Parametric information geometry with the package Geomstats

Alice Le Brigant, Jules Deschamps, Antoine Collas, Nina Miolane

PARAMETRIC INFORMATION GEOMETRY WITH THE PACKAGE GEOMSTATS

ALICE LE BRIGANT, JULES DESCHAMPS, ANTOINE COLLAS AND NINA MIOLANE

Abstract. We introduce the information geometry module of the Python package Geomstats. The module first implements Fisher-Rao Riemannian manifolds of widely used parametric families of probability distributions, such as normal, gamma, beta, Dirichlet distributions, and more. The module further gives the Fisher-Rao Riemannian geometry of any parametric family of distributions of interest, given a parameterized probability density function as input. The implemented Riemannian geometry tools allow users to compare, average, interpolate between distributions inside a given family. Importantly, such capabilities open the door to statistics and machine learning on probability distributions. We present the object-oriented implementation of the module along with illustrative examples and show how it can be used to perform learning on manifolds of parametric probability distributions.

1. Introduction

Geomstats [29] is an open-source Python package for statistics and learning on manifolds. Geomstats allows users to analyze complex data that belong to manifolds equipped with various geometric structures, such as Riemannian metrics. This type of data arise in many applications: in computer vision, the manifold of 3D rotations models movements of articulated objects like the human spine or robotics arms [5]; and in biomedical imaging, biological shapes are studied as elements of shape manifolds [15, 41]. The manifolds implemented in Geomstats come equipped with Riemannian metrics that allow users to compute distances and geodesics, among others. Geomstats also provides statistical learning algorithms that are compatible with the Riemannian structures, *i.e.*, that can be used in combination with any of the implemented Riemannian manifolds. These algorithms are geometric generalizations of common estimation, clustering, dimension reduction, classification and regression methods to nonlinear manifolds.

Probability distributions are a type of complex data often encountered in applications: in text classification, multinomial distributions are used to represent documents by indicating words frequencies [25]; in medical imaging, multivariate normal distributions are used to model diffusion tensor images [27]. Many more examples of applications can be found in the rest of this paper. Spaces of probability distributions possess a nonlinear structure that can be captured by two main geometric representations: one provided by optimal transport and one arising from information geometry [2]. In optimal transport, probability distributions are seen as elements of an infinite-dimensional manifold equipped with the Otto-Wasserstein

metric [30, 3]. By contrast, information geometry gives a finite-dimensional manifold representation of parametric families of distributions. [1]

Specifically, information geometry represents the probability distributions of a given parametric family by their parameter space, on which the Fisher information is used to define a Riemannian metric —the so-called *Fisher-Rao metric* or *Fisher information metric* [33]. This metric is a powerful tool to compare and analyze probability distributions inside a given parametric family. It is invariant to diffeomorphic changes of parametrization, and it is the only metric invariant with respect to sufficient statistics, as proved by Cencov [11]. Most importantly, the Fisher-Rao metric comes with Riemannian geometric tools such as geodesics, geodesic distance and intrinsic means, that give an intrinsic way to interpolate, compare, average probability distributions inside a given parametric family. By construction, geodesics and means for the Fisher-Rao metric never leave the parametric family of distributions, contrary to their Wasserstein-metric counterparts. These intrinsic computations can then serve as building blocks to apply learning algorithms to parametric probability distributions.

The geometries of several parametric families have been studied in the literature, and some relate to well-known Riemannian structures: the Fisher-Rao geometry of univariate normal distributions is hyperbolic [8]; the Fisher-Rao geometry of multinomial distributions is spherical [21]; and the Fisher-Rao geometry of multivariante distributions of fixed mean coincides with the affine-invariant metric on the space of symmetric positive definite matrices [31].

Contributions. Computational tools for optimal transport have been proposed, in Python in particular [16]. However, to the best of our knowledge, there exists no wide-ranging open source Python implementation of parametric information geometry, despite a recent implementation in Julia [6]. To fill this gap, this paper presents a module of Geomstats that implements the Fisher-Rao geometries of standard parametric families of probability distributions. Each parametric family of distributions is implemented through its Fisher-Rao manifold with associated exponential and logarithm maps, geodesic distance and geodesics. These manifolds are compatible with the statistical learning algorithms of Geomstats' learning module, which can therefore be applied to probability distributions data. As in the rest of Geomstats, the implementation is object-oriented and extensively unit-tested. All operations are vectorized for batch computation and support is provided for different execution backends — namely NumPy, Autograd, PyTorch, and TensorFlow.

Outline. The rest of the paper is organized as follows. Section 2 provides the necessary background of Riemannian geometry and introduces the structure of Geomstats' information geometry module, *i.e.*, the Python classes used to define a Fisher-Rao geometry. Section 3 details the geometries of the parametric families implemented in the module, along with code illustrations and examples of real-world usecases in the literature. Section 4 presents an application of the information geometry tools of Geomstats to geometric learning on probability distributions. Altogether, the proposed information geometry module represents the first comprehensive implementation of parametric information geometry in Python.

[1] There also exists a non parametric-version that can be defined on the infinite-dimensional space of probability distributions [17], that we do not consider here.

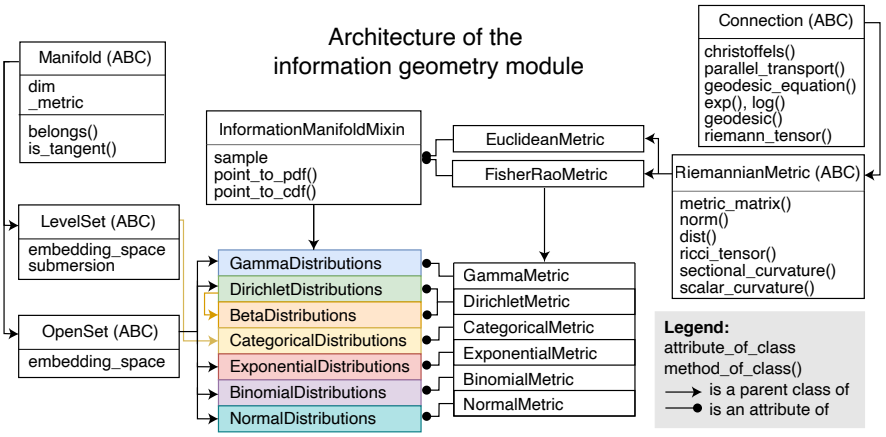

FIGURE 1. Architecture of the information geometry module of `Geomstats`. The `InformationManifold` Python mixin and the `FisherRaoMetric` Python class implement the building blocks of parametric information geometry. The most common parametric families of distributions are Python classes represented in colors, and inherit from the `InformationManifold` mixin. They are equipped with their respective Riemannian metrics, which themselves inherit from the `FisherRaoMetric` class. The abstract (ABC) Python classes `Manifold`, `OpenSet`, `LevelSet`, `Connection`, `RiemannianMetric` provides tools of Riemannian geometry to compute on the information manifolds.

