Geometric Optimization in Machine Learning

Suvrit Sra and Reshad Hosseini

Abstract Machine learning models often rely on sparsity, low-rank, orthogonality, correlation, or graphical structure. The structure of interest in this chapter is geometric, specifically the manifold of positive definite (PD) matrices. Though these matrices recur throughout the applied sciences, our focus is on more recent developments in machine learning and optimization. In particular, we study (i) models that might be nonconvex in the Euclidean sense but are convex along the PD manifold; and (ii) ones that are fully nonconvex but are nevertheless amenable to global optimization. We cover basic theory for (i) and (ii); subsequently, we present a scalable Riemannian limited-memory BFGS algorithm (that also applies to other manifolds). We highlight some applications from statistics and machine learning that benefit from the geometric structure studies.

1 Introduction

Fitting mathematical models to data invariably requires numerical optimization. The field is thus replete with tricks and techniques for better modeling, analysis, and implementation of optimization algorithms. Among other aspects, the notion of "structure," is of perennial importance: its knowledge often helps us obtain faster algorithms, permit scalability, gain insights, or capture a host of other attributes.

Structure has multifarious meanings, of which perhaps the best known is sparsity [5, 52]. But our focus is different: we study geometric structure, in particular where model parameters lie on a Riemannian manifold.

Suvrit Sra
Massachusetts Institute of Technology, Cambridge, MA, USA e-mail: suvrit@mit.edu

Reshad Hosseini
School of ECE, College of Engineering, University of Tehran, Tehran, Iran e-mail: reshad.hosseini@ut.ac.ir

Geometric structure has witnessed increasing interest, for instance in optimization over matrix manifolds (including orthogonal, low-rank, positive definite matrices, among others) [1, 12, 50, 55]. However, in distinction to general manifold optimization, which extends most Euclidean schemes to Riemannian manifolds [1, 53], we focus on specific case of "geometric optimization" problems that exploit the special structure of the manifold of positive definite (PD) matrices. Two geometric aspects play a crucial role here: (i) the nonpositive curvature of the manifold, which allows defining a global curved notion of convexity along geodesics on the manifold; and (ii) the convex (Euclidean) conic structure (e.g., as used in Perron-Frobenius theory, which includes the famous PageRank algorithm as a special case).

One of our key motivations for studying geometric optimization is that for many problems it may help uncover hidden (geodesic) convexity, and thus provably place global optimization of certain nonconvex problems within reach [50]. Moreover, exploiting geometric convexity can have remarkable empirical consequences for problems involving PD matrices [47]; which persist even without overall (geodesic) convexity, as will be seen in §3.1.

Finally, since PD matrices are ubiquitous in not only machine learning and statistics, but throughout the applied sciences, the new modeling and algorithmic tools offered by geometric optimization should prove to be valuable in many other settings. To stoke the reader's imagination beyond the material described in this chapter, we close with a short list of further applications in Section 3.3. We also refer the reader to our recent work [50], that develops the theoretical material related to geometric optimization in greater detail.

With this background, we are now ready to recall the geometric concepts at the heart of our presentation, before moving on to detailed applications of our ideas.

1.1 Manifolds and Geodesic Convexity

A smooth manifold is a space that locally resembles Euclidean space [29]. We focus on Riemannian manifolds (smooth manifolds equipped with a smoothly varying inner product on the tangent space) as their geometry permits a natural extension of many nonlinear optimization algorithms [1, 53].

In particular, we focus on the (matrix) manifold of real symmetric positive definite (PD) matrices. Most of the ideas that we describe apply more broadly to *Hadamard manifolds* (i.e., Riemannian manifolds with non-positive curvature), but we limit attention to the PD manifold for concreteness and due to its vast importance in machine learning and beyond [6, 17, 45].

A key concept on manifolds is that of *geodesics* which are curves that join points along shortest paths. Geodesics help one extend the notion of convexity to *geodesic convexity*. Formally, suppose $\mathscr{M}$ is a Riemmanian manifold, and x, y are two points on $\mathscr{M}$. Say γ is a unit speed geodesic joining x to y, such that

$$\gamma_{xy} : [0,1] \to \mathscr{M}, \quad \text{s.t.} \quad \gamma_{xy}(0) = x,\ \gamma_{xy}(1) = y.$$

Then, we call a set $\mathscr{A} \subseteq \mathscr{M}$ *geodesically convex*, henceforth g-convex, if the geodesic joining an arbitrary pair of points in $\mathscr{A}$ lies completely in $\mathscr{A}$. We say $f : \mathscr{A} \to \mathbb{R}$ is g-convex if for all $x, y \in \mathscr{A}$, the composition $f \circ \gamma_{xy} : [0,1] \to \mathbb{R}$ is convex in the usual sense. For example, on the manifold $\mathbb{P}_d$ of $d \times d$ PD matrices the geodesic γ_{XY} between $X, Y \in \mathbb{P}_d$ has the beautiful closed-form [6, Ch. 6]:

$$\gamma_{XY}(t) := X^{1/2}(X^{-1/2}YX^{-1/2})^t X^{1/2}, \quad 0 \le t \le 1. \tag{1}$$

It is common to write $X \sharp_t Y \equiv \gamma_{XY}(t)$, and we also use this notation for brevity. Therewith, a function $f : \mathbb{P}_d \to \mathbb{R}$ is *g-convex* if on a g-convex set $\mathscr{A}$ it satisfies

$$f(X \sharp_t Y) \le (1-t)f(X) + tf(Y), \qquad t \in [0,1],\ X, Y \in \mathscr{A}. \tag{2}$$

G-convex functions are remarkable in that they can be nonconvex in the Euclidean sense, but can still be globally optimized. Such functions on PD matrices have already proved important in several recent applications [17,18,23,44,48,49,57,58,62]. We provide below several examples, and refer the interested reader to our work [50] for a detailed, more systematic development of g-convexity on PD matrices.

Example 1 ([50]). The following functions are g-convex on $\mathbb{P}_d$: (i) $\operatorname{tr}(e^A)$; (ii) $\operatorname{tr}(A^\alpha)$ for $\alpha \ge 1$; (iii) $\lambda_1^{\downarrow}(e^A)$; (iv) $\lambda_1^{\downarrow}(A^\alpha)$ for $\alpha \ge 1$.

