arsigny, N. Ayache, and X. Pennec, "A Riemannian framework for the processing of tensor-valued images," in *Proc. 1st Int. Conf. Deep Struct., Singularities, Comput. Vis.*, 2005, pp. 112–123.

[9] P. T. Fletcher, "Geodesic regression and the theory of least squares on Riemannian manifolds," *Int. J. Comput. Vis.*, vol. 105, pp. 171–185, 2012.

[10] Y. Li and K. M. Wong, "Riemannian distances for signal classification by power spectral density," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 4, pp. 655–669, Aug. 2013.

[11] Y. Li, K. M. Wong, and H. Debruin, "EEG signals classification for sleep-state decision—A Riemannian geometry approach," *IET Signal Process.*, vol. 6, no. 4, pp. 288–299, 2012.

[12] M. Moakher, "On the averaging of symmetric positive-definite tensors," *J. Elasticity.*, vol. 82, no. 3, pp. 273–296, 2006.

[13] X. Zhang, Y. Wang, M. Gou, M. Sznaier, and O. Camps, "Efficient temporal sequence comparison and classification using gram matrix embeddings on a Riemannian manifold," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4498–4507.

[14] R. Bhatia, *Positive Definite Matrices*. Princeton, NJ, USA: Princeton Univ. Press, 2007.

[15] S. Sra, "Positive definite matrices and the S-divergence," *Proc. Amer. Math. Soc.*, vol. 144, pp. 2787–2797, 2016.

[16] Z. Chebbi and M. Moakher, "Means of Hermitian positive-definite matrices based on the log-determinant α-divergence function," *Linear Algebra Appl.*, vol. 436, no. 7, pp. 1872–1889, 2012.

[17] M. Moakher and M. Zéraï, "The Riemannian geometry of the space of positive-definite matrices and its application to the regularization of positive-definite matrix-valued data," *J. Math. Imaging Vis.*, vol. 40, pp. 171–187, 2011.

[18] M. Moakher, "A differential geometric approach to the arithmetic and geometric means of operators in some symmetric spaces," *SIAM. J. Matrix Anal. Appl.*, vol. 26, no. 3, pp. 735–747, 2005.

[19] R. Bhatia and J. Holbrook, "Riemannian geometry and matrix geometric mean," *Linear Algebra Appl.*, vol. 413, pp. 594–618, 2006.

[20] N. Nakamura, "Geometric means of positive operators," *Kyungpook Math. J.*, vol. 49, pp. 167–181, 2009.

[21] T. T. Georgiou, "Distances and Riemannian metrics for spectral density functions," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 3995–4003, Aug. 2007.

[22] X. Jiang, L. Ning, and T. T. Georgiou, "Distances and Riemannian metrics for multivariate spectral densities," *IEEE Trans. Autom. Control*, vol. 57, no. 7, pp. 1723–1735, Jul. 2012.

[23] M. Congedo, B. Afsari, A. Barachant, and M. Moakher, "Approximate joint diagonalization and geometric mean of symmetric positive definite matrices," *PLoS ONE*, vol. 10, no. 4, 2015, Art. no. e0121423.

[24] Y. Lim and M. Pálfia, "Matrix power means and the Karcher mean," *J. Funct. Anal.*, vol. 262, pp. 1498–1514, 2012.

[25] J. Lawson and Y. Lim, "Weighted means and Karcher equations of positive operators," *PNAS*, vol. 110, no. 39, pp. 15626–15632, 2013.

[26] J. Lawson and Y. Lim, "Karcher means and Karcher equations of positive definite operators," *Trans. Amer. Math. Soc.* vol. 1, pp. 1–22, 2014.

[27] M. Pálfia, "Operator means of probability measures and generalized Karcher equations," *Adv. Math.*, vol. 289, pp. 951–1007, 2016.

[28] M. Congedo, R. Phlypo, and A. Barachant, "A fixed-point algorithm for estimating power means of positive definite matrices," in *Proc. 24th Eur. Signal Process. Conf.*, 2016, pp. 2106–2110.

