Fast Equivariant K-Means on SPD Matrices Using Log-Extrinsic Means

Gabriel Trindade[1], Emmanuel Chevallier[1](✉), André Nicolet[1], and Frank Nielsen[2]

[1] Institut Fresnel, Aix-Marseille Univ, CNRS, Centrale Marseille, Marseille, France
emmanuel.chevallier@univ-amu.fr
[2] Sony Computer Science Laboratories Inc., Tokyo, Japan

Abstract. In this paper, we propose an efficient alternative to the affine-invariant Riemannian k-means algorithm on symmetric positive definite matrices. Recently introduced log-extrinsic means are coupled with the Jensen-Bregman log-det divergence, as a replacement for the Riemannian Fréchet mean and the Riemannian distance. Performances and computation times are compared for several frameworks on point clouds sampled from Riemannian Gaussians. Results show that our algorithm matches the clustering accuracy of the affine-invariant Riemannian k-means, while achieving runtimes comparable to those of log-Euclidean k-means.

Keywords: k-means algorithm · Log-extrinsic means · Affine-invariant Riemannian metric · Jensen-Bregman log-det divergence

1 Introduction

Twenty years have passed since the influential series of papers popularizing affine-invariant geometry for symmetric positive definite matrices (SPD) were published [8,10,12,13]. Over these two decades, the log-Euclidean framework has served as the main alternative to Riemannian methods for fast computations on SPD matrices [1,2].

Authors of [7] recently introduced a notion of log-extrinsic means based on a logarithmic mean on determinants and an extrinsic mean on the determinant one hypersurface. In the current paper, we exploit this new type of means to propose an efficient k-means algorithm on symmetric positive definite matrices.

Log-extrinsic means are designed to mimic Fréchet means computed with affine-invariant Riemannian metrics, that is to say metrics invariant under the congruence action $G \cdot X = GXG^T$. For a set of $n \times n$ positive definite matrices $X_1, \ldots, X_k$, the log-extrinsic mean is defined as

$$L(X_1, \ldots, X_k) = e^{\frac{1}{kn}\sum_i \log(\det(X_i))} \frac{\sum_i \frac{X_i}{\det(X_i)^{1/n}}}{\det\left(\sum_i \frac{X_i}{\det(X_i)^{1/n}}\right)^{1/n}}. \tag{1}$$

Since the mean has an explicit expression, the Fréchet mean in a Riemannian k-means algorithm could be advantageously replaced by the log-extrinsic mean. Beyond the mean, the k-means algorithm involves large numbers of evaluations

F. Nielsen and F. Barbaresco (Eds.): GSI 2025, LNCS 16035, pp. 263–270, 2026.
https://doi.org/10.1007/978-3-032-03924-8_27

of distances between points and cluster centers. An efficient alternative to Riemannian distances is the Jensen-Bregman log-det divergence,

$$JB(X,Y) = \log\det\left(\frac{X+Y}{2}\right) - \frac{1}{2}\log\det(XY), \tag{2}$$

see [5]. The k-means algorithm presented in this paper combines the log-extrinsic mean for the computation of cluster centers and the Jensen-Bregman log-det divergence for the evaluation of similarities with the centers. We analyze the relevance of the resulting algorithm for point clouds sampled from Riemannian-isotropic probability distributions.

Section 2 introduces the necessary concepts and results on log-extrinsic and Fréchet means. Section 3 describes the different variants of the k-means considered, as well as the procedures to generate clusters. Results are presented in Sect. 4 and Sect. 5 concludes the paper.

2 Log-Extrinsic Means on SPD(n) Cones

This section summarizes the notions and results presented in [7] for the log-extrinsic means on the cone SPD(n) of symmetric positive definite matrices of size $n \times n$ on $\mathbb{R}$. The main idea behind log-extrinsic means is to decompose the cone SPD(n) as a Cartesian product between determinant-one matrices and determinants. The log-extrinsic mean is then obtained from an extrinsic mean on determinant-one matrices and a logarithmic mean on determinants.

