

# Riemannian Online Algorithms for Estimating Mixture Model Parameters

Paolo Zanini<sup>1,2</sup>, Salem Said<sup>1,2(✉)</sup>, Yannick Berthoumiou<sup>1,2</sup>, Marco Congedo<sup>1,2</sup>,  
and Christian Jutten<sup>1,2</sup>

<sup>1</sup> Laboratoire IMS (CNRS - UMR 5218), Université de Bordeaux, Bordeaux, France  
{salem.said,yannick.berthoumiou}@ims-bordeaux.fr

<sup>2</sup> Gipsa-lab (CNRS - UMR 5216), Université de Grenoble, Grenoble, France  
{paolo.zanini,marco.congedo,christian.jutten}@gipsa-lab.fr

**Abstract.** This paper introduces a novel algorithm for the online estimate of the Riemannian mixture model parameters. This new approach counts on Riemannian geometry concepts to extend the well-known Titterington approach for the online estimate of mixture model parameters in the Euclidean case to the Riemannian manifolds. Here, Riemannian mixtures in the Riemannian manifold of Symmetric Positive Definite (SPD) matrices are analyzed in details, even if the method is well suited for other manifolds.

**Keywords:** Riemannian mixture estimation · Information geometry · Online EM algorithm

## 1 Introduction

Information theory and Riemannian geometry have been widely developed in the recent years in a lot of different applications. In particular, Symmetric Positive Definite (SPD) matrices have been deeply studied through Riemannian geometry tools. Indeed, the space  $\mathcal{P}_m$  of  $m \times m$  SPD matrices can be equipped with a Riemannian metric. This metric, usually called Rao-Fisher or affine-invariant metric, gives it the structure of a Riemannian manifold (specifically a homogeneous space of non-positive curvature). SPD matrices are of great interest in several applications, like diffusion tensor imaging, brain-computer interface, radar signal processing, mechanics, computer vision and image processing [1–5]. Hence, it is very useful to develop statistical tools to analyze objects living in the manifold  $\mathcal{P}_m$ . In this paper we focus on the study of Mixtures of Riemannian Gaussian distributions, as defined in [6]. They have been successfully used to define probabilistic classifiers in the classification of texture images [7] or Electroencephalography (EEG) data [8]. In these examples mixtures parameters are estimated through suitable EM algorithms for Riemannian manifolds. In this paper we consider a particular situation, that is the observations are observed one at a time. Hence, an online estimation of the parameters is needed. Following the Titterington’s approach [9], we derive a novel approach for the online estimate of parameters of Riemannian Mixture distributions.

The paper is structured as follows. In Sect. 2 we describe the Riemannian Gaussian Mixture Model. In Sect. 3, we introduce the reference methods for online estimate of mixture parameters in the Euclidean case, and we describe in details our approach for the Riemannian framework. For lack of space, some equation’s proofs will be omitted. Then, in Sect. 4, we present some simulations to validate the proposed method. Finally we conclude with some remarks and future perspectives in Sect. 5.

## 2 Riemannian Gaussian Mixture Model

We consider a Riemannian Gaussian Mixture model  $g(x; \theta) = \sum_{k=1}^K \omega_k p(x; \psi_k)$ , with the constraint  $\sum_{k=1}^K \omega_k = 1$ . Here  $p(x; \psi_k)$  is the Riemannian Gaussian distribution studied in [6], defined as  $p(x; \psi_k) = \frac{1}{\zeta(\sigma_k)} \exp\left(-\frac{d_R^2(x, \bar{x}_k)}{2\sigma_k^2}\right)$ , where  $x$  is a SPD matrix,  $\bar{x}_k$  is still a SPD matrix representing the center of mass of the  $k$ th component of the mixture,  $\sigma_k$  is a positive number representing the dispersion parameter of the  $k$ th mixture component,  $\zeta(\sigma_k)$  is the normalization factor, and  $d_R(\cdot, \cdot)$  is the Riemannian distance induced by the metric on  $\mathcal{P}_m$ .  $g(x; \theta)$  is also called incomplete likelihood. In the typical mixture model approach, indeed, we consider some latent variables  $Z_i$ , categorical variables over  $\{1, \dots, K\}$  with parameters  $\{\omega_k\}_{k=1}^K$ , assuming  $X_i | Z_i = k \sim p(\cdot, \psi_k)$ . Thus, the complete likelihood is defined as  $f(x, z; \theta) = \sum_{k=1}^K \omega_k p(x; \psi_k) \delta_{z,k}$ , where  $\delta_{z,k} = 1$  if  $z = k$  and 0 otherwise. We deal here with the problem to estimate the model parameters, gathered in the vector  $\theta = [\omega_1, \bar{x}_1, \sigma_1, \dots, \omega_K, \bar{x}_K, \sigma_K]$ . Usually, given a set of  $N$  i.i.d. observations  $\chi = \{x_i\}_{i=1}^N$ , we look for  $\hat{\theta}_N^{MLE}$ , that is the MLE of  $\theta$ , i.e. the maximizer of the log-likelihood  $l(\theta; \chi) = \frac{1}{N} \sum_{i=1}^N \log \sum_{k=1}^K \omega_k p(x_i; \psi_k)$ .