2. INFORMATION GEOMETRY MODULE OF GEOMSTATS

This section describes the design of the `information_geometry` module and its integration into Geomstats. The proposed module implements a Riemannian manifold structure for common parametric families of probability distributions, such as normal distributions, using the object-oriented architecture shown in Fig. 1. The Riemannian manifold structure is encoded by two Python classes: one for the parameter manifold of the family of distributions and one for the Fisher-Rao metric on this manifold. For example, in the case of normal distributions, these Python classes are called `NormalDistributions` and `NormalMetric`. They inherit from more general Python classes, in particular the `Manifold`, `Connection` and `RiemannianMetric` classes. These are abstract classes that define structure, but cannot be instantiated, contrary to their child classes `NormalDistributions` and `NormalMetric`. They also inherit from the `InformationManifold` mixin and the `FisherRaoMetric`: these are Python structures specific to the information geometry module. This section details this architecture along with some theoretical background. For more details on Riemannian geometry, we refer the interested reader to a standard textbook such as [14].

2.1. **Manifold.** The `Manifold` abstract class implements the structure of a *manifold*, *i.e.*, a space that locally resembles a vector space, without necessarily having its global flat structure. Manifolds of dimension d can be defined in an abstract

way, *i.e.*, without considering their embedding in an ambient space, by "gluing" together small pieces of Euclidean space $\mathbb{R}^d$ using charts. We will only consider smooth manifolds, for which the transition from one chart to another is smooth. In addition, submanifolds of a larger Euclidean space $\mathbb{R}^N$ can be defined locally in various ways: *e.g.*, using a parametrization, an implicit function or as the graph of a function [18]. The simplest examples of manifolds are Euclidean spaces $\mathbb{R}^d$, or more generally vector spaces in finite dimensions, open sets of vector spaces (there is only one chart which is the identity) and level sets of functions (defined globally by one implicit function). These important cases are implemented in the abstract classes `VectorSpace`, `OpenSet` and `LevelSet`, which are child classes of `Manifold` as shown in Figure 1.

A d-dimensional manifold M admits a *tangent space* T_xM at each point $x \in M$ that is a d-dimensional vector space. For open sets of $\mathbb{R}^d$, it can be identified with $\mathbb{R}^d$ itself. The classes that inherit from `Manifold` contain methods that allow users to verify that an input is a point belonging to the manifold via the `belongs()` method or that an input is a tangent vector to the manifold at a given base point via the method `is_tangent()` (see Figure 1).

2.2. **Connection.** The `Connection` class implements the structure of an affine connection, which is a geometric tool that defines the generalization of straight lines, addition and subtraction to nonlinear manifolds. To this end, a connection allows us to take derivatives of vector fields, *i.e.*, mappings $V : M \to TM$ that associate to each point p a tangent vector $V(p) \in T_pM$. Precisely, an *affine connection* is a functional ∇ that acts on pairs of vector fields $(U, V) \mapsto \nabla_U V$ according to the following rules: for any vector fields U, V, W and differentiable function f,

$$\nabla_{fU+V} W = f\nabla_U V + \nabla_U W,$$
$$\nabla_U(fV + W) = U(f)V + f\nabla_U V + \nabla_U W,$$

where $U(f)$ denotes the action of the vector field U on the differentiable function f. The action induced by the connection ∇ is referred to as *covariant derivative*.

Geodesics. If $\gamma(t)$ is a curve on M, its velocity $\dot\gamma(t)$ is a vector field along γ, i.e. $\dot\gamma(t) \in T_{\gamma(t)}M$ for all t. The acceleration of a curve is therefore the covariant derivative of this velocity field with respect to the affine connection ∇. A curve γ of zero acceleration

$$\nabla_{\dot\gamma}\dot\gamma = 0, \tag{1}$$

is called a ∇-*geodesic*. Geodesics are the manifolds counterparts of vector spaces' straight lines. Equation (1) translates into a system of ordinary differential equations (ODEs) for the coordinates of the geodesic $\gamma = (\gamma_1, \dots, \gamma_d)$

$$\ddot\gamma_k + \sum_{i,j=1}^{d} \Gamma_{ij}^k(\gamma)\dot\gamma_i\dot\gamma_j = 0, \quad k = 1, \dots, d, \tag{2}$$

where the coefficients Γ_{ij}^k are the *Christoffel symbols* that define the affine connection in local coordinates. In the `Connection` class, Equation (2) is implemented in the `geodesic_equation()` method and the Christoffel symbols are implemented in the `christoffels()` method (see Figure 1).

Exp and Log maps. Existence results for solutions of ODEs allow us to define geodesics starting at a point x with velocity $v \in T_xM$ for times t in a neighborhood of zero, or equivalently for all time $t \in [0,1]$ but for tangent vectors v of small norm. The *exponential map* at $x \in M$ associates to any $v \in T_xM$ of sufficiently small norm the end point $\gamma(1)$ of a geodesic γ starting from θ with velocity v:

$$\exp_x(v) = \gamma(1), \quad \text{where} \begin{cases} \gamma \text{ is a geodesic,} \\ \gamma(0) = x,\ \dot{\gamma}(0) = v. \end{cases}$$

If B is a small ball of the tangent space T_xM centered at 0 on which $\exp_x$ is defined, then $\exp_x$ is a diffeomorphism from B onto its image and its inverse $\log_x \equiv \exp_x^{-1}$ defines the *logarithm map*, which associates to any point y the velocity $v \in T_xM$ necessary to get to y when departing from x:

$$\log_x(y) = v \quad \text{where} \quad \exp_x(v) = y.$$

The exponential and logarithm maps can be seen as generalizations of the Euclidean addition and subtraction to nonlinear manifolds. Both maps are implemented in the `exp()` and `log()` methods of the `Connection` class, which further allow us to get other tools such as `parallel_transport()` (see Figure 1). We refer to [18] for additional details on the `Connection` class.