Firstly, define $\mathcal{S} := \{X \in \mathrm{SPD(n)} : \det(X) = 1\}$ and the projection $\pi : \mathrm{SPD(n)} \to \mathcal{S}$ given by

$$\pi(X) = \frac{X}{\det(X)^{1/n}}.$$

In the rest of the paper, the symbol $\mathbb{E}$ refers to the usual expectation of random variables. Additionally, for a random matrix X taking its values on $\mathcal{S} \subset \mathrm{SPD}(n)$, define its extrinsic mean $E(X)$ as

$$E(X) := \pi\left(\mathbb{E}(X)\right).$$

Now, consider the function $H : \mathrm{SPD}(n) \to \mathbb{R},\ X \mapsto \log\det(X)$, and the diffeomorphism $\phi : \mathrm{SPD(n)} \to \mathbb{R} \times \mathcal{S}$ given by $\phi = (H, \pi)$. Since the determinant is n-linear, the inverse of ϕ is given by

$$\phi^{-1}(\alpha, \Sigma) = e^{\frac{\alpha}{n}}\Sigma.$$

The log-extrinsic mean $L(X)$ of a random matrix X taking its values in SPD(n) is defined as

$$L(X) := \phi^{-1}\left(\mathbb{E}(H(X)), E(\pi(X))\right) = e^{\frac{1}{n}\mathbb{E}(H(X))}E(\pi(X)), \tag{3}$$

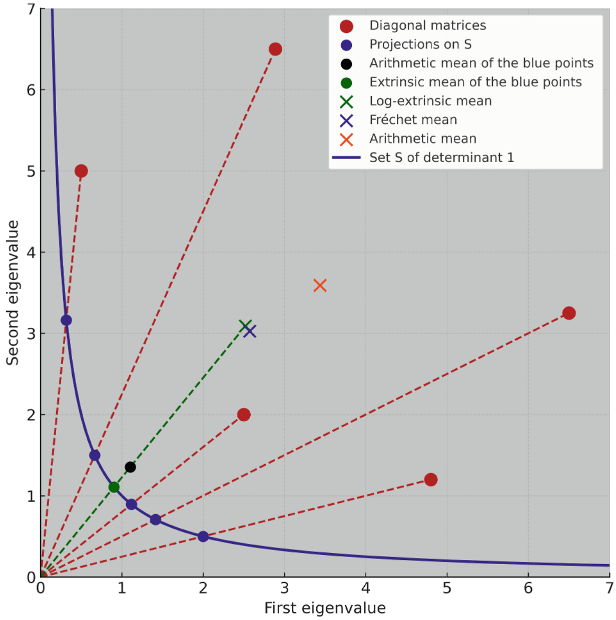

Fig. 1. Means of five 2×2 diagonal matrices (red points) (Color figure online)

It can be checked that Eq. 3 leads to the log-extrinsic mean of a finite sample presented in Eq. 1. Moreover, Fig. 1 shows a graphical illustration of the computation of the mean of 5 points.

It can be verified that the log-extrinsic mean is equivariant with respect to the congruence action: for all $G \in GL(n, \mathbb{R})$,

$$G \cdot L(X) = L(G \cdot X).$$

This can be proved by analyzing the action of G on the Cartesian product $\mathbb{R} \times \mathcal{S}$ induced by the map ϕ.

Let us now relate the log-extrinsic mean to Fréchet means of invariant Riemannian metrics. Recall that, given a distance d and a random variable X, its Fréchet mean $F(X)$ is defined by

$$F(X) := \underset{Y \in \mathrm{SPD(n)}}{\arg\min}\, \mathbb{E}\left(d(X,Y)^2\right),$$

provided that it exists. When d is a Riemannian distance invariant by the congruence action, the Fréchet mean admits the same decomposition as the log-extrinsic means:

$$F(X) = \phi^{-1}\left(\mathbb{E}(H(X)), F(\pi(X))\right) = e^{\frac{1}{n}\mathbb{E}(H(X))} F(\pi(X)). \quad (4)$$

The proof of it relies on the fact that any invariant Riemannian distance d can be decomposed as a product distance between a logarithmic distance on determinant and the induced distance on $\mathcal{S}$. Additionally, it is also possible to prove that the Fréchet mean is independent of the choice of the invariant Riemannian metric, see for instance [7].

We now provide a symmetry condition on the random matrix X under which $L(X)$ and $F(X)$ coincide. Assume that the random matrix X has a density f which is a function of an invariant Riemannian distance to a reference $\overline{X}$: $f(X) = h\left(d\left(X, \overline{X}\right)\right)$, where $h : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. It can then be proved that

$$L(X) = F(X) = \overline{X},$$

provided that both means exist. This result is similar in nature to the one relating the extrinsic mean and the Fréchet mean of an isotropic distribution on a sphere, see theorem 3.3 of [3].