To obtain  $\hat{\theta}_N^{MLE}$ , EM or stochastic EM approaches are used, based on the complete dataset  $\chi_c = \{(x_i, z_i)\}_{i=1}^N$ , with the unobserved variables  $Z_i$ . In this case, average complete log-likelihood can be written:

$$l_c(\theta; \chi_c) = \frac{1}{N} \sum_{i=1}^N \log \prod_{k=1}^K (\omega_k p(x_i; \psi_k))^{\delta_{z_i,k}} = \frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K \delta_{z_i,k} \log(\omega_k p(x_i; \psi_k)). \quad (1)$$

Here we consider a different situation, that is the dataset  $\chi$  is not available entirely, rather the observations are observed one at a time. In this situation online estimation algorithms are needed.

## 3 Online Estimation

In the Euclidean case, reference algorithms are the Titterington’s algorithm, introduced in [9], and the Cappé-Moulines’s algorithm presented in [10].

We focus here on Titterington’s approach. In classic EM algorithms, the Expectation step consists in computing  $Q(\theta; \hat{\theta}^{(r)}, \chi) = E_{\hat{\theta}^{(r)}}[l_c(\theta; \chi_c) | \chi]$ , and

then, in the Maximization step, in maximizing  $Q$  over  $\theta$ . These steps are performed iteratively and at each iteration  $r$  an estimate  $\hat{\theta}^{(r)}$  of  $\theta$  is obtained exploiting the whole dataset. In the online framework, instead, the current estimate will be indicated by  $\hat{\theta}^{(N)}$ , since in this setting, once  $x_1, x_2, \dots, x_N$  are observed we want to update our estimate for a new observation  $x_{N+1}$ . Titterington approach corresponds to the direct optimization of  $Q(\theta; \hat{\theta}^{(N)}, \chi)$  using a Newton algorithm:

$$\hat{\theta}^{(N+1)} = \hat{\theta}^{(N)} + \gamma^{(N+1)} I_c^{-1}(\hat{\theta}^{(N)}) u(x_{N+1}; \hat{\theta}^{(N)}), \tag{2}$$

where  $\{\gamma^{(N)}\}_N$  is a decreasing sequence, the Hessian of  $Q$  is approximated by the Fisher Information matrix  $I_c$  for the complete data  $I_c^{-1}(\hat{\theta}^{(N)}) = -E_{\hat{\theta}^{(N)}}[\frac{\log f(x, z; \theta)}{\partial \theta \partial \theta^T}]$ , and the score  $u(x_{N+1}; \hat{\theta}^{(N)})$  is defined as  $u(x_{N+1}; \hat{\theta}^{(N)}) = \nabla_{\hat{\theta}^{(N)}} \log g(x_{N+1}; \hat{\theta}^{(N)}) = E_{\hat{\theta}^{(N)}}[\nabla_{\hat{\theta}^{(N)}} \log f(x_{N+1}; \hat{\theta}^{(N)}) | x_{N+1}]$  (where last equality is presented in [10]).

