A probabilistic view on Riemannian machine learning models for SPD matrices

Thibault de Surrel[1][0009−0001−5341−1538], Florian Yger[2][0000−0002−7182−8062], Fabien Lotte[3][0000−0002−6888−9198], and Sylvain Chevallier[4][0000−0003−3027−8241]

[1] LAMSADE, CNRS, PSL Univ. Paris-Dauphine, France
thibault.de-surrel@lamsade.dauphine.fr
[2] LITIS, INSA Rouen-Normandy, Rouen, France
[3] Inria center, University of Bordeaux / LaBRI, Talence, France
[4] TAU, LISN, University Paris-Saclay, France

Abstract. The goal of this paper is to show how different machine learning tools on the Riemannian manifold $\mathcal{P}_d$ of Symmetric Positive Definite (SPD) matrices can be united under a probabilistic framework. For this, we will need several Gaussian distributions defined on $\mathcal{P}_d$. We will show how popular classifiers on $\mathcal{P}_d$ can be reinterpreted as Bayes Classifiers using these Gaussian distributions. These distributions will also be used for outlier detection and dimension reduction. By showing that those distributions are pervasive in the tools used on $\mathcal{P}_d$, we allow for other machine learning tools to be extended to $\mathcal{P}_d$.

Keywords: Symmetric Positive Definite matrices · Riemannian geometry · Probabilistic framework · Classification · Outlier detection · Dimension reduction.

1 Introduction

Symmetric Positive Definite (SPD) matrices appear in several applications of Machine Learning (ML), such as in Brain Computer Interfaces [34], biomedical image analysis [24] or video processing [29]. More precisely, the Riemannian structure of the set of $d \times d$ SPD matrices, denoted $\mathcal{P}_d$, has been leveraged to develop a range of tools to deal with data lying on this manifold. In this work, we use the affine invariant Riemannian framework [24], built from the following Riemannian metric on $\mathcal{P}_d$, called the Affine Invariant Riemannian Metric (AIRM) and defined on the tangent space $T_P\mathcal{P}_d$ at a point $P \in \mathcal{P}_d$ by:

$$\forall U, V \in T_P\mathcal{P}_d,\ \langle U, V\rangle_P = \mathrm{tr}(P^{-1}UP^{-1}V). \tag{1}$$

Using the geodesic induced by this geometry, one can define the AIRM distance between two points $P, Q \in \mathcal{P}_d$ as:

$$\delta(P, Q) = \| \log(P^{-1/2}QP^{-1/2})\|_F$$

The goal of this paper is to revisit some ML tools used on $\mathcal{P}_d$ with a probabilistic lens. More precisely, we will show how different probability distributions on

$\mathcal{P}_d$ can be used to reinterpret classifiers, tools detecting outliers or performing dimension reduction. We will show that all these tools can be united under the same probabilistic framework. For this, we will start by describing, in section 2, the different probability distributions on $\mathcal{P}_d$ that we will use. In section 3, popular classifiers on $\mathcal{P}_d$ will be reinterpreted using those distributions, as well as an outlier detection tool in section 4 and dimension reduction tools in section 5.

2 Different probability distributions on the manifold of SPD matrices

2.1 The isotropic Gaussian distribution

The authors of [26] define an isotropic Gaussian distribution $G(\bar{X}, \sigma^2)$ on $\mathcal{P}_d$ by its probability density function, with respect to the Riemannian volume:

$$p_{\bar{X},\sigma}(X) = \frac{1}{\zeta(\sigma)} \exp\left(-\frac{\delta(X,\bar{X})^2}{2\sigma^2}\right).$$

This distribution depends on two parameters: $\bar{X} \in \mathcal{P}_d$ acting as the center of mass of the distribution and $\sigma > 0$ acting as the spread of the distribution. The authors also give an exact expression of the normalization constant $\zeta(\sigma)$ that does not depend on $\bar{X}$, but only on σ. Using proposition 7 of [26], one can estimate the two parameters $\bar{X}$ and σ of the distribution from an independent sample $X_1, ..., X_N \sim G(\bar{X}, \sigma)$ using the Maximum Likelihood Estimator (MLE): for $\bar{X}$, the MLE $\hat{X}_N$ is the Riemannian mean of the sample (see [21]) and for σ, the MLE is the only solution of the following non-linear equation in σ:

$$\sigma^3 \frac{\mathrm{d}}{\mathrm{d}\sigma} \log \zeta(\sigma) = \frac{1}{N} \sum_{i=1}^{N} \delta(X_i, \hat{X}_N)^2.$$