3 *k*-Means Variants and Clusters Generation

Let us now describe the different variants of the k-means algorithm considered in our comparison, as well as the procedures used to generate the point clouds. Algorithm 1 shows the steps shared by the different variants.

Algorithm 1 k-means

Input: Set $M = \{M_i\}$ of SPD(n) matrices, number of clusters k, matrix divergence Div() (generalized distance function), matrix mean Mean().
Output: Set of k centroids $\{C_i\}$, partition $M = \cup A_i$.
1. Choose randomly k different centroids C_i in M.
2. For each pair (C_i, M_j), form $D(i, j) = \mathrm{Div}(C_i, M_j)$.
3. Define $A_i \subset M$ the set of matrices M_j such that $i = \mathrm{argmin}_q\, D(q, j)$.
4. Compute new centroids $C_i^{new} = \mathrm{Mean}(A_i)$
5. If $\max_i \|C_i - C_i^{new}\|_F < 10^{-12}$ or iteration ≥ 100, stop; otherwise set $C_i \leftarrow C_i^{new}$ and go to 2. See Eq.5 for the definition of $\|.\|_F$.
return C_i,A_i, $i \in \{1, \ldots, k\}$

Running a k-means algorithm requires a divergence and a notion of mean. In order to preserve both the equivariance and a low computational complexity of the algorithm, we pair the log-extrinsic mean with the Jensen-Bregman log-det divergence. The table below shows four variants of the k-means algorithm.

In Table 1, the Euclidean distance between two symmetric matrices X and Y refers to the Frobenius norm of matrices

$$\|X - Y\|_F^2 = \mathrm{Trace}\left((X - Y)(X - Y)^T\right) \tag{5}$$

and the log-Euclidean distance refers to the Frobenius norm on the matrix logarithm of matrices, that is,

$$\|\mathrm{Log}\, X - \mathrm{Log}\, Y\|_F\,.$$

Table 1. The four k-means frameworks

Acronym	Divergence	Mean	Equivariance
R	Riemannian distance	Fréchet	Yes
LE	Log-Euclidean distance	Log-arithmetic	No
JBLD	Jensen-Bregman log-det divergence	Log-extrinsic	Yes
E	Euclidean distance	Arithmetic	No

By log-arithmetic, we simply refer to the arithmetic mean on the logarithm of matrices. In the rest of the paper, the invariant Riemannian metric is set to

$$d(X,Y) = \left\| \operatorname{Log} X^{-1/2} Y X^{-1/2} \right\|_F .$$

Since we want to evaluate how well the log-extrinsic mean performs as a replacement for the Fréchet mean, we generate clusters that are isotropic for the Riemannian metric: they are drawn according to the Gaussian distribution presented in [14]. For $\overline{X} \in \mathrm{SPD}(n)$ and $\sigma > 0$, the density with respect to the Riemannian measure of the Riemannian Gaussian $\mathcal{G}(\overline{X}, \sigma)$ is proportional to

$$f\left(X; \overline{X}, \sigma\right) \propto e^{-\frac{d(X,\overline{X})^2}{2\sigma^2}}, \tag{6}$$

where the normalization term depends only on σ. Given that the density is a function of the distance to $\overline{X}$, it follows that

$$L(X) = F(X) = \overline{X}.$$

As suggested in [14], we used a Metropolis-Hastings algorithm to sample from $\mathcal{G}(\overline{X}, \sigma)$. Note that instead of using Riemannian Gaussians $\mathcal{G}$, another possible choice was to consider Exponential-wrapped Gaussian distributions, defined for instance in [6,15]. For these distributions, sampling is straightforward. However, due to the presence of a Jacobian term, their densities are not functions of the Riemannian distance. This makes them less suitable to study performances of the Riemannian k-means algorithm.