Example 2 ([50]). Let $X \in \mathbb{C}^{d \times k}$ be an arbitrary rank-k matrix ($k \le d$), then $A \mapsto \operatorname{tr} X^* A X$ is log-g-convex, that is,

$$\operatorname{tr} X^*(A \sharp_t B)X \le [\operatorname{tr} X^* A X]^{1-t} [\operatorname{tr} X^* B X]^t, \qquad t \in [0,1]. \tag{3}$$

Inequality (3) depends on a nontrivial property of $\sharp_t$ proved e.g., in [50, Thm. 2.8].

Example 3. If $h : \mathbb{R}_+ \to \mathbb{R}_+$ is nondecreasing and log-convex, then the map $A \mapsto \sum_{i=1}^k \log h(\lambda_i(A))$ is g-convex. For instance, if $h(x) = e^x$, we obtain the special case that $A \mapsto \log \operatorname{tr}(e^A)$ is g-convex.

Example 4. Let $A_i \in \mathbb{C}^{d \times k}$ with $k \le d$ such that $\operatorname{rank}([A_i]_{i=1}^m) = k$; also let $B \succeq 0$. Then $\phi(X) := \log\det(B + \sum_i A_i^* X A_i)$ is g-convex on $\mathbb{P}_d$.

Example 5. The *Riemannian distance* $\delta_R(A,X) := \|\log(X^{-1/2}AX^{-1/2})\|_{\mathrm{F}}$ between $A, X \in \mathbb{P}_d$ [6, Ch. 6] is well-known to be jointly g-convex, see e.g., [6, Cor. 6.1.11]. To obtain an infinite family of such g-convex distances see [50, Cor. 2.19].

Consequently, the Fréchet (Karcher) mean and median of PD matrices are g-convex optimization problems. Formally, these problems seek to solve

$$\begin{aligned} &\min_{X \succ 0} \quad \textstyle\sum_{i=1}^m w_i \delta_R(X, A_i), \qquad \text{(Geometric Median)}, \\ &\min_{X \succ 0} \quad \textstyle\sum_{i=1}^m w_i \delta_R^2(X, A_i), \qquad \text{(Geometric Mean)}, \end{aligned}$$

where $\sum_i w_i = 1$, $w_i \ge 0$, and $A_i \succ 0$ for $1 \le i \le m$. The latter problem has received extensive interest in the literature [6–8, 25, 39, 41, 48]. Its optimal solution is unique owing to the strict g-convexity of its objective.

1.2 Beyond g-Convexity: Thompson-Nonexpansivity

We highlight now a special class of nonconvex functions that is amenable to global optimization without requiring g-convexity. Specifically, we consider functions that admit "sup norm" contractions, namely contractions under the *Thompson metric*:

$$\delta_T(X,Y) := \|\log(Y^{-1/2}XY^{-1/2})\|, \tag{4}$$

where $\|\cdot\|$ is the usual operator norm (hence the 'sup'). This metric is an object of great interest in nonlinear Perron-Frobenius theory [28, 30].

We consider maps non-expansive under the Thompson metric (4). Since the metric space $(\mathbb{P}_d, \delta_T)$ is complete, non-expansive maps under this metric provide fertile grounds for designing convergent iterative algorithms (using fixed-point theory) without needing g-convexity. We say $\Phi : \mathbb{P}_d \to \mathbb{P}_d$ is *Thompson non-expansive* if

$$\delta_T(\Phi(X),\Phi(Y)) \le q\delta_T(X,Y), \qquad 0 \le q \le 1. \tag{5}$$

If $q < 1$, then Φ is called *q-contractive*. Since the Thompson metric is generated by the operator norm, it turns out to satisfy a larger body of properties (than δ_R) that are useful for analyzing fixed-point iterations. We recall some of these properties below—for details please see [28, 30, 31, 50].

Proposition 1. *Unless noted otherwise, all matrices are assumed to be PD.*

$$\begin{aligned}
\delta_T(X^{-1},Y^{-1}) &= \delta_T(X,Y) && && (6a)\\
\delta_T(B^*XB,B^*YB) &= \delta_T(X,Y), && B \in GL_n(\mathbb{C}) && (6b)\\
\delta_T(X^t,Y^t) &\le |t|\delta_T(X,Y), && \text{for } t \in [-1,1] && (6c)\\
\delta_T\Big(\textstyle\sum_i w_iX_i, \sum_i w_iY_i\Big) &\le \max_{1\le i\le m} \delta_T(X_i,Y_i), && w_i \ge 0, w \ne 0 && (6d)\\
\delta_T(X+A,Y+A) &\le \frac{\alpha}{\alpha+\beta}\delta_T(X,Y), && A \succeq 0, && (6e)
\end{aligned}$$

where $\alpha = \max\{\|X\|,\|Y\|\}$ *and* $\beta = \lambda_{\min}(A)$. *Moreover, for* $X \in \mathbb{C}^{d\times k}$ $(k \le d)$ *with full column rank we have the* compression inequality *[50, Thm. 4.3]:*

$$\delta_T(X^*AX,X^*BX) \le \delta_T(A,B). \tag{6f}$$

1.2.1 Why Thompson nonexpansivity?

Below we review a setting where Thompson nonexpansivity is useful. Consider the optimization problem: $\min_{S\succ 0} \Phi(S)$, where Φ is continuously differentiable on $\mathbb{P}_d$. Since the constraint set is open, a necessary condition of optimality of a point S^* is that its gradient vanishes, that is,

$$\nabla\Phi(S^*) = 0. \tag{7}$$

Various approaches could be used for solving the nonlinear (matrix) equation (7). And among these, fixed-point iterations may be particularly attractive. Here, one designs a map $\mathscr{G} : \mathbb{P}_d \to \mathbb{P}_d$, using which we can rewrite (7) in the form

$$S^* = \mathscr{G}(S^*), \tag{8}$$

that is, S^* is a fixed-point of $\mathscr{G}$, and by construction a stationary point of Φ.

Typically, finding fixed-points is difficult. However, if the map $\mathscr{G}$ can be chosen such that it is Thompson contractive, then simply running the Picard iteration

$$S_{k+1} \leftarrow \mathscr{G}(S_k), \quad k = 0, 1, \ldots, \tag{9}$$

will yield a unique solution to (7)—both existence and uniqueness follow from the Banach contraction theorem. The reason we insist on Thompson contractivity is because many of our examples fail to be Euclidean contractions (or even Riemannian contractions) but end up being Thompson contractions. Thus, studying Thompson nonexpansivity is valuable. We highlight below a concrete example that arises in some applications [15–18, 61], and is not a Euclidean but a Thompson contraction.

Application: Geometric Mean of PD Matrices.