2.3. **Riemannian metric.** Just like there is an abstract Python class that encodes the structure of manifolds, the abstract class `RiemannianMetric` encodes the structure of Riemannian metrics. A *Riemannian metric* is a collection of inner products $(\langle \cdot, \cdot \rangle_p)_{p \in M}$ defined on the tangent spaces of a manifold M, that depend on the base point $p \in M$ and varies smoothly with respect to it.

Levi-Civita Connection. A Riemannian metric is associated with a unique affine connection, called the *Levi-Civita connection*, which is the only affine connection that is symmetric and compatible with the metric, *i.e.*, that verifies

$$UV - VU = \nabla_U V - \nabla_V U$$
$$U\langle V, W \rangle = \langle \nabla_U V, W \rangle + \langle V, \nabla_U W \rangle$$

for all vector fields U, V, W. The geodesics of a Riemannian manifold are those of its Levi-Civita connection. The class `RiemannianMetric` is therefore a child class of `Connection` and inherits all its methods, including `geodesic()`, `exp()` and `log()`. The class `RiemannianMetric` overwrites the `Connection` class' method `christoffels()` and computes the Christoffel symbols using derivatives of the metric. The geodesics, by the compatibility property, have velocity of constant norm, *i.e.*, are parametrized by arc length.

Geodesic Distance. The `dist()` method implements the geodesic distance induced by the Riemannian metric, defined between two points $x, y \in M$ to be the length of the shortest curve linking them, where the length of a (piecewise) smooth curve $\gamma : (0,1) \to M$ is computed by integrating the norm of its velocity

$$d(x,y) = \inf_{\gamma; \gamma(0)=x, \gamma(1)=y} L(\gamma), \quad \text{where} \quad L(\gamma) = \int_0^1 ||\dot{\gamma}(t)||_{\gamma(t)} dt,$$

using the norm induced by the Riemannian metric. In a Riemannian manifold, geodesics extend another property of straight lines: they are locally length-minimizing.

In a geodesically complete manifold, any pair of points can be linked by a minimizing geodesic, not necessarily unique, and the `dist()` can be computed using the `log` map:

$$\forall x, y \in M, \quad d(x,y) = ||\log_x(y)||_x.$$

Curvatures. Finally, different notions of curvature are implemented, including the `riemann_curvature()` tensor and `sectional_curvature()`, among others (see Figure 1). The Riemann curvature tensor is defined from the connection, namely for any vector fields U, V, W as $R(U,V)W = \nabla_{[U,V]}W + \nabla_V\nabla_U W - \nabla_U\nabla_V W$. Sectional curvature at $x \in M$ is a generalization of the Gauss curvature of a surface in $\mathbb{R}^3$. It is defined for any two-dimensional subspace $\sigma(u,v) \subset T_xM$ spanned by tangent vectors u, v, as

$$K_{\sigma(u,v)}(x) = \frac{\langle R(u,v)v, u\rangle}{\langle u,u\rangle\langle v,v\rangle - \langle u,v\rangle^2}.$$

It yields important information on the behavior of geodesics, since a geodesically complete and simply connected manifold with everywhere negative sectional curvature (a *Hadamard manifold*) is globally diffeomorphic to $\mathbb{R}^d$ through the exponential map. Consequently, negatively curved spaces share some of the nice properties of Euclidean spaces: any two points can be joined by a unique minimizing geodesic, the length of which gives the geodesic distance.

2.4. **Information manifold.** The proposed `information_geometry` module is integrated into the differential geometry structures implemented in Geomstats. The module contains child classes of `Manifold` that represent parametric families of probability distributions, and child classes of `RiemannianMetric` that define the Fisher information metric on these manifolds. The combination of two such classes define what we call an *information manifold,* which is specified by an inheritance from the mixin: `InformationManifoldMixin` shown in Figure 1.

Parameter Manifolds. Specifically, consider a family of probability distributions on a space $\mathcal{X}$, typically $\mathcal{X} = \mathbb{R}^n$ for some integer n. Assume that the distributions in the family are absolutely continuous with respect to a reference measure λ (such as the Lebesgue measure on $\mathbb{R}^n$) with densities

$$f(x|\theta), \quad x \in \mathcal{X}, \theta \in \Theta,$$

with respect to λ, where θ is a parameter belonging to Θ an open subset of $\mathbb{R}^d$. Then, this parametric family is represented by the *parameter manifold* Θ. The `information_geometry` module implements this manifold as a child class of one of the abstract classes `OpenSet` and `LevelSet`, which are themselves children of `Manifold`. Most of the parameter manifolds are implemented as child classes of `OpenSet` as shown in Figure 1. Other parameter manifolds are implemented more easily with another class. This is the case of `CategoricalDistributions`, which inherits from `LevelSet` as its parameter space is the interior of the simplex.

Information Manifolds. Parameter manifolds also inherit from the mixin class, called `InformationManifoldMixin`, which turns them into *information manifolds.* First, this mixin endows them with specific methods such as `sample()`, which returns a sample of the distribution associated to a given parameter $\theta \in \Theta$, or `point_to_pdf()`, which returns the probability density function (or probability mass function) associated to a given parameter $\theta \in \Theta$ (see Figure 1).

For example, to generate at random a categorical distribution on a space of 5 outcomes, we instantiate an object of the class `CategoricalDistributions` with dimension 4 using `manifold = CategoricalDistributions(4)` and define `parameter = manifold.random_point()`. Then, in order to sample from this distribution, one uses `samples = manifold.sample(parameter, n_samples=10)`.