[29] T. Ando, C. K. Li, and R. Mathias, "Geometric means," *Linear Algebra Appl.*, vol. 385, pp. 305–334, 2004.

[30] B. Afsari, R. Tron, and R. Vidal, "On the convergence of gradient descent for finding the Riemannian center of mass," *SIAM J. Control Optim.*, vol. 51, no. 3, pp. 2230–2260, 2013.

[31] B. Jeuris, R. Vanderbril, and B. Vandereycken, "A survey and comparison of contemporary algorithms for computing the matrix geometric mean," *Electron. Trans. Numer. Anal.*, vol. 39, pp. 379–402, 2012.

[32] T. Zhang, "A majorization-minimization algorithm for the Karcher mean of positive definite matrices," *arXiv:1312.4654*, 2014.

[33] B. N. Parlett, "A recurrence among the elements of functions of triangular matrices," *Linear Algebra Appl.*, vol. 14, no. 2, pp. 117–121, 1976.

[34] P. I. Davies, and N. J. Higham, "A Schur-Parlett algorithm for computing matrix functions," *SIAM. J. Matrix Anal. Appl.*, vol. 25, no. 2, pp. 464–485, 2003.

[35] N. J. Higham, "Stable iterations for the matrix square root," *Numer. Algorithms*, vol. 15, no. 2, pp. 227–242, 1997.

[36] C. Kenney and A. J. Laub, "On scaling Newton's method for polar decomposition and the matrix sign function," *SIAM. J. Matrix Anal. Appl.*, vol. 13, no. 3, pp. 688–706, 1992.

[37] A. Kielbasinski, P. Zielinski and K. Zietak, "On iterative algorithms for the polar decomposition of a matrix and the matrix sign function," *Appl. Math. Comput.*, vol. 270, pp. 483–495, 2015.

[38] L. Korczowski, M. Congedo, and C. Jutten, "Single-trial classification of multi-user P300-based brain-computer interface using Riemannian geometry," in *Proc. 37th Int. Conf. IEEE Eng. Med. Biol. Soc.*, Milano, Italy, 2015, pp. 1769–1772.

[39] M. Congedo *et al.*, "'Brain invaders': A prototype of an open-source P300-based video game working with the OpenViBE platform," in *Proc. 5th Int. BCI Conf.*, 2011, pp. 280–283.

[40] F. Barbaresco, "Information geometry of covariance matrix: Cartan-Siegel homogeneous bounded domains, Mostow/Berger fibration and Frechet median," in *Matrix Information Geometry*, R. Bhatia and F. Nielsen, Eds. New York, NY, USA: Springer-Verlag, 2012, pp. 199–256.

[41] P. T. Fletcher, S. Venkatasubramanian, and S. Joshi, "The geometric median on Riemannian manifolds with application to robust atlas estimation," *Neuroimage*, vol. 45, no. 1 suppl., pp. S143–S152, 2009.

[42] S. Said, L. Bombrun, Y. Berthoumieu, and J. H. Manton, "Riemannian Gaussian distributions on the space of symmetric positive definite matrices," *IEEE Trans. Inf. Theory*, vol. PP, no. 99, pp. 1–1, 2017.

[43] P. Zanini, M. Congedo, C. Jutten, S. Said, and Y. Berthoumieu, "Parameters estimate of Riemannian Gaussian distribution in the manifold of covariance matrices," in *Proc. IEEE Sensor Array Multichannel Signal Process. Workshop*, 2016, pp. 1–5.

[44] M. Arnaudon and L. Miclo, "A stochastic algorithm finding p-means on the circle," *Bernoulli*, vol. 22, no. 4, pp. 2237–2300, 2016.

[45] M. Arnaudon and L. Miclo, "A stochastic algorithm finding generalized means on compact manifolds," *Stoch. Process. Appl.*, vol. 124, pp. 3463–3479, 2014.