We compared the four frameworks presented in Table 1 in two different scenarios. In the first one (i), k matrices $\overline{X}_i$ are sampled from $\mathcal{G}(I, 1)$. For each $\overline{X}_i$, N matrices are sampled from $\mathcal{G}(\overline{X}_i, 0.5)$. This results in k clusters of size N. In the second scenario (ii), the number k of clusters is required to be even. The matrices $\overline{X}_i$ are positioned in a way that exacerbates differences between the log-Euclidean metric and the Riemannian metric. Let T be the subspace of symmetric matrices with vanishing diagonal and let B be the unit ball of T for the Frobenius norm. Consider also D a $n \times n$ diagonal matrix with eigenvalues 10^{-2} and 10^2 with respective multiplicities $\lfloor \frac{n}{2} \rfloor$ and $\lceil \frac{n}{2} \rceil$. Then, we uniformly sampled $\frac{k}{2}$ matrices $\widetilde{X}_i$ in B and defined $\overline{X}_i$ as

$$\overline{X}_i = D^{1/2} \operatorname{Exp}\left(\widetilde{X}_i\right) D^{1/2}.$$

For each $\overline{X}_i$, N matrices are sampled from $\mathcal{G}(\overline{X}_i, 0.1)$, resulting in $\frac{k}{2}$ clusters of size N. The second half of clusters is obtained by matrix inversion of the $\frac{k}{2}$ first clusters. This last step ensures that the Fréchet mean of all clusters is the identity matrix, which guarantees that the matrix logarithm provides the most adapted Euclidean approximation.

4 Implementation Details and Results

The whole code was written in Python and run on a personal computer with an Intel i9 processor. The Riemannian Fréchet means were computed using the `geomstats` library [11]. The most compute-intensive functions used across the different frameworks are the matrix logarithm and the matrix determinant. In order to take advantage of the symmetric matrices, the logarithm is not computed using the `linalg` package of the `scipy` library. Instead, we used the `eigh` function from the `linalg` package of the `numpy` library to diagonalize matrices and compute the logarithm of the eigenvalues. Times are computed using the method `perf_counter` of the library `time`.

To assess the efficiency of the four methods, we considered both computation time and the adjusted Rand index (ARI). In essence, ARI quantifies the agreement between two partitions by examining whether each pair of points is placed together or not in both partitions of the same dataset, see [4]. An ARI of 1 indicates perfect concordance between the partitions, whereas an ARI of 0 means their agreement is no better than what would be expected by random clustering. In our case, we compare the initial partition determined by the sampling strategy and the final partition obtained from each method's determined centroids. ARIs were computed by the function `adjusted_rand_score` from the library `scikit-learn`.

For each scenario, we sampled 100 sets of point clouds. For each point cloud, the different frameworks were initialized with the same centroids. The Tables 2 and 3 show the mean values ± and standard deviation of the ARI score, the computation time, and the number of iterations before convergence for each framework. Note that for each framework, the computation times depend both on the computation time of each k-mean iteration and the number of iterations before reaching convergence.

Scenario (ii)

Results show that our approach performs comparably to the Riemannian k-means in both scenarios. Each JensenBregman iteration costs roughly twice as much as a log-Euclidean one because determinants must be recomputed, whereas all the logarithms are cached. When both methods converge in roughly the same number of iterations (scenario (i)), this extra time per iteration makes our total run time longer. In scenario (ii), however, the greater number of iterations required by the log-Euclidean approach compensates for this time difference.

Table 2. Number of clusters: $k = 30$, Points per cluster: 100

Dim.		ARI	Time (s)	Iterations
3×3	R	0.84 ± 0.12	3.04 ± 2.39	15.95 ± 13.79
	LE	0.84 ± 0.12	0.09 ± 0.07	15.52 ± 10.18
	JBLD	0.85 ± 0.12	0.19 ± 0.14	16.88 ± 13.91
	E	0.28 ± 0.07	0.10 ± 0.05	34.40 ± 16.71
6×6	R	0.79 ± 0.17	9.48 ± 9.53	18.60 ± 15.97
	LE	0.79 ± 0.16	0.31 ± 0.27	18.85 ± 13.66
	JBLD	0.80 ± 0.16	0.60 ± 0.46	18.50 ± 13.62
	E	0.31 ± 0.09	0.26 ± 0.16	31.13 ± 14.79