Second, the `InformationManifoldMixin` endows the parameter manifolds with a Riemannian metric defined using the Fisher information, called the *Fisher-Rao metric* and implemented in the `FisherRaoMetric` class shown in Figure 1. The Fisher information is a notion from statistical inference that measures the quantity of information on the parameter θ contained in an observation with density $f(\cdot, \theta)$. It is defined, under certain regularity conditions [26], as

$$I(\theta) = -\mathbb{E}_\theta \left[\mathrm{Hess}_\theta \left(\log f(X|\theta)\right)\right], \tag{3}$$

where Hess_θ denotes the hessian with respect to θ and $\mathbb{E}_\theta$ is the expectation taken with respect to the random variable X with density $f(\cdot, \theta)$. If this d-by-d matrix is everywhere definite, it provides a Riemannian metric on Θ, called the Fisher-Rao metric, where the inner product between two tangent vectors u, v at $\theta \in \Theta$ is defined by

$$\langle u, v\rangle_\theta = u^\top I(\theta) v. \tag{4}$$

Here the tangent vectors u, v are simply vectors of $\mathbb{R}^d$ since Θ is an open subset of $\mathbb{R}^d$. In the sequel, we will describe the Fisher-Rao metric for different parametric statistical families by providing the expression of the infinitesimal length element

$$ds^2 = \langle d\theta, d\theta\rangle_\theta = d\theta^\top I(\theta) d\theta$$

The metric matrix I is implemented using automatic differentiation in the `FisherRaoMetric` class. This allows users to get the Fisher-Rao Metric of any parametric family of probability distributions, for which the probability density function is known. For example, a user can compute the Fisher-Rao metric of the normal distributions with the syntax given below, which uses automatic differentiation behind the scenes.

```
class MyInformationManifold(InformationManifoldMixin):
    def __init__(self):
        self.dim = 2
    def point_to_pdf(self, point):
        means = point[..., 0]
        stds = point[..., 1]
        def pdf(x):
            constant = (1. / gs.sqrt(2 * gs.pi * stds**2))
            return  constant * gs.exp(-((x - means) ** 2) / (2 * stds**2))
        return pdf

metric = FisherRaoMetric(
    information_manifold=MyInformationManifold(), support=(-10, 10))
```

The user can then access the Fisher-Rao metric matrix $I(\theta)$ at $\theta = (1., 1.)$ with the code below.

```
print(metric.metric_matrix(gs.array([1., 1.])))
>>> array([[1.00000000e+00, 1.11022302e-16],
       [1.11022302e-16, 2.00000000e+00]])
```

We recognize here the metric matrix of the Fisher-Rao metric on the univariate normal distributions. For convenience, the Fisher-Rao metrics for well-known parameter manifolds are already implemented in classes such as `NormalMetric`, `GammaMetric`, `CategoricalMetric`, etc, as shown in Figure 1. These classes implement the closed-forms of the Fisher-Rao metric when these are known. The corresponding parameter manifolds in the classes `NormalDistributions`, `GammaDistributions`, `CategoricalDistributions`, etc, are equipped with their Fisher-Rao metric, which is found as an attribute called `metric`.

For example, the Fisher-Rao metric on the categorical distributions on a support of cardinal 5 is found in the `metric` attribute of the class of categorical distributions, i.e. `metric = CategoricalDistributions(4).metric`. Its methods allow to compute exponential, logarithm maps and geodesics using `metric.exp()`, `metric.log()`, `metric.geodesic()`, together with the various notions of curvatures.

3. Information manifolds implemented in Geomstats

This section details the tools of information geometry that we implement in each of the information manifold classes. As such, this section also provides a comprehensive review of the field of computational information geometry and its main applications. Each subsection further showcases code snippets using each information manifold to demonstrate the diversity of use cases of the proposed `information_manifold` module.

3.1. **One-dimensional parametric families.**

3.1.1. *Main results.* The information geometry of one-dimensional information manifolds is simple: there is no curvature, the parameter manifold Θ is always diffeomorphic to $\mathbb{R}$, and there is only one path to go from one point to another in Θ. However, the parametrization of this path can vary and leads to different interpolations between the probability distribution functions, as seen in Figure 2.

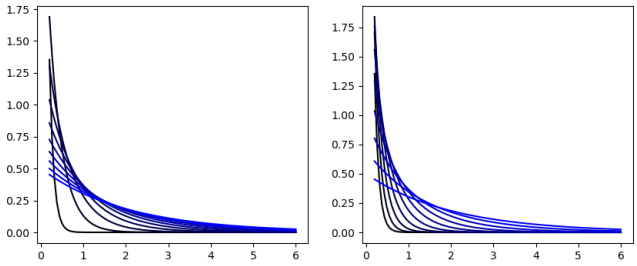

Figure 2. Comparison between affine (left) and geodesic (right) interpolations between pdfs of exponential distributions of parameter $\lambda_0 = 0.1$ (black) and $\lambda_1 = 2$ (blue).

The Fisher-Rao geodesic distances are given in closed forms for the Poisson, exponential, binomial (and Bernoulli) distributions in [8]. We compute it for geometric distributions too (see the appendix). Results are summarized in Table 1 and implemented in the `dist()` methods of the corresponding metric classes.

Distribution	P.d.f. (or P.m.f.)	Geodesic distance
Poisson (mean λ)	$\forall k \in \mathbb{N},\ P(k\|\lambda) = \frac{\lambda^k}{k!}e^{-\lambda},\ \lambda > 0$	$d(\lambda_1, \lambda_2) = 2\|\sqrt{\lambda_1} - \sqrt{\lambda_2}\|$
Exponential (mean $\frac{1}{\lambda}$)	$\forall x \geq 0, f(x\|\lambda) = \lambda e^{-\lambda x},\ \lambda > 0$	$d(\lambda_1, \lambda_2) = \|\log \frac{\lambda_1}{\lambda_2}\|$
Binomial (known index n)	$\forall k \in \{0, ..., n\},\ P(k\|p) = \binom{n}{k} p^k(1-p)^{n-k},\ 0 < p < 1$	$d(p_1, p_2) = 2\sqrt{n}\|\sin^{-1}(\sqrt{p_1}) - \sin^{-1}(\sqrt{p_2})\|$
Bernoulli (1-binomial)	$\forall k \in \{0, 1\},\ P(k\|p) = p^k(1-p)^{1-k},\ 0 < p < 1$	$d(p_1, p_2) = 2\|\sin^{-1}(\sqrt{p_1}) - \sin^{-1}(\sqrt{p_2})\|$
Geometric	$\forall k \in \mathbb{N}^*,\ P(k\|p) = (1-p)^{k-1}p,\ 0 < p < 1$	$d(p_1, p_2) = 2\|\tanh^{-1}(\sqrt{1-p_1}) - \tanh^{-1}(\sqrt{1-p_2})\|$

TABLE 1. Fisher-Rao distance for one-dimensional parametric families of probability distributions implemented in the information geometry module. P.d.f. means probability density function and P.m.f. means probability mass function. These formulas are implemented in the `dist()` methods in the metric Python classes of Figure 1.