Geometrically speaking, Tittetington algorithm consists in modifying the current estimate  $\hat{\theta}^{(N+1)}$  adding the term  $\xi^{(N+1)} = \gamma^{(N+1)} I_c^{-1}(\hat{\theta}^{(N)}) u(x_{N+1}; \hat{\theta}^{(N)})$ . If we want to consider parameters belonging to Riemannian manifolds, we have to suitably modify the update rule. Furthermore, even in the classical framework, Titterington update does not necessarily constraint the estimates to be in the parameters space. For instance, the weights could be assume negative values. The approach we are going to introduce solves this problem, and furthermore is suitable for Riemannian Mixtures.

We modify the update rule, exploiting the Exponential map. That is:

$$\hat{\theta}^{(N+1)} = \text{Exp}_{\hat{\theta}^{(N)}}(\xi^{(N+1)}), \tag{3}$$

where our parameters become  $\theta_k = [s_k, \bar{x}_k, \eta_k]$ . Specifically,  $s_k^2 = w_k \rightarrow s = [s_1, \dots, s_K] \in \mathbb{S}^{K-1}$  (i.e., the sphere),  $\bar{x}_k \in P(m)$  and  $\eta_k = -\frac{1}{2\sigma_k^2} < 0$ .

Actually we are not forced to choose the exponential map, in the update formula (3), but we can consider any retraction operator. Thus, we can generalize (3) in  $\hat{\theta}^{(N+1)} = \mathcal{R}_{\hat{\theta}^{(N)}}(\xi^{(N+1)})$ .

In order to develop a suitable update rule, we have to define  $I(\theta)$  and the score  $u()$  in the manifold, noting that every parameter belongs to a different manifold. Firstly we note that the Fisher Information matrix  $I(\theta)$  can be written as:

$$I(\theta) = \begin{pmatrix} I(s) & & \\ & I(\bar{x}) & \\ & & I(\eta) \end{pmatrix}.$$

Now we can analyze separately the update rule for  $s$ ,  $\bar{x}$ , and  $\eta$ . Since they belong to different manifold the exponential map (or the retraction) will be different, but the philosophy of the algorithm is still the same.

For the update of weights  $s_k$ , the Riemannian manifold considered is the sphere  $\mathbb{S}^{K-1}$ , and, given a point  $s \in \mathbb{S}^{K-1}$ , the tangent space  $T_s \mathbb{S}^{K-1}$  is identified as  $T_s \mathbb{S}^{K-1} = \{\xi \in \mathbb{R}^K : \xi^T s = 0\}$ . We can write the complete log-likelihood only

in terms of  $s$ :  $l(x, z; s) = \log f(x, z; s) = \sum_{k=1}^K \log s_k^2 \delta_{z,k}$ . We start by evaluating  $I(s)$ , that will be a  $K \times K$  matrix of the quadratic form

$$I_s(u, w) = E[\langle u, v(z, s) \rangle \langle v(z, s), w \rangle], \tag{4}$$

for  $u, w$  elements of the tangent space in  $s$ , and  $v(z, s)$  is the Riemannian gradient, defined as  $v(z, s) = \frac{\partial l}{\partial s} - \left(\frac{\partial l}{\partial s}, s\right) s$ . In this case we obtain  $\frac{\partial l}{\partial s_k} = 2 \frac{\delta_{z,k}}{s_k} \rightarrow v(z, s_k) = 2 \left(\frac{\delta_{z,k}}{s_k} - s_k\right)$ . It is easy to see that the matrix of the quadratic form has elements

$$\begin{aligned} I_{kl}(s) &= E[v_k(z, s)v_l(z, s)] = E\left[4 \left(\frac{\delta_{z,k}}{s_k} - s_k\right) \left(\frac{\delta_{z,l}}{s_l} - s_l\right)\right] \\ &= E\left[4 \left(\frac{\delta_{z,k}\delta_{z,l}}{s_k s_l} - \frac{s_l}{s_k} \delta_{z,k} - \frac{s_k}{s_l} \delta_{z,l} + s_k s_l\right)\right] = 4(\delta_{kl} - s_k s_l - s_k s_l + s_k s_l) = 4(\delta_{kl} - s_k s_l). \end{aligned}$$