Let $A_1, \ldots, A_n \in \mathbb{P}_d$ be input matrices and $w_i \geq 0$ be nonnegative weights such that $\sum_{i=1}^n w_i = 1$. A particular geometric mean of the $\{A_i\}_{i=1}^n$, called the *S-mean*, is obtained by computing [15, 18]

$$\min_{X \succ 0} \; h(X) := \sum\nolimits_{i=1}^n w_i \delta_S^2(X, A_i), \tag{10}$$

where δ_S^2 is the squared *Stein-distance*

$$\delta_S^2(X,Y) := \log\det\left(\tfrac{X+Y}{2}\right) - \tfrac{1}{2}\log\det(XY), \qquad X, Y \succ 0. \tag{11}$$

It can be shown that δ_S^2 is strictly g-convex (in both arguments) [48]. Thus, Problem (10) is a g-convex optimization problem. It is easily seen to possess a solution, whence the strict g-convexity of $h(X)$ immediately implies that this solution must be unique. What remains is to obtain an algorithm to compute this solution.

Following (7), we differentiate $h(X)$ and obtain the nonlinear matrix equation

$$0 = \nabla h(X) \equiv X^{-1} = \sum\nolimits_i w_i \left(\tfrac{X+A_i}{2}\right)^{-1},$$

from which we naively obtain the Picard iteration

$$X_{k+1} \leftarrow \left[\sum\nolimits_i w_i \left(\tfrac{X_k+A_i}{2}\right)^{-1}\right]^{-1}. \tag{12}$$

Applying (6a), (6d), and (6e) in sequence we see that (12) is Thompson contraction, which immediately allows us to conclude its validity as a Picard iteration and its linear rate of convergence (in the Thompson metric) to the global optimum of (10).

2 Manifold Optimization

Creating fixed-point iterations is somewhat of an art, and it is not always clear how to obtain one for a given problem. Therefore, developing general purpose iterative optimization algorithms is of great practical importance.

For Euclidean optimization a common recipe is to iteratively (a) find a descent direction; and (b) obtain sufficient decrease via line-search which also helps ensure convergence. We follow a similar recipe for Riemannian manifolds by replacing Euclidean concepts by their Riemannian counterparts. For example, we now compute descent directions in the tangent space. At a point X, the tangent space T_X is the approximating vector space (see Fig. 1). Given a descent direction $\xi_X \in T_X$, we perform line-search along a smooth curve on the manifold (red curve in Fig. 1). The derivative of this curve at X provides the descent direction ξ_X. We refer the reader to [1, 53] for an in depth introduction to manifold optimization.

Euclidean methods such as conjugate-gradient and LBFGS combine gradients at the current point with gradients and descent directions at previous points to obtain a new descent direction. To adapt such algorithms to manifolds we need to define how to transport vectors in a tangent space at one point to vectors in a tangent space at another point.

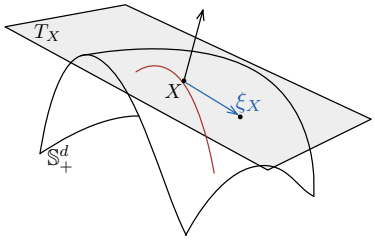


Fig. 1 Line-search on a manifold: X is a point on the manifold, T_X is the tangent space at the point X, ξ_X is a descent direction at X; the red curve is the curve along which line-search is performed.

On Riemannian manifolds, the gradient is a direction in the tangent space, where the inner-product of the gradient with another direction in the tangent space gives the directional derivative of the function. Formally, if g_X defines the inner product in the tangent space T_X, then

$$Df(X)\xi = g_X(\text{grad} f(X), \xi), \quad \text{for } \xi \in T_X.$$

Given a descent direction the curve along which we perform line-search can be a geodesic. A map that combines the direction and a step-size to obtain a corresponding point on the geodesic is called an *exponential map*. Riemannian manifolds also come equipped with a natural way to transport vectors on geodesics that is called parallel transport. Intuitively, a parallel transport is a differential map with zero derivative along the geodesics.

Using these ideas, and in particular deciding where to perform the vector transport we can obtain different variants of Riemannian LBFGS. We recall one specific LBFGS variant from [50] (presented as Alg. 1), which yields the best performance in our applications, once we combine it with a suitable line-search algorithm.

In particular, to ensure Riemannian LBFGS always produces a descent direction, we must ensure that the line-search algorithm satisfies the *Wolfe conditions* [44]:

Table 1 Summary of key Riemannian objects for the PD matrix manifold.

Definition	Expression for PD matrices
Tangent space	Space of symmetric matrices
Metric between two tangent vectors ξ, η at Σ	$g_\Sigma(\xi,\eta) = \mathrm{tr}(\Sigma^{-1}\xi\Sigma^{-1}\eta)$
Gradient at Σ if Euclidean gradient is $\nabla f(\Sigma)$	$\mathrm{grad} f(\Sigma) = \frac{1}{2}\Sigma(\nabla f(X) + \nabla f(X)^T)\Sigma$
Exponential map at point Σ in direction ξ	$R_\Sigma(\xi) = \Sigma \exp(\Sigma^{-1}\xi)$
Parallel transport of tangent vector ξ from Σ_1 to Σ_2	$\mathscr{T}_{\Sigma_1,\Sigma_2}(\xi) = E\xi E^T, \quad E = (\Sigma_2\Sigma_1^{-1})^{1/2}$

$$f(R_{X_k}(\alpha\xi_k)) \leq f(X_k) + c_1\alpha Df(X_k)\xi_k, \tag{13}$$
$$Df(X_{k+1})\xi_{k+1} \geq c_2 Df(X_k)\xi_k, \tag{14}$$

where $0 < c_1 < c_2 < 1$. Note that $\alpha Df(X_k)\xi_k = g_{X_k}(\mathrm{grad} f(X_k), \alpha\xi_k)$, i.e., the derivative of $f(X_k)$ in the direction $\alpha\xi_k$ is the inner product of descent direction and gradient of the function. Practical line-search algorithms implement a stronger (Wolfe) version of (14) that enforces

$$|Df(X_{k+1})\xi_{k+1}| \leq c_2 Df(X_k)\xi_k. \tag{15}$$

Key details of a practical way to implement this line-search may be found in [23].