2.2 Wrapped distributions

We now share a way to define an anisotropic Gaussian distribution on the manifold $\mathcal{P}_d$ using the concept of *wrapped Gaussian* (WG) distribution [11]. This distribution depends on three parameters: $P \in \mathcal{P}_d$, $\mu \in \mathbb{R}^{d(d+1)/2}$ and $\Sigma \in \mathcal{P}_{d(d+1)/2}$. Then, we say that the random variable $\mathbf{X}$ on $\mathcal{P}_d$ follows a wrapped Gaussian distribution $\mathrm{WG}(P; \mu, \Sigma)$ if:

$$\mathbf{X} = \mathrm{Exp}_P(\mathbf{t}),\ \mathbf{t} \sim \mathcal{N}(\mu, \Sigma)$$

The wrapped Gaussian corresponds to a multivariate Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ in normal coordinates at P. This distribution is further analyzed in [28]. In particular, the density of $\mathrm{WG}(P; \mu, \Sigma)$ is given by:

$$f_{P;\mu,\Sigma}(X) = \frac{g_{\mu,\Sigma}(\mathrm{Log}_P(X))}{|J_P(\mathrm{Log}_P(X))|}$$

where $g_{\mu,\Sigma}$ is the density of the multivariate Gaussian $\mathcal{N}(\mu, \Sigma)$ and $J_P(\cdot) = \det(\mathrm{d}\,\mathrm{Exp}_P(\cdot))$ is the Jacobian determinant of the exponential map Exp_P. The closed-form formula of the Jacobian determinant J_P is given in Proposition 4.3 of [28]. Moreover, the authors show that the parameters (P, μ, Σ) of the WG can be estimated from a sample $X_1, ..., X_N$ using the method of moments in the case of $\mu = 0$ *a priori* and using an MLE in a general setting where $\mu \neq 0$.

2.3 Other probability distributions on $\mathcal{P}_d$

An alternative definition of an anisotropic Gaussian distribution on $\mathcal{P}_d$ is given in [22]. This approach maximizes entropy given a mean and covariance matrix (see Theorem 13.2.2 of [17]). On $\mathcal{P}_d$, the entropy-maximizing distribution with mean $\bar{X}$ and concentration matrix Γ has density:

$$p_{\bar{X},\Gamma}(X) = k \exp\left(-\frac{\mathrm{Log}_{\bar{X}}(X)^\top \Gamma\, \mathrm{Log}_{\bar{X}}(X)}{2}\right),$$

where k is a normalization constant without a closed form. The relation between Γ and the covariance Σ is given in Theorem 3 of [22]. When $\Sigma = \sigma^2 I_d$, this reduces to the isotropic Gaussian of section 2.1. Another example of a Gaussian distribution of $\mathcal{P}_d$ is given in [8], where the author uses Souriau's covariant Gibbs density to compute a Gaussian density of SPD matrices.

Other - non-Gaussian - probability distributions have been defined on the manifold $\mathcal{P}_d$ of SPD matrices. For example, a famous distribution used for SPD matrices is the Wishart distribution [32]. It is the distribution of sample covariance matrices of random vectors drawn from a multivariate Gaussian distribution and can be seen as a generalization of the gamma distribution to multiple dimensions. This Wishart distribution was first extended in [3] to the t-Wishart distribution similarly to the way the multivariate t-distribution extends the multivariate Gaussian. In [4], the authors extend the Wishart distribution to an elliptical version, leading to a more robust and flexible distribution. In [1], the authors describe a Cholesky normal distribution on the manifold of SPD matrices. This distribution is related to the Wishart distribution as it relies on random vectors drawn from a multivariate Gaussian distribution. Moreover, the authors show that the Wishart distribution is approximately the Cholesky normal distribution for large degrees of freedom.