Thus, the Fisher Information matrix  $I(s)$  applied to an element  $\xi$  of the tangent space results to be  $I(s)\xi = 4\xi$ , hence  $I(s)$  corresponds to 4 times the identity matrix. Thus, if we consider update rule (3), we have  $\xi^{(N+1)} = \frac{\gamma^{(N+1)}}{4} u(x_{N+1}; \hat{\theta}^{(N)})$ . We have to evaluate  $u(x_{N+1}; \hat{\theta}^{(N)})$ . We proceed as follows:

$$u_k(x_{N+1}; \hat{\theta}^{(N)}) = E[v_k(z, s)|x_{N+1}] = E\left[2 \left(\frac{\delta_{z,k}}{s_k} - s_k\right) |x_{N+1}\right] = 2 \left(\frac{h_k(x_{N+1}; \hat{\theta}^{(N)})}{s_k} - s_k\right),$$

where  $h_k(x_{N+1}; \hat{\theta}^{(N)}) \propto s_k^2 p(x_{N+1}; \hat{\theta}_k^{(N)})$ . Thus we obtain

$$\begin{aligned} \hat{s}^{(N+1)} &= \text{Exp}_{\hat{s}^{(N)}} \left( \frac{\gamma^{(N+1)}}{2} \left( \frac{h_1(x_{N+1}; \hat{\theta}^{(N)})}{\hat{s}_1^{(N)}} - \hat{s}_1^{(N)}, \dots, \frac{h_K(x_{N+1}; \hat{\theta}^{(N)})}{\hat{s}_K^{(N)}} - \hat{s}_K^{(N)} \right) \right) \\ &= \text{Exp}_{\hat{s}^{(N)}} (\xi^{(N+1)}). \end{aligned} \tag{5}$$

Considering the classical exponential map on the sphere (i.e., the geodesic), the update rule (5) becomes

$$\hat{s}_k^{(N+1)} = \hat{s}_k^{(N)} \cos(\|\xi^{(N+1)}\|) + \frac{\frac{\gamma^{(N+1)}}{2} \left(\frac{h_k}{\hat{s}_k^{(N)}} - \hat{s}_k^{(N)}\right)}{\|\xi^{(N+1)}\|} \sin(\|\xi^{(N+1)}\|). \tag{6}$$

Actually, as anticipated before, we are not forced to use the exponential map, but we can consider other retractions. In particular, on the sphere, we could consider the ‘‘projection’’ retraction  $\mathcal{R}_x(\xi) = \frac{x+\xi}{\|x+\xi\|}$ , deriving update rule accordingly.

For the update of barycenters  $\bar{x}_k$  we have, for every barycenter  $\bar{x}_k, k = 1, \dots, K$ , an element of  $\mathcal{P}_m$ , the Riemannian manifold of  $m \times m$  SPD matrices. Thus, we derive the update rule for a single  $k$ .

First of all we have to derive expression (4). But this expression is true only for irreducible manifolds, as the sphere. In the case of  $\mathcal{P}_m$  we have to introduce some theoretical results. Let  $\mathcal{M}$  a symmetric space of negative curvature (like  $\mathcal{P}_m$ ), it can be expressed as a product  $\mathcal{M} = \mathcal{M}_1 \times \dots \times \mathcal{M}_R$ , where each  $\mathcal{M}_r$

is an irreducible space [11]. Now let  $\bar{x}$  an element of  $\mathcal{M}$ , and  $v, w$  elements of the tangent space  $T_{\bar{x}}\mathcal{M}$ . We can write  $\bar{x} = (\bar{x}_1, \dots, \bar{x}_R)$ ,  $v = (v_1, \dots, v_R)$  and  $w = (w_1, \dots, w_R)$ . We can generalize (4) by the following expression:

$$I_{\bar{x}}(u, w) = \sum_{r=1}^R E[\langle u_r, v_r(\bar{x}_r) \rangle_{\bar{x}} \langle v_r(\bar{x}_r), w_r \rangle_{\bar{x}}], \tag{7}$$