Algorithm 1 Pseudocode for Riemannian LBFGS

Given: Riemannian manifold $\mathscr{M}$ with Riemannian metric g; parallel transport $\mathscr{T}$ on $\mathscr{M}$; geodesics R; initial value X_0; a smooth function f
Set initial $H_{\text{diag}} = 1/\sqrt{g_{X_0}(\mathrm{grad} f(X_0), \mathrm{grad} f(X_0))}$
for $k = 0, 1, \ldots$ **do**
  Obtain descent direction ξ_k by unrolling the RBFGS method
  Compute $\xi_k \leftarrow \textsc{HessMul}(-\mathrm{grad} f(X_k), k)$
  Use line-search to find α such that it satisfies Wolfe conditions
  Calculate $X_{k+1} = R_{X_k}(\alpha\xi_k)$
  Define $S_k = \mathscr{T}_{X_k, X_{k+1}}(\alpha\xi_k)$
  Define $Y_k = \mathrm{grad} f(X_{k+1}) - \mathscr{T}_{X_k, X_{k+1}}(\mathrm{grad} f(X_k))$
  Update $H_{\text{diag}} = g_{X_{k+1}}(S_k, Y_k)/g_{X_{k+1}}(Y_k, Y_k)$
  Store Y_k; S_k; $g_{X_{k+1}}(S_k, Y_k)$; $g_{X_{k+1}}(S_k, S_k)/g_{X_{k+1}}(S_k, Y_k)$; H_{diag}
end for
return X_k
function $\textsc{HessMul}(P, k)$
if $k > 0$ **then**
  $P_k = P - \frac{g_{X_{k+1}}(S_k, P_{k+1})}{g_{X_{k+1}}(Y_k, S_k)} Y_k$
  $\hat{P} = \mathscr{T}_{X_{k+1}, X_k} \textsc{HessMul}(\mathscr{T}_{X_k, X_{k+1}} P_k, k-1)$ **return** $\hat{P} - \frac{g_{X_{k+1}}(Y_k, \hat{P})}{g_{X_{k+1}}(Y_k, S_k)} S_k + \frac{g_{X_{k+1}}(S_k, S_k)}{g_{X_{k+1}}(Y_k, S_k)} P$
else
  return $H_{\text{diag}} P$
end if
end function

3 Applications

We are ready to present two applications of geometric optimization. Section 3.1 summarizes recent progress in fitting Gaussian Mixture Models (GMMs), for which g-convexity proves remarkably useful and ultimately helps Alg. 1 to greatly outperform the famous Expectation Maximization (EM) algorithm—this is remarkable as previously many believed it impossible to outdo EM via general nonlinear optimization techniques. Next, in Section 3.2 we present an application to maximum likelihood parameter estimation for non-Gaussian elliptically contoured distributions. These problems are Euclidean nonconvex but often either g-convex or Thompson nonexpansive, and thus amenable to geometric optimization.

3.1 Gaussian Mixture Models

This material of this section is based on the authors' recent work [23]; the interested reader is encouraged to consult that work for additional details.

Gaussian Mixture Models (GMMs) have a long history in machine learning and signal processing and continue to enjoy widespread use [10, 21, 36, 40]. For GMM parameter estimation, Expectation Maximization (EM) [20] still seems to be the *de facto* choice—although other approaches have also been considered [43], typical nonlinear programming methods such as conjugate gradients, quasi-Newton, Newton, are usually viewed as inferior to EM [59].

One advantage that EM enjoys is that its M-Step satisfies the PD constraint on covariances by construction. Other methods often struggle when dealing with this constraint. An approach is to make the problem unconstrained by performing a change-of-variables using Cholesky decompositions (as also exploited in semidefinite programming [14]). Another possibility is to formulate the PD constraint via a set of smooth convex inequalities [54] or to use log-barriers and to invoke interior-point methods. But such methods tend to be much slower than EM-like iterations, especially in higher dimensions [49].

Driven by these concerns the authors view GMM fitting as a manifold optimization problem in [23]. But surprisingly, an out-of-the-box invocation of manifold optimization completely fails! To compete with and to outdo EM, further work is required: *g-convexity supplies the missing link.*

3.1.1 Problem Setup

Let $\mathcal{N}$ denote the Gaussian density with mean $\mu \in \mathbb{R}^d$ and covariance $\Sigma \in \mathbb{P}_d$, i.e.,

$$\mathcal{N}(x;\mu,\Sigma) := \det(\Sigma)^{-1/2}(2\pi)^{-d/2}\exp\big(-\tfrac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\big).$$

A Gaussian Mixture Model has the probability density

$$p(x) := \sum_{j=1}^{K} \alpha_j \mathcal{N}(x; \mu_j, \Sigma_j), \qquad x \in \mathbb{R}^d,$$

where $\alpha \in \Delta_K$, the K-dimensional probability simplex, and $\{\mu_j \in \mathbb{R}^d, \Sigma_j \succ 0\}_{j=1}^K$. Given i.i.d. samples $\{x_1, \ldots, x_n\}$, we wish to estimate these parameters by maximum likelihood. This leads to the *GMM optimization* problem

$$\max_{\alpha \in \Delta_K, \{\mu_j, \Sigma_j \succ 0\}_{j=1}^K} \quad \sum_{i=1}^{n} \log\Big(\sum_{j=1}^{K} \alpha_j \mathcal{N}(x_i; \mu_j, \Sigma_j)\Big). \tag{16}$$

Problem (16) is well-known to be a difficult nonconvex problem. So like EM, we also seek only efficient computation of local solutions. As alluded to above, before we can successfully apply manifold optimization (in particular, our LBFGS algorithm) to solve (16), we need to expose its g-convexity.

To that end, we begin with maximum likelihood estimation for a single Gaussian

$$\max_{\mu, \Sigma \succ 0} \mathcal{L}(\mu, \Sigma) := \sum_{i=1}^{n} \log \mathcal{N}(x_i; \mu, \Sigma). \tag{17}$$

Although (17) is Euclidean convex, it is *not* g-convex. In [23] a simple reformulation[1] is used that makes (17) g-convex and ends up having far-reaching impact on the overall GMM problem. More precisely, we augment the sample vectors x_i to instead consider $y_i^T = [x_i^T \; 1]$. Therewith, problem (17) turns into

$$\max_{S \succ 0} \widehat{\mathcal{L}}(S) := \sum_{i=1}^{n} \log \widehat{\mathcal{N}}(y_i; S), \tag{18}$$

where $\widehat{\mathcal{N}}(y_i; S) := \sqrt{2\pi} \exp(\frac{1}{2}) \mathcal{N}(y_i; 0, S)$. Theorem 1 shows that (18) is g-convex and its solution yields the solution to the original problem (17).