3 Application to classification

In this section, we show that popular classification algorithms on $\mathcal{P}_d$ can be reinterpreted in the light of a probabilistic framework using the different Gaussian distributions defined in section 2.1. In the following, we assume that we are in a supervised setting where we have K classes each modeled by a distribution denoted α_k. We start by introducing the Bayes Classifier (BC) :

Definition 1 (Bayes Classifier (BC) [10]). *Given K classes each modeled by a distribution α_k, a Bayes Classifier (BC) assigns a new sample Z to the class k that maximizes the likelihood of Z under the distribution α_k.*

Our goal is to show that popular classification algorithms on $\mathcal{P}_d$ can be interpreted as BC using the different Gaussian distributions defined in section 2. We will focus on the Minimum Distance to Mean (MDM) algorithm [7], on the Tangent Space Linear Discriminant Analysis (LDA) and on the Tangent Space Quadratic Discriminant Analysis (QDA) algorithms [5].

3.1 Minimum Distance to Mean (MDM)

The *Minimum Distance to Mean* (MDM) algorithm is a simple and popular classification algorithm for SPD matrices described in [7]. Given a training set of SPD matrices, this algorithm estimates the Riemannian mean $\hat{X}^k$ for each class $k \in \{1, ..., K\}$ and assigns a new sample Z to the class k that minimizes the Riemannian distance to the estimated mean $\hat{X}^k$. We can reinterpret the MDM algorithm in the light of the isotropic Gaussian distribution defined in section 2.1. Indeed, we have the following:

Proposition 1. *Let us suppose that each class is modeled by an isotropic Gaussian distribution centered at $\bar{X}^k$ with a shared σ, i.e. $\alpha_k = G(\bar{X}^k, \sigma^2)$, then, the MDM converges to the BC when the number of data tends to infinity.*

When the number of data tends to infinity, the estimation of the Riemannian mean of each class $\hat{X}^k$ converges to the true mean of the k-th class $\bar{X}^k$. Here, the spread σ does not play any role in the classification, so one only needs to estimate the Riemannian mean $\bar{X}^k$ of each class.

3.2 Tangent Space LDA/QDA

The Tangent Space LDA (resp. QDA) algorithm is a generalization of the LDA (resp. QDA) algorithm to the manifold of SPD matrices introduced in [5]. The idea is to first estimate the Riemannian mean $\hat{X}_N$ of the whole training set, and then project the training SPD matrices onto the tangent space $T_{\hat{X}_N}\mathcal{P}_d$. This tangent space being Euclidean, one finally applies the classical LDA (resp. QDA) algorithm (see section 4.3 of [13]) in this tangent space. Using the wrapped Gaussian distribution described in section 2.2, one has the following proposition:

Proposition 2. *When the number of data points tends to infinity, the Tangent Space LDA converges to a BC where the classes are modeled by wrapped Gaussian distributions centered at*

$$\bar{X} = \underset{Y \in \mathcal{P}_d}{\arg\min} \int_{\mathcal{P}_d} \delta(X, Y)^2 \mathrm{d}\alpha(X)$$

where $\alpha = \frac{1}{k}\sum_{i=1}^{k} \alpha_k$ is the total distribution, and with a shared covariance matrix Σ, in other word, $\alpha_k = \mathrm{WG}(\bar{X}, \mu_k, \Sigma)$. Similarly, the Tangent Space QDA converges to a BC where the classes are modeled by $\alpha_k = \mathrm{WG}(\bar{X}, \mu_k, \Sigma_k)$.

Here, $\bar{X}$ is the Fréchet mean of total distribution as defined in definition 2 of [23] and can be estimated by the Riemannian mean $\hat{X}_N$ of the training data. Then, estimating μ_k and Σ_k boils down to the Euclidean estimation of the mean and covariance matrix in $T_{\hat{X}}\mathcal{P}_d$. One can use wrapped Gaussian to build new BC on $\mathcal{P}_d$ by modeling the classes α_k by $\mathrm{WG}(\bar{X}_k, \mu_k, \Sigma)$ or by $\mathrm{WG}(\bar{X}_k, \mu_k, \Sigma_k)$. This is done in [28]. These algorithms project each class to their own tangent space whereas only one tangent space is used in the Tangent Space LDA/QDA.

3.3 Other probabilistic classifiers for SPD matrices

In this paper, we mainly focus on the different Gaussian distributions described in section 2. However, other authors have inquired BC based on other probability distributions. For instance, in [4], the authors propose a BC based on the Elliptical Wishart distribution. They also adapt it to an unsupervised clustering scenario leveraging the K-means clustering algorithm. In [3], the authors show that if the metric used in the MDM algorithm is based on the Kullback-Leibler divergence between two centered multivariate Gaussian distributions, then, the MDM algorithm is a BC based on the t-Wishart distribution.