with  $v_r(\bar{x}_r) = \nabla_{\bar{x}} l(\bar{x})$  being the Riemannian score. In our case  $\mathcal{P}_m = \mathbb{R} \times \mathcal{SP}_m$ , where  $\mathcal{SP}_m$  represents the manifold of SPD matrices with unitary determinant, while  $\mathbb{R}$  takes into account the part relative to the determinant. Thus, if  $x \in \mathcal{P}_m$ , we can consider the isomorphism  $\phi(x) = (x_1, x_2)$  with  $x_1 = \log \det x \in \mathbb{R}$  and  $x_2 = e^{-x_1/m} x \in \mathcal{SP}_m$ , ( $\det x_2 = 1$ ). The idea is to use the procedure adopted to derive  $\widehat{\sigma}^{(N+1)}$ , for each component of  $\widehat{\bar{x}}_k^{(N+1)}$ . Specifically we proceed as follows:

- we derive  $I(\bar{x}_k)$  through formula (7), with components  $I_r$ .
- we derive the Riemannian score  $u(x_{N+1}; \widehat{\theta}^{(N)}) = E \left[ v(x_{N+1}, z_{N+1}; \widehat{\bar{x}}_k^{(N)}, \widehat{\sigma}_k^{(N)}) | x_{N+1} \right]$ , with components  $u_r$ .
- for each component  $r = 1, 2$  we evaluate  $\xi_r^{(N+1)} = \gamma^{(N+1)} I_r^{-1} u_r$
- we update each component  $\left( \widehat{\bar{x}}_k^{(N+1)} \right)_r = \text{Exp}_{\left( \widehat{\bar{x}}_k^{(N)} \right)_r} \left( \xi_r^{(N+1)} \right)$  and we could use  $\phi^{-1}(\cdot)$  to derive  $\widehat{\bar{x}}_k^{(N+1)}$  if needed.

We start deriving  $I(\bar{x}_k)$  for the complete model (see [12] for some derivations):

$$\begin{aligned} I_{\bar{x}_k}(u, w) &= E[\langle u, v(x, z; \bar{x}_k, \sigma_k) \rangle \langle v(x, z; \bar{x}_k, \sigma_k), w \rangle] = E \left[ \frac{\delta_{z,k}}{\sigma_k^4} \langle u, \text{Log}_{\bar{x}_k} x \rangle \langle \text{Log}_{\bar{x}_k} x, w \rangle \right] = \\ &= E \left[ \frac{\delta_{z,k}}{\sigma_k^4} I(u, w) \right] = \frac{\omega_k}{\sigma_k^4} \sum_{r=1}^2 \frac{\psi'_r(\eta_k)}{\dim(\mathcal{M}_r)} \langle u_r, w_r \rangle_{(\bar{x}_k)_r}, \end{aligned} \tag{8}$$

where  $\psi(\eta_k) = \log \zeta$  as a function of  $\eta_k = -\frac{1}{2\sigma_k^2}$ , and we have the result introduced in [13] that says that if  $x \in \mathcal{M}$  is distributed with a Riemannian Gaussian distribution on  $\mathcal{M}$ ,  $x_r$  is distributed as a Riemannian Gaussian distribution on  $\mathcal{M}_r$  and  $\zeta(\sigma_k) = \prod_{r=1}^R \zeta_r(\sigma_k)$ . In our case  $\zeta_1(\sigma_k) = \sqrt{2\pi m \sigma_k^2}$  ( $\psi_1(\eta_k) = \frac{1}{2} \log(-\frac{\pi m}{\eta_k})$ ), and then we obtain  $\zeta_2(\sigma_k) = \frac{\zeta(\sigma_k)}{\zeta_1(\sigma_k)}$  easily, since  $\zeta(\sigma_k)$  has been derived in [6, 8]. From (8), we observe that for both components  $r = 1, 2$  the Fisher Information matrix is proportional to the identity matrix with a coefficient  $\frac{\omega_k}{\sigma_k^4} \frac{\psi'_r(\eta_k)}{\dim(\mathcal{M}_r)}$ .