Theorem 1 ([23]). *The map $-\widehat{\mathcal{L}}(S)$ is g-convex. Moreover, if μ^*, σ^* maximize* (17)*, and S^* maximizes* (18)*, then $\widehat{\mathcal{L}}(S^*) = \mathcal{L}(\mu^*, \Sigma^*)$ for $S^* = \begin{pmatrix} \Sigma^* + \mu^* \mu^{*T} & \mu^* \\ \mu^{*T} & 1 \end{pmatrix}$.*

Theorem 2 states a local version of this result for GMMs.

Theorem 2 ([23]). *A local maximum of the reparameterized GMM log-likelihood*

$$\widehat{\mathcal{L}}(\{S_j\}_{j=1}^K) := \sum_{i=1}^{n} \log\Big(\sum_{j=1}^{K} \alpha_j \widehat{\mathcal{N}}(y_i; S_j)\Big) \tag{19}$$

is a local maximum of the original log-likelihood (16).

3.1.2 Numerical Results

We solve the reparameterized problem (19)[2] using Alg. 1. We illustrate the performance through experiments on real and simulated data. All compared methods

[1] This reformulation essentially uses the "natural parameters"

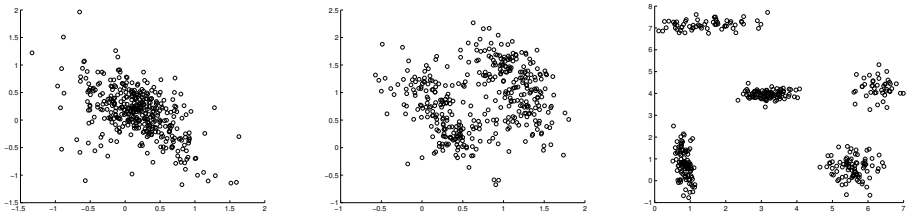
[2] Actually, we solve a slightly different *unconstrained* problem that also reparameterizes α_j.

are initialized using k-means++ [3], and all share the same stopping criterion. The methods stop when the difference of *average log-likelihood* (i.e., log-likelihood$/n$) between iterations falls below 10^{-6}, or when the iteration count exceeds 1500.

Since EM's performance depends on the degree of separation of the mixture components [32, 59], we also assess the impact of separation on our methods. We generate data as proposed in [19, 56]. The distributions are chosen so their means satisfy the following separation inequality:

$$\forall_{i\neq j} : \|\mu_i - \mu_j\| \geq c \max_{i,j}\{\mathrm{tr}(\Sigma_i), \mathrm{tr}(\Sigma_j)\}.$$

The parameter c shows level of separation; we use e to denote *eccentricity*, i.e., the ratio of the largest eigenvalue of the covariance matrix to its smallest eigenvalue. A typical 2D data with $K = 5$ created for different separations is shown in Figure 2.



(a) low separation (b) medium separation (c) high separation

Fig. 2 Data clouds for three levels of separation: $c = 0.2$ (low); $c = 1$ (medium); $c = 5$ (high).

We tested both high eccentricity ($e = 10$) and spherical ($e = 1$) Gaussians. Table 2 reports the results, which are obtained after running 20 different random initializations. Without our reformulation Riemannian optimization is not competitive (we omit the results), while with the reformulation our Riemannian LBFGS matches or exceeds EM. We note in passing that a Cholesky decomposition based formulation ends up being vastly inferior to both EM and our Riemannian methods. Numerical results supporting this claim may be found in [23].

Table 2 Speed and average log-likelihood (ALL) comparisons for $d = 20$, $e = 10$ and $e = 1$. The numbers are averaged values for 20 runs over different sampled datasets, therefore the ALL values are not comparable to each other. The standard-deviation are also reported in the table.

		EM ($e=10$)		LBFGS ($e=10$)		EM ($e=1$)		LBFGS ($e=1$)	
		Time (s)	ALL	Time (s)	ALL	Time (s)	ALL	Time (s)	ALL
$c=0.2$	$K=2$	1.1 ± 0.4	-10.7	5.6 ± 2.7	-10.7	65.7 ± 33.1	17.6	39.4 ± 19.3	17.6
	$K=5$	30.0 ± 45.5	-12.7	49.2 ± 35.0	-12.7	365.6 ± 138.8	17.5	160.9 ± 65.9	17.5
$c=1$	$K=2$	0.5 ± 0.2	-10.4	3.1 ± 0.8	-10.4	6.0 ± 7.1	17.0	12.9 ± 13.0	17.0
	$K=5$	104.1 ± 113.8	-13.4	79.9 ± 62.8	-13.3	40.5 ± 61.1	16.2	51.6 ± 39.5	16.2
$c=5$	$K=2$	0.2 ± 0.2	-11.0	3.4 ± 1.4	-11.0	0.2 ± 0.1	17.1	3.0 ± 0.5	17.1
	$K=5$	38.8 ± 65.8	-12.8	41.0 ± 45.7	-12.8	17.5 ± 45.6	16.1	20.6 ± 22.5	16.1

Table 3 Speed and ALL comparisons for natural image data $d = 35$.

	EM Algorithm		LBFGS Reformulated		CG Reformulated		CG Original		CG Cholesky Reformulated	
	Time (s)	ALL	Time (s)	ALL	Time (s)	ALL	Time (s)	ALL	Time (s)	ALL
$K = 2$	16.61	29.28	14.23	29.28	17.52	29.28	947.35	29.28	476.77	29.28
$K = 4$	165.77	31.65	106.53	31.65	153.94	31.65	6380.01	31.64	2673.21	31.65
$K = 8$	596.01	32.81	332.85	32.81	536.94	32.81	14282.80	32.58	9306.33	32.81
$K = 10$	2159.47	33.05	658.34	33.06	1048.00	33.06	17711.87	33.03	7463.72	33.05

Next, we present an evaluation on a natural image dataset, for which GMMs have been reported to be effective [64]. We extracted 200K image patches of size 6×6 from random locations in the images and subtracted the DC component. GMM fitting results obtained by different algorithms are reported in Table 3. As can be seen, manifold LBFGS performs better than EM and manifold CG. Moreover, our reformulation proves crucial: without it manifold optimization is substantially slower. The Cholesky-based model without our reformulation has the worst performance (not reported), and even with reformulation it is inferior to the other approaches.

Fig. 3 visualizes evolution of the objective function with the number of iterations (i.e., the number of log-likelihood and gradient evaluations, or the number of E- and M-steps). The datasets used in Fig. 3 are the 'magic telescope' and 'year prediction' datasets[3], as well as natural image data used in Table 2. It can be seen that although manifold optimization methods spend time doing line-search they catch up with EM algorithm in a few iterations.