4 Application to outlier detection: Riemannian potato

Another tool that is used when the SPD matrices are covariance matrices of ElectroEncephaloGraphy (EEG) signals is the *Riemannian potato* introduced in [6]. This tool is used to automatically detect outliers in the data. The idea of the Riemannian potato is to estimate a reference SPD matrix and a measure of dispersion (z-score), and then to reject all SPD matrices that are too far from the reference matrix. Let $(X_i)_{i=1,\dots,N}$ be a set of SPD matrices, and let us denote $\bar{X}$ the reference SPD matrix. Then, the z-score z of $X \in \mathcal{P}_d$ is computed as:

$$z = \frac{\delta(X, \bar{X}) - \mu}{\sigma} \text{ where } \mu = \frac{1}{N}\sum_{i=1}^{N} \delta(X_i, \bar{X}) \text{ and } \sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (\delta(X_i, \bar{X}) - \mu)^2}.$$

Provided a threshold z_{th}, the matrix X is accepted if $z \leq z_{\mathrm{th}}$. The Riemannian potato is a simple yet efficient tool to detect outliers in the data. In practice, the reference matrix $\bar{X}$ is a Riemannian mean computed iteratively as new samples are added to the dataset. Using the isotropic Gaussian distribution defined in section 2.1, we can see that the Riemannian potato rejects points that are too unlikely under a certain isotropic Gaussian:

Proposition 3. *Let $\bar{X} \in \mathcal{P}_d$ be a reference matrix, $\mu \in \mathbb{R}, \sigma > 0$ and $z_{th} > 0$. Then, $X \in \mathcal{P}_d$ is accepted by the Riemannian potato with threshold z_{th} if and only if $\mathcal{L}_{\bar{X},\sigma}(X) \geq \ell_{th}$ where $\mathcal{L}_{\bar{X},\sigma}$ denotes the likelihood of $G(\bar{X}, \sigma)$ and where*

$$\ell_{th} = \exp\left(-\frac{1}{2}\left(z_{th} + \frac{\mu}{\sigma}\right)\right)$$

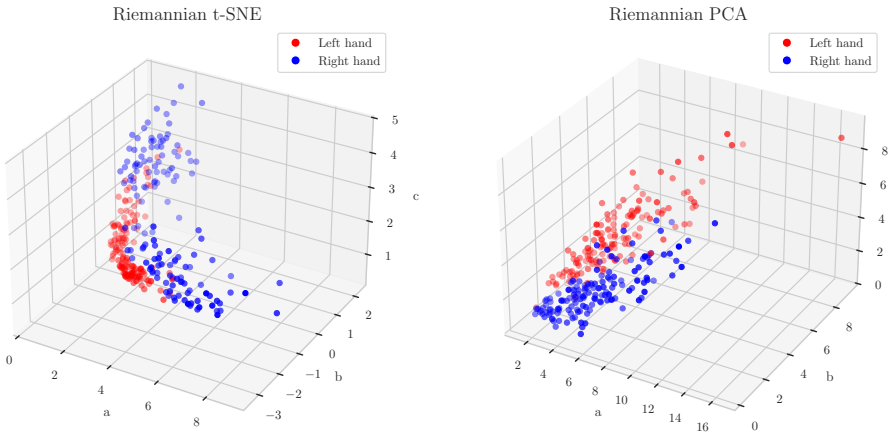

Fig. 1. Results of the Riemannian t-SNE and PCA algorithms applied to the covariance matrices of EEG signals from subject 8 of the BNCI2014001 [18] dataset. The points are colored according to the class of the task performed by the subject. Both algorithms reduce the data to 2×2 SPD matrices, which can be visualized in 3D. Indeed, a 2×2 SPD matrices is of the form $\begin{pmatrix} a & b \\ b & c \end{pmatrix}$, which can be represented in 3D by the point (a, b, c).

Riemannian potato have been extended to a Riemannian potato field [9]. Another use of the isotropic Gaussian distribution to detect outliers is done in [31] where the author built an online change detection algorithm based on the estimation of the Riemannian mean of the data before and after the change.