We derive now the Riemannian score  $u(x_{N+1}; \widehat{\theta}_k^{(N)}) \in T_{\widehat{\bar{x}}_k^{(N)}} P(m)$ :

$$u(x_{N+1}; \widehat{\theta}_k^{(N)}) = E \left[ v(x, z; \widehat{\bar{x}}_k^{(N)}, \widehat{\sigma}_k^{(N)}) | x_{N+1} \right] = \frac{h_k(x_{N+1}; \widehat{\theta}_k^{(N)})}{\widehat{\sigma}_k^{2(N)}} \text{Log}_{\widehat{\bar{x}}_k^{(N)}} x_{N+1}.$$

In order to find  $u_1$  and  $u_2$  we have simply to apply the Logarithmic map of Riemannian manifold  $\mathcal{M}_1$  and  $\mathcal{M}_2$ , which in our case are  $\mathbb{R}$  and  $\mathcal{SP}_m$ , respectively, to the component 1 and 2 of  $x_{N+1}$  and  $\widehat{x}_k^{(N)}$ :

$$u_1 = \frac{h_k(x_{N+1}; \widehat{\theta}^{(N)})}{\widehat{\sigma}_k^{2(N)}} \left( (\widehat{x}_k^{(N)})_1 - (x_{N+1})_1 \right)$$

$$u_2 = \frac{h_k(x_{N+1}; \widehat{\theta}^{(N)})}{\widehat{\sigma}_k^{2(N)}} (\widehat{x}_k^{(N)})_2^{1/2} \log \left( (\widehat{x}_k^{(N)})_2^{-1/2} (x_{N+1})_2 (\widehat{x}_k^{(N)})_2^{-1/2} \right) (\widehat{x}_k^{(N)})_2^{1/2}$$

Expliciting  $\psi'_r(\eta_k)$ , specifically  $\psi'_1(\eta_k) = -\frac{1}{2\eta_k} = \sigma_k^2$  and  $\psi'_2(\eta_k) = \psi'(\eta_k) + \frac{1}{2\eta_k}$ , we can easily apply the Fisher Information matrix to  $u_r$ . In this way we can derive  $\xi_1^{(N+1)} = \gamma^{(N+1)} I_1^{-1}(\widehat{\theta}^{(N)}) u_1$  and  $\xi_2^{(N+1)} = \gamma^{(N+1)} I_2^{-1}(\widehat{\theta}^{(N)}) u_2$ . We are now able to obtain the update rules through the respective exponential maps:

$$\left( \widehat{x}_k^{(N+1)} \right)_1 = \left( \widehat{x}_k^{(N)} \right)_1 - \xi_1^{(N+1)} \tag{9}$$

$$\left( \widehat{x}_k^{(N+1)} \right)_2 = \left( \widehat{x}_k^{(N)} \right)_2^{1/2} \exp \left( \left( \widehat{x}_k^{(N)} \right)_2^{-1/2} \xi_2^{(N+1)} \left( \widehat{x}_k^{(N)} \right)_2^{-1/2} \right) \left( \widehat{x}_k^{(N)} \right)_2^{1/2} \tag{10}$$

For the update of dispersion parameters  $\sigma_k$ , we consider  $\eta_k = -\frac{1}{2\sigma_k^2}$ . Thus, we consider a real parameter, and then our calculus will be done in the classical Euclidean framework. First of all we have  $l(x, z; \eta_k) = \log f(x, z; \eta_k) = \sum_{k=1}^K \delta_{z,k} (-\psi(\eta_k) + \eta_k d_R^2(x, \bar{x}_k))$ . Thus, we can derive  $v(x, z; \eta_k) = \frac{\partial l}{\partial \eta_k} = \delta_{z,k} (-\psi'(\eta_k) + d_R^2(x, \bar{x}_k))$ . Knowing that  $I(\eta_k) = \omega_k \psi''(\eta_k)$ , we can evaluate the score:

$$u(x_{N+1}; \widehat{\theta}^{(N)}) = E[v(x, z; \eta_k) | x_{N+1}] = h_k(x_{N+1}; \widehat{\theta}^{(N)}) \left( d_R^2(x_{N+1}, \widehat{x}_k^{(N)}) - \psi'(\widehat{\eta}_k^{(N)}) \right). \tag{11}$$

Hence we can obtain the updated formula for the dispersion parameter

$$\widehat{\eta}_k^{(N+1)} = \widehat{\eta}_k^{(N)} + \gamma^{(N+1)} \frac{h_k(x_{N+1}; \widehat{\theta}^{(N)})}{\widehat{\omega}_k^{(N)} \psi''(\widehat{\eta}_k^{(N)})} \left( d_R^2(x_{N+1}, \widehat{x}_k^{(N)}) - \psi'(\widehat{\eta}_k^{(N)}) \right), \tag{12}$$

and, obviously  $\widehat{\sigma}_k^{2(N+1)} = -\frac{1}{2\widehat{\eta}_k^{(N+1)}}$ .