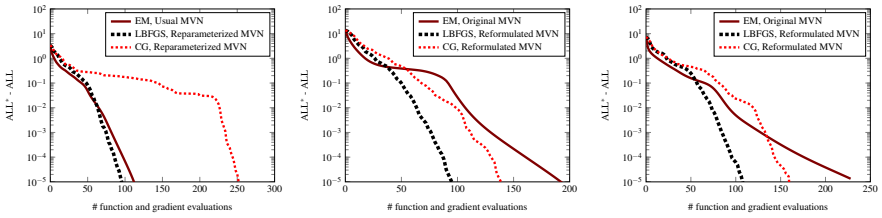


Fig. 3 The objective function with respect to the number of function and gradient evaluations. The objective function is the Best ALL minus current ALL values. Left: 'magic telescope' ($K = 5, d = 10$). Middle: 'year predict' ($K = 6, d = 90$). Right: natural images ($K = 8, d = 35$).

3.2 MLE for Elliptically Contoured Distributions

Our next application is to maximum likelihood parameter estimation for Kotz-type distributions. Here given i.i.d. samples $(x_1, \ldots, x_n)$ from an Elliptically Contoured Distribution $\mathscr{E}_\varphi(S)$, up to constants the log-likelihood is of the form

[3] Available at UCI machine learning dataset repository via https://archive.ics.uci.edu/ml/datasets

$$\mathscr{L}(x_1,\ldots,x_n;S) = -\tfrac{1}{2}n\log\det S + \sum_{i=1}^n \log\varphi(x_i^T S^{-1}x_i), \tag{20}$$

where φ is a so-called *density generating function* (dgf). We write $\Phi \equiv -\mathscr{L}$, so that computing the MLE amounts to minimizing Φ. But this is in general difficult: Φ can be nonconvex and may have multiple local minima. However, under suitable assumptions on φ, we can still maximize (20) to global optimality. Some examples are already known [26, 42, 63], and geometric optimization yields results that are more general than previously known examples. We refer the reader to [50] for the precise details, and provide a quick summary of the main ideas below.

The "suitable assumptions" alluded to above cover two main classes of dgfs:

(i) Geodesically convex (g-convex): This class contains functions for which the negative log-likelihood $\Phi(S)$ is g-convex. Some members of this class have been previously studied (possibly without exploiting g-convexity);
(ii) Log-Nonexpansive (LN): This class was introduced in [50]. It exploits the "non-positive curvature" property of the PD manifold and it covers several ECDs outside the scope of previous methods [26, 57, 63]. This class is essentially the same as what we call Thompson nonexpansive in this chapter.

In [50], the authors also discuss the class Log-Convex (LC), for which the dgf φ is log-convex, whereby Φ is nonconvex. But since Φ is now a difference of convex functions it is amenable to majorization-minimization.

Several examples of strictly g-convex ECDs are: (i) Multivariate Gaussian; (ii) Kotz with $\alpha \leq \frac{d}{2}$ (its special cases include Gaussian, multivariate power exponential, multivariate W-distribution with shape parameter smaller than one, elliptical gamma with shape parameter $\nu \leq \frac{d}{2}$); (iii) Multivariate-t; (iv) Multivariate Pearson type II with positive shape parameter; (v) Elliptical multivariate logistic distribution.

For the class LN, we can circumvent the machinery of manifold optimization and obtain simple fixed-point algorithms as alluded to in Sec. 1.2.

As an illustrative example, consider the problem of finding the minimum of negative log-likelihood solution of *Kotz-type distribution* (which is a particular ECD):

$$\Phi(S) = \tfrac{n}{2}\log\det(S) + (\tfrac{d}{2}-\alpha)\sum_{i=1}^n \log(x_i^T S^{-1}x_i) + \sum_{i=1}^n \left(\tfrac{x_i^T S^{-1}x_i}{b}\right)^{\beta}, \tag{21}$$

where α, β, and b are (known) fixed parameters. To minimize Φ, following Sec. 1.2, we seek to solve $\nabla\Phi(S) = 0$. This amounts to the nonlinear matrix equation

$$S = \tfrac{2}{n}\sum_{i=1}^n x_i h(x_i^T S^{-1}x_i)x_i^T, \tag{22}$$

where $h(\cdot) = (\frac{d}{2}-\alpha)(\cdot)^{-1} + \frac{\beta}{b^\beta}(\cdot)^{\beta-1}$. If (22) has positive definite solution, then it is a candidate MLE. If it is unique, then it is the desired minimum of (21).

The question now is whether upon setting $\mathscr{G} := \frac{2}{n}\sum_{i=1}^n x_i h(x_i^T S^{-1}x_i)x_i^T$ and simply iterating $S_{k+1} \leftarrow \mathscr{G}(S_k)$, we can obtain a solution to (22). This is where the theory developed in §1.2 comes into play. We mention below a slightly stronger result.

Let $\tau = 1-\beta$ and $c = \frac{b^\beta(d/2-\alpha)}{\beta}$. Knowing that $h(\cdot) = g((\frac{d}{2}-\alpha)(\cdot)^{-1})$ has the same contraction factor as $g(\cdot)$, it can be shown that h in the iteration (22) for Kotz-

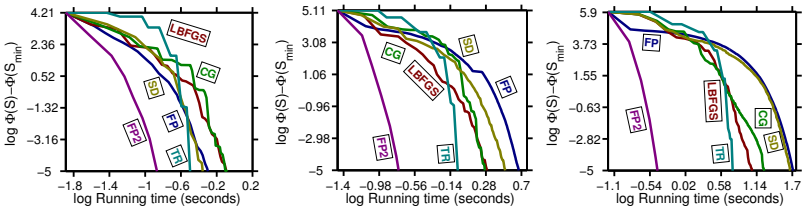


Fig. 4 Running times comparison between normal fixed-point iteration (FP), fixed-point iteration with scaling factor (FP2) and four different manifold optimization methods. The objective function is Kotz-type negative log-likelihood with parameters $\beta = 0.5$ and $\alpha = 1$. The plots show (from left to right), running times for estimating $S \in \mathbb{P}_d$, for $d \in \{4, 16, 64\}$.

type distributions for which $0 < \beta < 2$ and $\alpha < \frac{d}{2}$ is Thompson-contractive. Therewith, one can show the following convergence result.