3.1.2. *`Geomstats` example.* The following code snippet shows how to compute the middle of the geodesic between points $p_1 = .4$ and $p_2 = .7$ on the one-dimensional 5-binomial manifold.

```
import geomstats.backend as gs
from geomstats.information_geometry.binomial import BinomialDistributions

manifold = BinomialDistributions(5)

point_a = .4
point_b = .7

times = gs.linspace(0, 1, 100)
geodesic = manifold.metric.geodesic(initial_point=point_a, end_point=point_b)(times)

middle = geodesic(.5)
print(middle)

>>>  0.5550055679356352
```

The geodesic middle point of $p_1 = .4$ and $p_2 = .7$ on the 5-binomial manifold is roughly $p = .555$, a little higher than the Euclidean middle point (=.55)!

3.2. **Multinomial and categorical distributions.**

3.2.1. *Main results. Multinomial distributions* model the results of an experiment with a finite number k of outcomes, repeated n times. When there is no repetition ($n = 1$), it is called a *categorical distribution.* Here the number of repetitions n is always fixed. The parameter θ of the parameter manifold encodes the probabilities of the different outcomes. The parameter manifold Θ is therefore the interior of the $k-1$ dimensional simplex $\Theta = \Delta_{k-1} = \{\theta \in \mathbb{R}^k : \forall i, \theta_i > 0, \theta_1 + \ldots + \theta_k = 1\}$.

Definition 3.2.1.1 (Probability mass function of the multinomial distribution)**.** Given $k, n \in \mathbb{N}^*$ and $\theta = (\theta_1, ..., \theta_k) \in \Delta_{k-1}$, the p.m.f. of the n-multinomial distribution of parameter θ is

$$p(x = (x_1, \ldots, x_k)|\theta) = \frac{n!}{x_1! \ldots x_k!}\theta_1^{x_1} \ldots \theta_k^{x_k},$$

where $x_i \in \{0, \ldots, n\}$ for all $i = 1, \ldots, k$ and $x_1 + \ldots + x_k = n$.

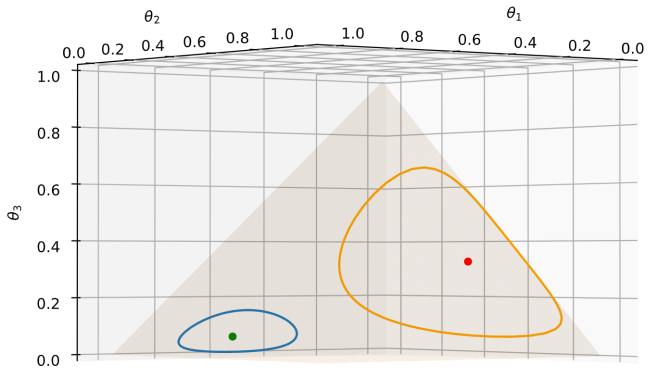

FIGURE 3. Information geometry of the 3-Categorical manifold implemented in the Python class `CategoricalDistributions`. The orange geodesic ball is of radius 0.7 and centered on the red point $(0.1, 0.58, 0.32)$, the blue geodesic ball is of radius 0.3 and centered on the green point $(0.74, 0.21, 0.05)$.

The Fisher-Rao geometry on the parameter manifold Δ_{k-1} is well-known, see for example [21]. We summarize the geometry with the following propositions.

Proposition 3.2.1.1 (Fisher-Rao metric on the multinomial manifold)**.** *The Fisher-Rao metric on the parameter manifold $\Theta = \Delta_{k-1}$ of n-multinomial distributions is given by*

$$ds^2 = n\left(\frac{d\theta_1^2}{\theta_1} + ... + \frac{d\theta_k^2}{\theta_k}\right).$$

Thus, one can see that the Fisher-Rao metric on the parameter manifold $\Theta = \Delta_{k-1}$ of multinomial distributions can be obtained as the pullback of the Euclidean metric on the positive $(k-1)$-sphere of radius $2\sqrt{n}$, $S_{k-1}^+ = \{\theta \in \mathbb{R}^k : \forall i, \theta_i > 0, \sum_{i=1}^k \theta_i^2 = 2\sqrt{n}\}$ by the diffeomorphism

$$R : \theta \mapsto R(\theta) = (2\sqrt{n\theta_1}, ..., 2\sqrt{n\theta_k}).$$

Therefore the distance between two given parameters is the spherical distance of their images by transormation R, and the curvature of the parameter manifold is that of the $(k-1)$-sphere of radius $2\sqrt{n}$.

Proposition 3.2.1.2 (Geodesic distance on the multinomial manifold)**.** *The geodesic distance between two parameters $\theta^1, \theta^2 \in \Delta_{k-1}$ has the following analytic expression:*

$$d(\theta^1, \theta^2) = 2\sqrt{n}\arccos\left(\sum_{i=1}^k \sqrt{\theta_i^1 \theta_i^2}\right).$$

Proposition 3.2.1.3 (Curvature of the multinomial manifold)**.** *The Fisher-Rao manifold of multinomial distributions has constant sectional curvature $K = 2\sqrt{n}$.*

We implement the p.m.f, Fisher-Rao metric, geodesic distance, and curvatures in the Python classes `MultinomialDistributions` and `MultinomialMetric` of the `information_geometry` module.

3.2.2. *Applications.* The Fisher-Rao geometry of multinomial distributions has been used in the literature, *e.g.*, to formulate concepts in evolutionary game theory [19] and to classify documents after term-frequency representation in the simplex [25].

3.2.3. *`Geomstats` example.* This example shows how we use the `information_geometry` module to compute on the 6-categorical manifold, *i.e.*, the 5-dimensional manifold of categorical distributions with $k = 6$ outcomes. The following code snippet computes the geodesic distances between a given point on the 6-categorical manifold and the vertices of the simplex Δ_5.

```
import geomstats.backend as gs
from geomstats.information_geometry.categorical import CategoricalDistributions

manifold = CategoricalDistributions(dim=5)

point_a = gs.array([.1, .2, .1, .3, .15, .15])
point_b = gs.array([.25, .25, .1, .05, .05, .3])

vertices = list(gs.eye(6))

distances_a = [manifold.metric.dist(point_a, extremity) for vertex in vertices]
distances_b = [manifold.metric.dist(point_b, extremity) for vertex in vertices]

print(f"distances_a = {[float(str(distance)[:5]) for distance in distances_a]}")
print(f"distances_b = {[float(str(distance)[:5]) for distance in distances_b]}")

>>> distances_a = [2.498, 2.214, 2.498, 1.982, 2.346, 2.346]
>>> distances_b = [2.094, 2.094, 2.498, 2.69, 2.69, 1.982]

closest_a = vertices[gs.argmin(distances_a)]
closest_b = vertices[gs.argmin(distances_b)]

print(f"closest extremity to {point_a} is {closest_a}")
print(f"closest extremity to {point_b} is {closest_b}")

>>> closest extremity to [0.1  0.2  0.1  0.3  0.15 0.15] is [0. 0. 0. 1. 0. 0.]
>>> closest extremity to [0.25 0.25 0.1  0.05 0.05 0.3 ] is [0. 0. 0. 0. 0. 1.]
```