### 4 Simulations

We consider here two simulation frameworks to test the algorithm described in this paper.

The first framework corresponds to the easiest case. Indeed we consider only one mixture component (i.e.,  $K = 1$ ). Thus, this corresponds to a simple online mean and dispersion parameter estimate for a Riemannian Gaussian sample.

We consider matrices in  $\mathcal{P}_3$  and we analyze three different simulations corresponding to three different value of the barycenter  $\bar{x}_1$ :

$$\bar{x}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} ; \bar{x}_1 = \begin{pmatrix} 1 & 0.8 & 0.64 \\ 0.8 & 1 & 0.8 \\ 0.64 & 0.8 & 1 \end{pmatrix} ; \bar{x}_1 = \begin{pmatrix} 1 & 0.3 & 0.09 \\ 0.3 & 1 & 0.3 \\ 0.09 & 0.3 & 1 \end{pmatrix}$$

The value of dispersion parameter  $\sigma$  is taken equal to 0.1 for the three simulations. We analyze different initial estimates  $\hat{\theta}_{in}$ , closer to the true values at the beginning, and further at the end. We focus only on the barycenter, while the initial estimate for  $\sigma$  corresponds to the true value. We consider two different initial values for each simulation. Specifically for case (a),  $d_R(\bar{x}_1, \hat{x}_1^{(0)})$  is lower, varying between 0.11 and 0.14. For case (b) it is greater, varying between 1.03 and 1.16. For every simulation we generate  $N_{rep} = 100$  samples, each one of  $N = 100$  observations. Thus at the end we obtain  $N_{rep}$  different estimates  $(\hat{x}_{1r}, \hat{\sigma}_r)$  for every simulation and we can evaluate the mean  $m$  and standard deviation  $s$  of the error, where the error is measured as the Riemannian distance between  $\hat{x}_{1r}$  and  $\bar{x}_1$  for the barycenter, and as  $|\sigma - \hat{\sigma}|$  for the dispersion parameter. The results are summarized in Table 1.

**Table 1.** Mean and standard deviation of the error for the first framework

Simulation	$m_{\bar{x}_1}$	$s_{\bar{x}_1}$	$m_\sigma$	$s_\sigma$
1	0.0308	0.0092	0.0097	0.0556
2	0.0309	0.0098	0.0117	0.0570
3	0.0308	0.0096	0.0047	0.0051

In the second framework we consider the mixture case, in particular  $K = 2$ . The true weight are 0.4 and 0.6, while  $\sigma_1 = \sigma_2 = 0.1$ . The true barycenters are:

$$\bar{x}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \bar{x}_2 = \begin{pmatrix} 1 & 0.7 & 0.49 \\ 0.7 & 1 & 0.7 \\ 0.49 & 0.7 & 1 \end{pmatrix}$$

We make the initial estimates varying from the true barycenters to some SPD different from the true ones. In particular we analyze three cases. Case (a), where  $d_R(\bar{x}_1, \hat{x}_1^{(0)}) = d_R(\bar{x}_2, \hat{x}_2^{(0)}) = 0$ ; case (b), where  $d_R(\bar{x}_1, \hat{x}_1^{(0)}) = 0.2$  and  $d_R(\bar{x}_2, \hat{x}_2^{(0)}) = 0.26$ ; case (c), where  $d_R(\bar{x}_1, \hat{x}_1^{(0)}) = d_R(\bar{x}_2, \hat{x}_2^{(0)}) = 0.99$ . The results obtained are shown in Table 2. In both frameworks it is clear that we can obtain very good results when starting close to the real parameter values, while the goodness of the estimates becomes weaker as the starting points are further from real values.