Theorem 3 ([50]). *For Kotz-type distributions with* $0 < \beta < 2$ *and* $\alpha < \frac{d}{2}$*, Iteration* (22) *converges to a unique fixed point.*

3.2.1 Numerical Results

We compare now the convergence speed of fixed-point (FP) MLE iterations for different sets of parameters α and β. For our experiments, we sample 10,000 points from a Kotz-type distribution with a random scatter matrix and prescribed values of α and β. We compare the fixed-point approach with four different manifold optimization methods: (i) steepest descent (SD); (ii) conjugate gradient (CG); (iii) limited-memory RBFGS (LBFGS); (iv) trust-region (TR). All methods are initialized with the same random covariance matrix.

The first experiment (Fig. 4) fixes α, β, and shows the effect of dimension on convergence. Next, in Fig. 5, we fix dimension and consider the effect of varying α and β. As it is evident from the figures, FP and steepest descent method could have very slow convergence in some cases. FP2 denotes a re-scaled version of the basic fixed-point iteration FP (see [50] for details); the scaling improves conditioning and accelerates the method, leading to an overall best performance.

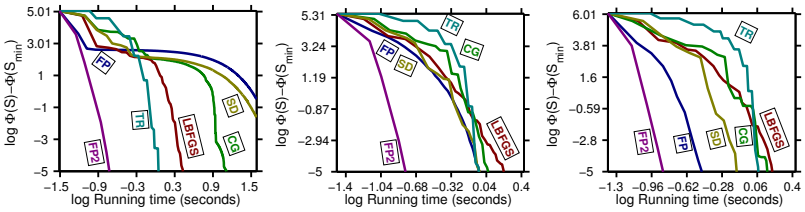


Fig. 5 Running time variance for Kotz-type distributions with $d = 16$ and $\alpha = 2\beta$ for different values of $\beta \in \{0.1, 1, 1.7\}$.

3.3 Other applications

To conclude we briefly mention below additional applications that rely on geometric optimization. Our listing is by no means complete, and is biased towards work more closely related to machine learning. However, it should provide a starting point for the interested reader in exploring other applications and aspects of the rapidly evolving area of geometric optimization.

Computer vision. Chapter **??** (see references therein) describes applications to image retrieval, dictionary learning, and other problems in computer vision that involve PD matrix data, and therefore directly or indirectly rely on geometric optimization.

Signal processing. Diffusion Tensor Imaging (DTI) [27]; Radar and signal processing [2, 41]; Brain Computer Interfaces (BCI) [60];

ML and Statistics. Social networks [46]; Deep learning [33, 35]; Determinantal point processes [24, 34]; Fitting elliptical gamma distributions [51]; Fitting mixture models [22, 37]; see also [11].

Others. Structured PD matrices [9]; Manifold optimization with rank constraints [55] and symmetries [38]. We also mention here two key theoretical references: general g-convex optimization [4], and wider mathematical background [13].

Acknowledgements SS acknowledges partial support from NSF grant IIS-1409802.

References

1. Absil, P.A., Mahony, R., Sepulchre, R.: Optimization algorithms on matrix manifolds. Princeton University Press (2009)
2. Arnaudon, M., Barbaresco, F., Yang, L.: Riemannian medians and means with applications to radar signal processing. Selected Topics in Signal Processing, IEEE Journal of **7**(4), 595–604 (2013)
3. Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (SODA), pp. 1027–1035 (2007)
4. Bacák, M.: Convex analysis and optimization in Hadamard spaces, vol. 22. Walter de Gruyter GmbH & Co KG (2014)
5. Bach, F., Jenatton, R., Mairal, J., Obozinski, G.: Optimization with sparsity-inducing penalties. Foundations and Trends® in Machine Learning **4**(1), 1–106 (2012)
6. Bhatia, R.: Positive Definite Matrices. Princeton University Press (2007)
7. Bhatia, R., Karandikar, R.L.: The matrix geometric mean. Tech. Rep. isid/ms/2-11/02, Indian Statistical Institute (2011)
8. Bini, D.A., Iannazzo, B.: Computing the Karcher mean of symmetric positive definite matrices. Linear Algebra and its Applications **438**(4), 1700–1710 (2013)
9. Bini, D.A., Iannazzo, B., Jeuris, B., Vandebril, R.: Geometric means of structured matrices. BIT Numerical Mathematics **54**(1), 55–83 (2014)
10. Bishop, C.M.: Pattern recognition and machine learning. Springer (2007)
11. Boumal, N.: Optimization and estimation on manifolds. Ph.D. thesis, Université catholique de Louvain (2014)
12. Boumal, N., Mishra, B., Absil, P.A., Sepulchre, R.: Manopt, a matlab toolbox for optimization on manifolds. The Journal of Machine Learning Research **15**(1), 1455–1459 (2014)