5 Application to dimension reduction

5.1 Riemannian t-SNE

In [27], the authors define a Riemannian version of the t-SNE algorithm from [20] that reduces $d \times d$ SPD matrices into 2×2 SPD matrices. As the set $\mathcal{P}_2$ of 2×2 SPD matrices is of dimension 3, it can be visualized in a 3D space. To adapt the algorithm to the manifold $\mathcal{P}_d$, the authors replaced the Euclidean distance by the AIRM distance and showed that the algorithm is still valid. The Riemannian t-SNE algorithm is based on the computation of similarities between the high-dimensional SPD matrices using a Gaussian kernel. This kernel is based on the isotropic Gaussian distribution defined in section 2.1. Denoting $X_1, ..., X_N$ the set of high-dimensional SPD matrices, the similarity $p_{i|j}$ of the matrix X_i to X_j is the conditional probability $p_{j|i}$ that X_i would pick X_j as its neighbor if they were picked in proportion to their probability density under an isotropic Gaussian $G(X_i, \sigma_i^2)$ centered at X_i. The dispersion σ_i is computed using the perplexity parameter of the t-SNE algorithm. Then, the Riemannian

t-SNE aims at learning $Y_1, ..., Y_N \in \mathcal{P}_2$, i.e., the SPD matrices with the reduced dimension, that reflect the similarities $p_{i|j}$ as well as possible. For this, the joint probabilities q_{ij} of the low-dimensional SPD matrices Y_i and Y_j are computed using a Riemannian version of the Student-t distribution with one degree of freedom. Finally, the Kullback-Leibler divergence between the two distributions is minimized using a Riemannian gradient descent algorithm.

We apply this Riemannian t-SNE to a dataset of SPD matrices that are covariance matrices of EEG signals used in Brain Computer Interfaces. The dataset is the BNCI2014001 dataset [18] from MOABB [2] that contains 9 subjects performing motor imagery tasks. The Riemannian t-SNE algorithm is applied to the covariance matrices of the EEG signals of subject 8 of the dataset, and the results of given at fig. 1. The Riemannian t-SNE algorithm reduces the data to 2×2 SPD matrices, which can be visualized in 3D. The points are colored according to the class of the task performed by the subject. The Riemannian t-SNE algorithm is able to separate the different classes of tasks performed by the subject, showing that it is a good tool for dimension reduction on $\mathcal{P}_d$.

5.2 Riemannian PCA

The goal of the *Principal Component Analysis* (PCA) algorithm is to project the data onto a lower-dimensional space while maximizing the variance of the projected points. In [14], the authors extend the PCA to a Riemannian setting of SPD matrices. Given a set $X_1, ..., X_N$ of SPD matrices, their goal is to find a W that maximizes the variance of the projected points $W^\top X_1 W, ..., W^\top X_N W$. The matrix W is chosen in $\mathcal{G}(d, p)$, the Grassmann manifold, that is the manifold of $d \times p$ matrices of rank p. This is to make sure that the projected points are SPD matrices. Therefore, the Riemannian PCA is defined as the following optimization problem, where $\hat{X}_N$ is the Riemannian mean of the set $X_1, ..., X_N$:

$$W \in \underset{W \in \mathcal{G}(d,p)}{\operatorname{argmax}} \sum_{i=1}^{N} \delta(W^\top X_i W, W^\top \hat{X}_N W)^2$$

We recall that the variance of a distribution μ on $\mathcal{P}_d$, at a point X, as defined in [23], is $\sigma^2(X) = \int_{\mathcal{P}_d} \delta(X, Y)^2 \mathrm{d}\mu(Y)$. In the discrete setting of Riemannian PCA, the integral is replaced by a sum. Therefore, the underlying distribution of the Riemannian PCA is the isotropic Gaussian distribution defined in section 2.1.

We also apply this Riemannian PCA algorithm to the same dataset of covariance matrices of EEG signals used in the Riemannian t-SNE algorithm. The results are given at fig. 1. As the Riemannian t-SNE, we reduce the covariance matrices to 2×2 SPD matrices, which can be visualized in 3D. The Riemannian PCA algorithm also demonstrates the ability to separate the different classes of tasks performed by the subject, confirming its effectiveness for dimension reduction on $\mathcal{P}_d$.

6 Conclusion

In this paper, we first demonstrated how the isotropic Gaussian distribution and the wrapped Gaussian distribution can be used to reinterpret the MDM, Tangent Space LDA and Tangent Space QDA classifiers as Bayes Classifiers. Therefore, these algorithms share a common probabilistic framework, differing only in their choice of data distribution. Using the same framework, we showed that the Riemannian potato rejects points that are unlikely under an isotropic Gaussian distribution. Finally, the Riemannian t-SNE and the Riemannian PCA algorithms were also encompassed in the same probabilistic framework. This work shows that the various tools used on the manifold of SPD matrices can be brought together under the same probabilistic framework.