This result confirms the intuition that the vertex of the simplex that is closest, in terms of the Fisher-Rao geodesic distance, to a given categorical distribution is the one corresponding to its mode. Indeed, noting $e_i = (\delta_{ij})_j$, $i = 1, \dots, 6$ the extremities of the simplex, we see that for all $i \in \{1, \dots, 6\}$ and $\theta \in \Delta_5$, $d(\theta, e_i) = \arccos(\sqrt{\theta_i})$ is minimal when i matches the mode of the distribution.

3.3. **Normal distributions.** Normal distributions are ubiquitous in probability theory and statistics, especially via the Central limit theorem. They are a very widely used modelling tool in practice, and provide one of the first non trivial Fisher-Rao geometries to be studied in the literature.

3.3.1. *Main results.* Let us start by reviewing the univariate normal model.

Definition 3.3.1.1 (Probability density function of the univariate normal distribution)**.** The p.d.f. of the normal distribution of mean $m \in \mathbb{R}$ and variance $\sigma^2 \in \mathbb{R}_+^*$

is

$$f(x|\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right).$$

It is well known since the 1980s [8] that the corresponding Fisher-Rao metric with respect to $\theta = (m, \sigma)$ defines hyperbolic geometry on the parameter manifold $\Theta = \mathbb{R} \times \mathbb{R}_+^*$.

Proposition 3.3.1.1 (Fisher-Rao metric for the univariate normal manifold)**.** *The Fisher-Rao metric on the parameter manifold $\Theta = \mathbb{R} \times \mathbb{R}_+^*$ of normal distributions is*

$$ds^2 = \frac{dm^2 + 2d\sigma^2}{\sigma^2}.$$

Indeed, using the change of variables $m \mapsto m/\sqrt{2}$, we retrieve a multiple of the Poincaré metric $ds^2 = 2(dx^2 + dy^2)/y^2$ on the upper half-plane $\{(x, y) : x \in \mathbb{R}, y > 0\}$, a model of two-dimensional hyperbolic geometry. Thus, closed-form expressions are known for the geodesics, which are either vertical segments or portions of half-circles orthogonal to the m-axis. The same is true for the distance.

Proposition 3.3.1.2 (Geodesic distance on the univariate normal manifold [38])**.** *The geodesic distance between normal distributions of parameters (m_1, σ_1) and (m_2, σ_2) in $\mathbb{R} \times \mathbb{R}_+^*$ is given by*

$$d((m_1, \sigma_1), (m_2, \sigma_2)) = \sqrt{2} \cosh^{-1}\left(\frac{(m_1 - m_2)^2/2 + (\sigma_1 + \sigma_2)^2}{2\sigma_1\sigma_2}\right).$$

The curvature is the same as that of the 2-Poincaré metric, and rescaling the Poincaré metric by a factor 2 implies dividing the sectional curvature by the same factor. The manifold of univariate normal distributions has therefore constant negative curvature, and since it is simply connected and geodesically complete we get the following result.

Proposition 3.3.1.3 (Curvature of the univariate normal manifold [38].)**.** *The Fisher-Rao manifold of normal distributions has constant sectional curvature $K = -1/2$. In particular, any two normal distributions can be linked by a unique geodesic, the length of which gives the Fisher-Rao distance.*

We implement the p.d.f, Fisher-Rao metric, geodesics, geodesic distance, and curvatures in the Python classes `NormalDistributions` and `NormalMetric` of the `information_geometry` module. Figure 4 shows 2 geodesics, 2 geodesic spheres, and 1 geodesic grid on the information manifold of univariate normal distributions.

We now turn to the multivariate case.

Definition 3.3.1.2 (Probability density function of multivariate normal distributions)**.** In higher dimensions $p \geq 2$, the p.d.f. of the normal distribution of mean $m \in \mathbb{R}^p$ and covariance matrix $\Sigma \in S_p(\mathbb{R})^+$ is

$$f(x|\theta) = \frac{1}{\sqrt{(2\pi)p|\Sigma|}} \exp\left(-\frac{1}{2}(x-m)^\top \Sigma^{-1}(x-m)\right).$$

The Fisher-Rao geometry of multivariate normal distributions was first studied in the early 1980's [35], [8][38]. In general, no closed form expressions are known for the distance nor the geodesics associated to the Fisher information metric in the multivariate case. However, analytic expressions for these quantities are known

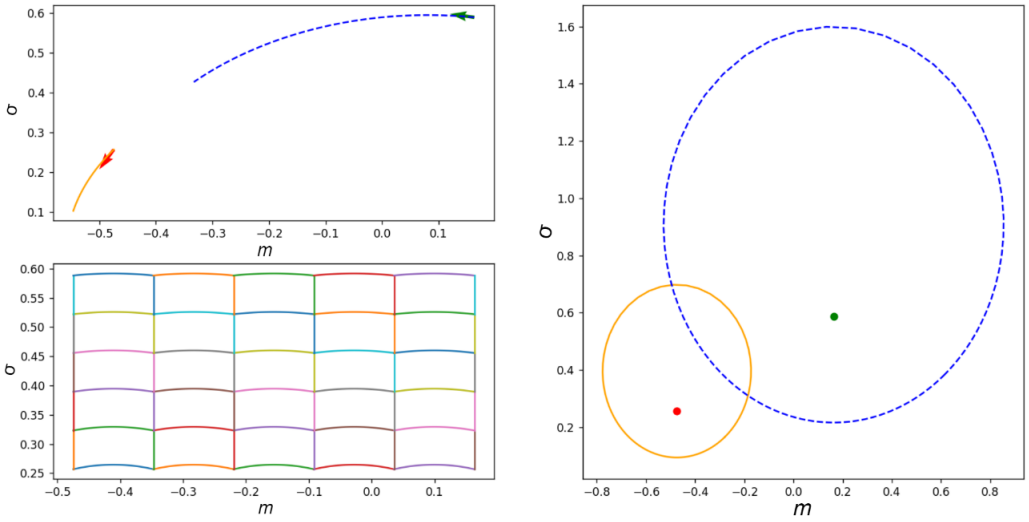

FIGURE 4. Information geometry of the manifold of normal distributions implemented in the Python class `NormalDistributions`. Up-left: two geodesics of length 1 departing from two random points A and B; Bottom-left: geodesic grid between A and B. Right: two geodesic spheres of radius 1 centered on A and B

for some particular submanifolds, and can be found e.g. in the review paper [32]. The first of these particular cases corresponds to multivariate distributions with diagonal covariances.