Table 3. Number of clusters: $k = 30$, Points per cluster: 100

Dim.		ARI	Time (s)	Iterations
3×3	R	0.80 ± 0.05	4.23 ± 1.86	21.76 ± 9.32
	LE	0.16 ± 0.01	0.38 ± 0.11	36.08 ± 10.24
	JBLD	0.80 ± 0.06	0.38 ± 0.18	21.19 ± 9.26
	E	0.09 ± 0.01	0.52 ± 0.26	56.05 ± 18.36
6×6	R	0.76 ± 0.07	5.03 ± 1.76	10.93 ± 3.16
	LE	0.25 ± 0.04	1.58 ± 0.54	35.69 ± 9.56
	JBLD	0.76 ± 0.07	0.83 ± 0.29	10.96 ± 3.35
	E	0.13 ± 0.01	1.34 ± 0.56	39.77 ± 10.72

5 Conclusion and Perspectives

In this paper, we introduced a fast, congruence-equivariant k-means algorithm for symmetric positive definite matrices that combines the JensenBregman log-det divergence with the recently introduced log-extrinsic mean. On synthetic datasets sampled from Riemannian Gaussians, our method matches the clustering accuracy of the affine-invariant Riemannian k-means while achieving runtimes on par with the log-Euclidean variant and far below those of the full Riemannian approach.

The procedure is immediately applicable to Hermitian positive definite matrices, making it suitable for complex-valued signal-processing tasks. Future work should investigate performance in higher dimensions, for instance by exploiting stochastic log-det approximations [9].

Acknowledgments. We would like to thank Augustin Chevallier, for the discussions regarding the choice of distribution used for generating clusters, as well as for the implementation of the Metropolis-Hastings algorithm.

References

1. Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Log-euclidean metrics for fast and simple calculus on diffusion tensors. Magnetic Resonance Med. Official J. Inter. Soc. Magnetic Resonance Med. **56**(2), 411–421 (2006)
2. Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Geometric means in a novel vector space structure on symmetric positive-definite matrices. SIAM J. Matrix Anal. Appl. **29**(1), 328–347 (2007)
3. Bhattacharya, R., Patrangenaru, V.: Large sample theory of intrinsic and extrinsic sample means on manifolds. Ann. Stat. **31**(1), 1–29 (2003)
4. Chacón, J.E., Rastrojo, A.I.: Minimum adjusted rand index for two clusterings of a given size. Adv. Data Anal. Classif. **17**(1), 125–133 (2023)
5. Cherian, A., Sra, S., Banerjee, A., Papanikolopoulos, N.: Jensen-bregman logdet divergence with application to efficient similarity search for covariance matrices. IEEE Trans. Pattern Anal. Mach. Intell. **35**(9), 2161–2174 (2012)
6. Chevallier, E., Li, D., Lu, Y., Dunson, D.: Exponential-wrapped distributions on symmetric spaces. SIAM J. Math. Data Sci. **4**(4), 1347–1368 (2022)
7. Chevallier, E., Nielsen, F.: Equivariant log-extrinsic means on irreducible symmetric cones (Jul 2024). https://hal.science/hal-04642282, working paper or preprint
8. Förstner, W., Moonen, B.: A metric for covariance matrices. Geodesy-the Challenge of the 3rd Millennium, pp. 299–309 (2003)
9. Han, I., Malioutov, D., Shin, J.: Large-scale log-determinant computation through stochastic chebyshev expansions. In: International Conference on Machine Learning, pp. 908–917. PMLR (2015)
10. Lenglet, C., Rousson, M., Deriche, R., Faugeras, O.: Statistics on the manifold of multivariate normal distributions: theory and application to diffusion tensor mri processing. J. Math. Imaging Vision **25**, 423–444 (2006)
11. Miolane, N., et al.: Geomstats: a python package for riemannian geometry in machine learning. J. Mach. Learn. Res. **21**(223), 1–9 (2020). http://jmlr.org/papers/v21/19-027.html
12. Moakher, M.: A differential geometric approach to the geometric mean of symmetric positive-definite matrices. SIAM J. Matrix Anal. Appl. **26**(3), 735–747 (2005)
13. Pennec, X., Fillard, P., Ayache, N.: A riemannian framework for tensor computing. Int. J. Comput. Vision **66**, 41–66 (2006)
14. Said, S., Bombrun, L., Berthoumieu, Y., Manton, J.H.: Riemannian gaussian distributions on the space of symmetric positive definite matrices. IEEE Trans. Inf. Theory **63**(4), 2153–2170 (2017)
15. Schwartzman, A.: Lognormal distributions and geometric averages of symmetric positive definite matrices. Int. Stat. Rev. **84**(3), 456–486 (2016)