13. Bridson, M.R., Haefliger, A.: Metric spaces of non-positive curvature, vol. 319. Springer Science & Business Media (1999)
14. Burer, S., Monteiro, R.D., Zhang, Y.: Solving semidefinite programs via nonlinear programming. part i: Transformations and derivatives. Tech. Rep. TR99-17, Rice University, Houston TX (1999)
15. Chebbi, Z., Moahker, M.: Means of Hermitian positive-definite matrices based on the log-determinant α-divergence function. Linear Algebra and its Applications **436**, 1872–1889 (2012)
16. Cherian, A., Sra, S.: Riemannian dictionary learning and sparse coding for positive definite matrices. IEEE Transactions on Neural Networks and Learning Systems (2015). *Submitted*
17. Cherian, A., Sra, S.: Positive Definite Matrices: Data Representation and Applications to Computer Vision. In: Riemannian geometry in machine learning, statistics, optimization, and computer vision, Advances in Computer Vision and Pattern Recognition. Springer (2016). *this book*
18. Cherian, A., Sra, S., Banerjee, A., Papanikolopoulos, N.: Jensen-bregman logdet divergence for efficient similarity computations on positive definite tensors. IEEE Transactions Pattern Analysis and Machine Intelligence (2012)
19. Dasgupta, S.: Learning mixtures of gaussians. In: Foundations of Computer Science, 1999. 40th Annual Symposium on, pp. 634–644. IEEE (1999)
20. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B **39**, 1–38 (1977)
21. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. John Wiley & Sons (2000)
22. Hosseini, R., Mash'al, M.: Mixest: An estimation toolbox for mixture models. arXiv preprint arXiv:1507.06065 (2015)
23. Hosseini, R., Sra, S.: Matrix manifold optimization for Gaussian mixtures. In: Advances in Neural Information Processing Systems (NIPS) (2015). *To appear.*
24. Hough, J.B., Krishnapur, M., Peres, Y., Virág, B., et al.: Determinantal processes and independence. Probab. Surv **3**, 206–229 (2006)
25. Jeuris, B., Vandebril, R., Vandereycken, B.: A survey and comparison of contemporary algorithms for computing the matrix geometric mean. Electronic Transactions on Numerical Analysis **39**, 379–402 (2012)
26. Kent, J.T., Tyler, D.E.: Redescending M-estimates of multivariate location and scatter. The Annals of Statistics **19**(4), 2102–2119 (1991)
27. Le Bihan, D., Mangin, J.F., Poupon, C., Clark, C.A., Pappata, S., Molko, N., Chabriat, H.: Diffusion tensor imaging: concepts and applications. Journal of magnetic resonance imaging **13**(4), 534–546 (2001)
28. Lee, H., Lim, Y.: Invariant metrics, contractions and nonlinear matrix equations. Nonlinearity **21**, 857–878 (2008)
29. Lee, J.M.: Introduction to Smooth Manifolds. No. 218 in GTM. Springer (2012)
30. Lemmens, B., Nussbaum, R.: Nonlinear Perron-Frobenius Theory. Cambridge University Press (2012)
31. Lim, Y., Pálfia, M.: Matrix power means and the Karcher mean. Journal of Functional Analysis **262**, 1498–1514 (2012)
32. Ma, J., Xu, L., Jordan, M.I.: Asymptotic convergence rate of the em algorithm for gaussian mixtures. Neural Computation **12**(12), 2881–2907 (2000)
33. Mariet, Z., Sra, S.: Diversity networks. arXiv:1511.05077 (2015)
34. Mariet, Z., Sra, S.: Fixed-point algorithms for learning determinantal point processes. In: International Conference on Machine Learning (ICML) (2015)
35. Masci, J., Boscaini, D., Bronstein, M.M., Vandergheynst, P.: ShapeNet: Convolutional Neural Networks on Non-Euclidean Manifolds. arXiv preprint arXiv:1501.06297 (2015)
36. McLachlan, G.J., Peel, D.: Finite mixture models. John Wiley and Sons, New Jersey (2000)
37. Mehrjou, A., Hosseini, R., Araabi, B.N.: Mixture of ICAs model for natural images solved by manifold optimization method. In: 7th International Conference on Information and Knowledge Technology (2015)

38. Mishra, B.: A riemannian approach to large-scale constrained least-squares with symmetries. Ph.D. thesis, Université de Namur (2014)
39. Moakher, M.: A differential geometric approach to the geometric mean of symmetric positive-definite matrices. SIAM Journal on Matrix Anal. Appl. (SIMAX) **26**, 735–747 (2005)
40. Murphy, K.P.: Machine Learning: A Probabilistic Perspective. MIT Press (2012)
41. Nielsen, F., Bhatia, R. (eds.): Matrix Information Geometry. Springer (2013)
42. Ollila, E., Tyler, D., Koivunen, V., Poor, H.V.: Complex elliptically symmetric distributions: Survey, new results and applications. IEEE Transactions on Signal Processing **60**(11), 5597–5625 (2011)
43. Redner, R.A., Walker, H.F.: Mixture densities, maximum likelihood, and the EM algorithm. Siam Review **26**, 195–239 (1984)
44. Ring, W., Wirth, B.: Optimization methods on riemannian manifolds and their application to shape space. SIAM Journal on Optimization **22**(2), 596–627 (2012)
45. Schölkopf, B., Smola, A.J.: Learning with Kernels. MIT Press, Cambridge, MA (2002)
46. Shrivastava, A., Li, P.: A new space for comparing graphs. In: Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on, pp. 62–71. IEEE (2014)
47. Sra, S.: On the matrix square root and geometric optimization. arXiv:1507.08366 (2015)
48. Sra, S.: Positive Definite Matrices and the S-Divergence. Proceedings of the American Mathematical Society (2015). *also* arXiv:1110.1773v4
49. Sra, S., Hosseini, R.: Geometric optimisation on positive definite matrices for elliptically contoured distributions. In: Advances in Neural Information Processing Systems, pp. 2562–2570 (2013)
50. Sra, S., Hosseini, R.: Conic geometric optimisation on the manifold of positive definite matrices. SIAM Journal on Optimization **25**(1), 713–739 (2015)
51. Sra, S., Hosseini, R., Theis, L., Bethge, M.: Data modeling with the elliptical gamma distribution. In: Artificial Intelligence and Statistics (AISTATS), vol. 18 (2015)
52. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) pp. 267–288 (1996)
53. Udrişte, C.: Convex functions and optimization methods on Riemannian manifolds. Kluwer (1994)
54. Vanderbei, R.J., Benson, H.Y.: On formulating semidefinite programming problems as smooth convex nonlinear optimization problems. Tech. rep., Princeton (2000)
55. Vandereycken, B.: Riemannian and multilevel optimization for rank-constrained matrix problems. Ph.D. thesis, Department of Computer Science, KU Leuven (2010)
56. Verbeek, J.J., Vlassis, N., Kröse, B.: Efficient greedy learning of gaussian mixture models. Neural computation **15**(2), 469–485 (2003)
57. Wiesel, A.: Geodesic convexity and covariance estimation. IEEE Transactions on Signal Processing **60**(12), 6182–89 (2012)
58. Wiesel, A.: Unified framework to regularized covariance estimation in scaled Gaussian models. IEEE Transactions on Signal Processing **60**(1), 29–38 (2012)
59. Xu, L., Jordan, M.I.: On convergence properties of the EM algorithm for Gaussian mixtures. Neural Computation **8**, 129–151 (1996)
60. Yger, F.: A review of kernels on covariance matrices for BCI applications. In: Machine Learning for Signal Processing (MLSP), 2013 IEEE International Workshop on, pp. 1–6. IEEE (2013)
61. Zhang, J., Wang, L., Zhou, L., Li, W.: Learning discriminative stein kernel for spd matrices and its applications. arXiv preprint arXiv:1407.1974 (2014)
62. Zhang, T.: Robust subspace recovery by geodesically convex optimization. arXiv preprint arXiv:1206.1386 (2012)
63. Zhang, T., Wiesel, A., Greco, S.: Multivariate generalized Gaussian distribution: Convexity and graphical models. IEEE Transaction on Signal Processing **60**(11), 5597–5625 (2013)
64. Zoran, D., Weiss, Y.: Natural images, gaussian mixtures and dead leaves. In: Advances in Neural Information Processing Systems, pp. 1736–1744 (2012)