Proposition 3.3.1.4 (Multivariate normal distributions with diagonal covariance matrices [38])**.** *The submanifold of Gaussian distributions with mean* $m = (m_1, \dots, m_p)$ *and diagonal covariance matrix* $\Sigma = diag(\sigma_1^2, ..., \sigma_p^2)$ *can be identified with the product manifold* $(\mathbb{R} \times \mathbb{R}_+^*)^p = \{(m_1, \sigma_1, \dots, m_p, \sigma_p) : m_i \in \mathbb{R}, \sigma_i > 0\}$*, on which the Fisher-Rao metric is the product metric*

$$ds^2 = \sum_{i=1}^{p} \frac{dm_i^2 + 2d\sigma_i^2}{\sigma_i^2}.$$

The induced geodesic distance bertween distributions of means $m_j = (m_{ji})_{1 \leq i \leq p}$ *and covariance matrices* $\Sigma_j = \mathrm{diag}(\sigma_{j1}^2, \dots, \sigma_{jp}^2)$*,* $j = 1, 2$*, is given by*

$$d_p((m_1, \Sigma_1), (m_2, \Sigma_2)) = \sqrt{\sum_{i=1}^{p} d^2((m_{1i},\, \sigma_{1i}), (m_{2i}, \sigma_{2i}))},$$

where d *is the geodesic distance on the space of univariate normal distributions.*

The second particular case when the geometry is explicit corresponds to multivariate normal distributions with fixed mean. In this case, the parameter space is the space of symmetric positive definite matrices and the Fisher-Rao metric coincides with the affine-invariant metric [31]. Note that even though the parameter with respect to which the Fisher information is computed differs between the different submanifolds of the multivariate normal distributions, this does not affect the

distance, which is invariant with respect to diffeomorphic change of parametrization.

Proposition 3.3.1.5 (Multivariate normal distributions with fixed mean [8])**.** *Let* $\mathbf{m} \in \mathbb{R}^p$*. The geodesic distance between Gaussian distributions with fixed mean* $\mathbf{m}$ *and covariance matrices* Σ_1*,* Σ_2 *is*

$$d(\Sigma_1, \Sigma_2) = \sqrt{\frac{1}{2}\sum_{i=1}^{p} \log(\lambda_i)^2},$$

where the λ_j *are the eigenvalues of* $(\Sigma_1)^{-1}\Sigma_2$*.*

The sectional curvature in the fixed mean case is negative, although non constant [27]. We implement the information geometry of the normal distributions reviewed here within the Python classes `NormalDistributions` and `NormalMetric` shown in Figure 1.

3.3.2. *Applications.* The Fisher-Rao geometry of normal distributions has proved very useful in the field of diffusion tensor imaging [27] and more generally in image analysis, *e.g.*, for detection [28], mathematical morphology [4] and segmentation [40, 39]. We refer the interested reader to the review paper [32] and the references therein.

3.3.3. ***`Geomstats`*** *example.* This example shows how users can leverage the proposed `information_geometry` module to get intuition on the Fisher-Rao geometry of normal distributions. Specifically, we compute the geodesics and geodesic distance between two normal distributions with same variance and different means $m_1 = 1, m_2 = 4$, for two different values $\sigma^2 = 1, \sigma'^2 = 4$ of the common variance.

```
import matplotlib.pyplot as plt
import geomstats.backend as gs
from geomstats.information_geometry.beta import BetaDistributions
from geomstats.information_geometry.normal import NormalDistributions

manifold = NormalDistributions()
point_a = gs.array([1., 1.])
point_b = gs.array([4., 1.])
point_c = gs.array([1., 2.])
point_d = gs.array([4., 2.])

print(manifold.metric.dist(point_a, point_b))
print(manifold.metric.dist(point_c, point_d))

>>> 2.38952643457422
>>> 1.3862943611198915

times = gs.linspace(0, 1, 100)
geod_ab = manifold.metric.geodesic(initial_point=point_a, end_point=point_b)(times)
geod_cd = manifold.metric.geodesic(initial_point=point_c, end_point=point_d)(times)

max_variance_ab = geodesic_ab[gs.argmax(geod_ab[:, 1])]
max_variance_cd = geodesic_cd[gs.argmax(geod_cd[:, 1])]

plt.plot(*gs.transpose(geod_ab))
plt.scatter(*point_a, color='g')
```

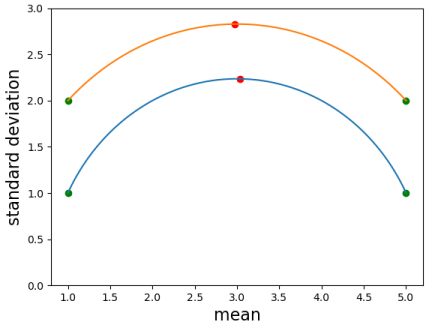

FIGURE 5. Geodesics in the manifold of normal distributions. When the variance of the normal distributions at the extremities (green points) increases, the geodesic becomes shorter. Variance increases along the geodesic and reaches a maximum in the middle (red points).

```
plt.scatter(*point_b, color='g')
plt.scatter(*max_variance_ab, color='r')
plt.plot(*gs.transpose(geod_cd))
plt.scatter(*point_c, color='g')
plt.scatter(*point_d, color='g')
plt.scatter(*max_variance_cd, color='r')
plt.ylim([0., 3.])
plt.show()
```

The two geodesics generated by this code snippet yield the two curves in Figure 5. We see that the higher the variance, the smaller the distance. As pointed out in [13], this result reflects the fact that the p.d.f.s overlap more when the variance increases. On each geodesic, we observe that the point of maximum variance corresponds to the geodesic' middle point.

3.4. **Gamma distributions.** Gamma distributions form a 2-parameter family of distributions defined on the positive half-line, and are used to model the time between independent events that occur at a constant average rate. They have been widely used to model right-skewed data, such as cancer rates [37], insurance claims [36], and rainfall [20].