**Table 2.** Mean and standard deviation of the error for the second framework

	$m_w$	$s_w$	$m_{\bar{x}_1}$	$s_{\bar{x}_1}$	$m_{\sigma_1}$	$s_{\sigma_1}$	$m_{\bar{x}_2}$	$s_{\bar{x}_2}$	$m_{\sigma_2}$	$s_{\sigma_2}$
Case a	0.059	0.077	0.078	0.078	0.142	0.172	0.051	0.050	0.071	0.241
Case b	0.089	0.114	0.119	0.136	0.379	0.400	0.100	0.109	0.265	0.325
Case c	0.515	0.090	1.035	0.215	0.455	0.230	0.812	0.292	0.184	0.323

## 5 Conclusion

This paper has addressed the problem of the online estimate of mixture model parameters in the Riemannian framework. In particular we dealt with the case of mixtures of Gaussian distributions in the Riemannian manifold of SPD matrices. Starting from a classical approach proposed by Titterton for the Euclidean case, we extend the algorithm to the Riemannian case. The key point was that to look at the innovation part in the step-wise algorithm as an exponential map, or a retraction, in the manifold. Furthermore, an important contribution was that to consider Information Fisher matrix in the Riemannian manifold, in order to implement the Newton algorithm. Finally, we presented some first simulations to validate the proposed method. We can state that, when the starting point of the algorithm is close to the real parameters, we are able to estimate the parameters very accurately. The simulation results suggested us the next future work needed, that is to investigate on the starting point influence in the algorithm, to find some ways to improve convergence towards the good optimum. Another perspective is to apply this algorithm on some real dataset where online estimation is needed.

## References

1. Pennec, X., Fillard, P., Ayache, N.: A riemannian framework for tensor computing. *Int. J. Comput. Vis.* **66**(1), 41–66 (2006)
2. Barachant, A., Bonnet, S., Congedo, M., Jutten, C.: Multiclass brain-computer interface classification by riemannian geometry. *IEEE Trans. Biomed. Eng.* **59**(4), 920–928 (2012)
3. Arnaudon, M., Barbaresco, F., Yang, L.: Riemannian medians and means with applications to Radar signal processing. *IEEE J. Sel. Topics Signal Process.* **7**(4), 595–604 (2013)
4. Tuzel, O., Porikli, F., Meer, P.: Pedestrian detection via classification on Riemannian manifolds. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(10), 1713–1727 (2008)
5. Dong, G., Kuang, G.: Target recognition in SAR images via classification on riemannian manifolds. *IEEE Geosci. Remote Sens. Lett.* **21**(1), 199–203 (2015)
6. Said, S., Bombrun, L., Berthoumieu, Y., Manton, J.H.: Riemannian gaussian distributions on the space of covariance matrices. *IEEE Trans. Inf. Theory* **63**(4), 2153–2170 (2017)
7. Said, S., Bombrun, L., Berthoumieu, Y.: Texture classification using Rao’s distance: An EM algorithm on the Poincaré half plane. In: *International Conference on Image Processing (ICIP)* (2015)



8. Zanini, P., Congedo, M., Jutten, C., Said, S., Berthomieu, Y.: Parameters estimate of riemannian gaussian distribution in the manifold of covariance matrices. In: IEEE Sensor Array and Multichannel Signal Processing Workshop (IEEE SAM 2016) (2016)
9. Titterton, D.: Recursive parameter estimation using incomplete data. *J. Royal Stat. Soc. Ser. B (Stat. Methodologies)* **46**(2), 257–267 (1984)
10. Cappé, O., Moulines, E.: Online EM algorithm for latent data models. *J. Roy. Stat. Soc. Ser. B (Stat. Methodologies)* 593–613 (2009)
11. Helgason, S.: *Differential Geometry, Lie Groups, and Symmetric Space*, vol. 34. American Mathematical Society, Providence (2012)
12. Said, S., Berthomieu, Y.: Warped metrics for location-scale models (2017). [arXiv:1702.07118v1](https://arxiv.org/abs/1702.07118v1)
13. Said, S., Hajri, H., Bombrun, L., Vemuri, B.: Gaussian distributions on riemannian symmetric spaces: statistical learning with structured covariance matrices (2016). [arXiv:1607.06929v1](https://arxiv.org/abs/1607.06929v1)