This work opens the door to using other probability distributions on $\mathcal{P}_d$ and constructing other ML tools using them. For example, a possible use of the different Gaussian distributions is the construction of Gaussian kernels on $\mathcal{P}_d$. Kernels on $\mathcal{P}_d$ have already been investigated [33, 16], and the use of the isotropic Gaussian distribution to build a kernel leads to a strong limitation due to the curvature of $\mathcal{P}_d$ (see [12]). However, the wrapped Gaussian distribution could lead to a more flexible kernel that could be used in a wide range of applications. Deep Learning methods have also been investigated on $\mathcal{P}_d$ [15, 19, 30], and the use of the different probability distributions could lead to a new outlook on deep models on $\mathcal{P}_d$. On more general Riemannian manifolds, the isotropic Gaussian has for example been used to extend Variational Flow Matching to Riemannian manifolds in [35] and wrapped distributions have been leveraged to learn Riemannian latent spaces in [25].

Acknowledgments. This work was funded by the French National Research Agency for project PROTEUS (grant ANR-22-CE33-0015-01). Sylvain Chevallier is supported by DATAIA (ANR-17-CONV-0003).

References

1. Ahanda, B., Ellingson, L., Osborne, D.E.: The Cholesky normal distribution for SPD matrices and inference for the mean. Statistical Papers **66**(1), 19 (Dec 2024)
2. Aristimunha, B., Carrara, I., Guetschel, P., Sedlar, S., Rodrigues, P., Sosulski, J., Narayanan, D., Bjareholt, E., Quentin, B., Schirrmeister, R.T., Kalunga, E., Darmet, L., Gregoire, C., Abdul Hussain, A., Gatti, R., Goncharenko, V., Thielen, J., Moreau, T., Roy, Y., Jayaram, V., Barachant, A., Chevallier, S.: Mother of all BCI Benchmarks (2023). https://doi.org/10.5281/zenodo.10034223, https://github.com/NeuroTechX/moabb
3. Ayadi, I., Bouchard, F., Pascal, F.: T-WDA: A novel Discriminant Analysis applied to EEG classification. In: EUSIPCO. Helsinki, Finland (Sep 2023)
4. Ayadi, I., Bouchard, F., Pascal, F.: Elliptical Wishart distributions: Information geometry, maximum likelihood estimator, performance analysis and statistical learning (Nov 2024)

5. Barachant, A., Bonnet, S., Congedo, M., Jutten, C.: Multiclass Brain–Computer Interface Classification by Riemannian Geometry. IEEE Trans. Biomed. Eng. **59**(4), 920–928 (Apr 2012)
6. Barachant, A., Andreev, A., Congedo, M.: The Riemannian Potato: An automatic and adaptive artifact detection method for online experiments using Riemannian geometry. In: TOBI Workshop lV. pp. 19–20. Sion, Switzerland (Jan 2013)
7. Barachant, A., Bonnet, S., Congedo, M., Jutten, C.: Riemannian geometry applied to BCI classification. In: LVA/ICA 2010. vol. 6365, p. 629. Springer (Sep 2010)
8. Barbaresco, F.: Gaussian Distributions on the Space of Symmetric Positive Definite Matrices from Souriau's Gibbs State for Siegel Domains by Coadjoint Orbit and Moment Map. In: Nielsen, F., Barbaresco, F. (eds.) Geometric Science of Information. pp. 245–255. Springer International Publishing, Cham (2021)
9. Barthélemy, Q., Mayaud, L., Ojeda, D., Congedo, M.: The Riemannian Potato Field: A Tool for Online Signal Quality Index of EEG. IEEE Transactions on Neural Systems and Rehabilitation Engineering **27**(2), 244–255 (Feb 2019)
10. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer, 1 edn. (2007)
11. Chevallier, E., Li, D., Lu, Y., Dunson, D.: Exponential-Wrapped Distributions on Symmetric Spaces. SIMODS **4**(4), 1347–1368 (Dec 2022)
12. Feragen, A., Lauze, F., Hauberg, S.: Geodesic exponential kernels: When curvature and linearity conflict. In: CVPR. pp. 3032–3042. IEEE (2015)
13. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer Series in Statistics, Springer, New York, NY (2009)
14. Horev, I., Yger, F., Sugiyama, M.: Geometry-aware principal component analysis for symmetric positive definite matrices. ACML **45**, 1–16 (20–22 Nov 2016)
15. Huang, Z., Van Gool, L.: A riemannian network for spd matrix learning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 31 (2017)
16. Jayasumana, S., Hartley, R., Salzmann, M., Li, H., Harandi, M.: Kernel methods on the riemannian manifold of symmetric positive definite matrices (2014), https://arxiv.org/abs/1412.4172
17. Kagan, A., Linnik, I., Rao, C.: Characterization Problems in Mathematical Statistics. Probability and Statistics Series, Wiley (1973)
18. Leeb, R., Lee, F., Keinrath, C., Scherer, R., Bischof, H., Pfurtscheller, G.: Brain–computer communication: Motivation, aim, and impact of exploring a virtual apartment. IEEE Transactions on Neural Systems and Rehabilitation Engineering **15**(4), 473–482 (2007). https://doi.org/10.1109/TNSRE.2007.906956
19. Li, Y., Yu, Z., He, G., Shen, Y., Li, K., Sun, X., Lin, S.: SPD-DDPM: Denoising Diffusion Probabilistic Models in the Symmetric Positive Definite Space (Dec 2023)
20. van der Maaten, L., Hinton, G.: Visualizing Data using t-SNE. Journal of Machine Learning Research **9**(86), 2579–2605 (2008)
21. Moakher, M.: A Differential Geometric Approach to the Geometric Mean of Symmetric Positive-Definite Matrices. SIMAX **26**(3), 735–747 (Jan 2005)
22. Pennec, X.: Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements. J. Math. Imaging Vis. **25**(1), 127–154 (Jul 2006)
23. Pennec, X.: Curvature effects on the empirical mean in riemannian and affine manifolds: a non-asymptotic high concentration expansion in the small-sample regime. arXiv preprint arXiv:1906.07418 (2019)
24. Pennec, X.: Manifold-valued image processing with SPD matrices. In: Pennec, X., Sommer, S., Fletcher, T. (eds.) Riemannian Geometric Statistics in Medical Image Analysis, pp. 75–134. Academic Press (Jan 2020)

25. Rozo, L., González-Duque, M., Jaquier, N., Hauberg, S.: Riemann^2: Learning riemannian submanifolds from riemannian data. In: Proceedings of the 19th international Conference on Artificial Intelligence and Statistics (AISTATS) (2025)
26. Said, S., Bombrun, L., Berthoumieu, Y., Manton, J.: Riemannian Gaussian Distributions on the Space of Symmetric Positive Definite Matrices (Dec 2016)
27. de Surrel, T., Chevallier, S., Lotte, F., Yger, F.: Geometry-aware visualization of high dimensional symmetric positive definite matrices. TMLR (2025)
28. de Surrel, T., Lotte, F., Chevallier, S., Yger, F.: Wrapped gaussian on the manifold of symmetric positive definite matrices. In: Forty-second International Conference on Machine Learning (2025), https://openreview.net/forum?id=EhStXG4dCS
29. Tuzel, O., Porikli, F., Meer, P.: Pedestrian detection via classification on riemannian manifolds. IEEE PAMI **30**(10), 1713–1727 (2008)
30. Wang, R., Wu, X.J., Chen, Z., Xu, T., Kittler, J.: DreamNet: A Deep Riemannian Manifold Network for SPD Matrix Learning. In: ACCV 2022, vol. 13846, pp. 646–663. Springer Nature Switzerland, Cham (2023)
31. Wang, X., Borsoi, R.A., Richard, C.: Non-parametric Online Change Point Detection on Riemannian Manifolds. In: Proceedings of the 41st ICML (Jul 2024)
32. Wishart, J.: The Generalised Product Moment Distribution in Samples from a Normal Multivariate Population. Biometrika **20A**(1/2), 32–52 (1928)
33. Yger, F.: A review of kernels on covariance matrices for bci applications. In: MLSP. pp. 1–6 (2013)
34. Yger, F., Berar, M., Lotte, F.: Riemannian Approaches in Brain-Computer Interfaces: A Review. IEEE TNSRE **25**(10), 1753–1762 (Oct 2017)
35. Zaghen, O., Eijkelboom, F., Pouplin, A., Bekkers, E.J.: Towards variational flow matching on general geometries (2025), https://arxiv.org/abs/2502.12981