3.4.1. *Main results.* Standard Gamma distributions take support over $\mathbb{R}_+^*$ and consist of one of the prime examples of information geometry, namely for for the variety of parametrizations they have been endowed with [22], [10], [7].

Definition 3.4.1.1 (Probability density function for Gamma distributions in natural coordinates)**.** In natural coordinates, given $(\nu, \kappa) \in \left(\mathbb{R}_+^*\right)^2$, the p.d.f. of the two-parameter Gamma distribution of rate ν and shape κ is:

$$\forall x > 0,\ f(x|\nu,\kappa) = \frac{\nu^\kappa}{\Gamma(\kappa)} x^{\kappa-1} e^{-\nu x}, \text{ where } \Gamma \text{ is the Gamma function.}$$

Proposition 3.4.1.1 (Fisher-Rao metric for the Gamma manifold in natural coordinates [7])**.** *The Fisher-Rao metric on the Gamma manifold* $\Theta = \left(\mathbb{R}_+^*\right)^2$ *is*

$$ds^2 = \frac{\kappa}{\nu^2} d\nu^2 - 2\frac{d\nu d\kappa}{\nu} + \psi'(\kappa) d\kappa^2,$$

where ψ *is the digamma function, i.e.* $\psi = \frac{\Gamma'}{\Gamma}$.

However, the fact that this metric is not diagonal for the natural parametrization encourages one to consider the manifold under a different set of coordinates. Getting rid of the middle term in $d\nu d\kappa$ highly simplifies the geometry.

Proposition 3.4.1.2 (Fisher-Rao metric for the Gamma manifold in (γ, κ) coordinates [7])**.** *The change of variable* $(\gamma, \kappa) = (\frac{\kappa}{\nu}, \kappa)$ *gives the following expression of the Fisher-Rao metric:*

$$ds^2 = \frac{\kappa}{\gamma^2} d\gamma^2 + \left(\psi'(\kappa) - \frac{1}{\kappa}\right) d\kappa^2.$$

Both parametrizations (γ, κ) and (κ, γ) can be found in the literature. The use of of (κ, γ) is standard in information geometry and it is the one we use to implement the Gamma manifold. This yields the following expression of the p.d.f.

Definition 3.4.1.2 (Probability density function for Gamma distributions in (κ, γ) coordinates)**.** The p.d.f. of the two-parameter Gamma distribution of parameters γ, κ is:

$$\forall x > 0,\ f(x|\gamma, \kappa) = \frac{\kappa^\kappa}{\gamma^\kappa \Gamma(\kappa)} x^{\kappa-1} e^{-\frac{\kappa x}{\gamma}}.$$

Proposition 3.4.1.3 (Geodesic equations on the Gamma manifold [7])**.** *The associated geodesic equations are:*

$$\begin{cases} \ddot{\gamma} = \frac{\dot{\gamma}^2}{\gamma} - \frac{\dot{\gamma}\dot{\kappa}}{\kappa} \\ \ddot{\kappa} = \frac{\kappa\dot{\gamma}^2}{2\gamma^2(\kappa\psi'(\kappa)-1)} - \frac{(\psi"(\kappa)\kappa^2+1)\dot{\kappa}^2}{2\kappa(\kappa\psi'(\kappa)-1)}. \end{cases}$$

No closed form expressions are known for the distance nor the geodesics associated to the Fisher information geometry with respect to (γ, κ). Yet, our information module is able to compute both numerically by leveraging the automatic differentiation computations available in the parent Python class of the `FisherRaoMetric`. Figure 6 shows 3 geodesics, 2 geodesic spheres, and a geodesic grid for the Gamma manifold. Running code from the information geometry module shows that some geodesics are horizontal (with γ constant), which is notable. This can also be directly seen from the geodesic equation $\ddot{\gamma} = \dot{\gamma}\left(\frac{\dot{\gamma}}{\gamma} - \frac{\dot{\kappa}}{\kappa}\right)$: a geodesic with a horizontal initial direction ($\dot{\gamma} = 0$) will stay horizontal.

There is a closed-form expression of the geodesic distance in the manifold of Gamma distributions with fixed κ, which is therefore a one-dimensional manifold.

Proposition 3.4.1.4 (Geodesic distance on the Gamma manifold with fixed κ.)**.** *The geodesic distance* d *on the Gamma manifold, for a fixed* κ *is given in* (κ, γ) *parameterization by:*

$$\forall \gamma_1, \gamma_2 > 0,\ d(\gamma_1, \gamma_2) = \sqrt{\kappa} \left| \log \frac{\gamma_1}{\gamma_2} \right|,$$

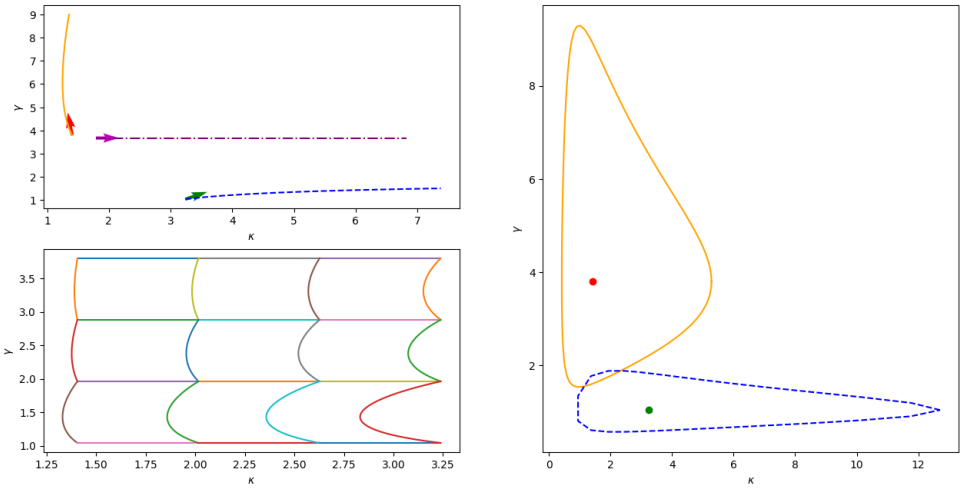

FIGURE 6. Information geometry of the manifold of Gamma distributions implemented in the Python class `GammaDistributions`. Up-left: three geodesics of length 1 departing from two random points A (red) and B (green) and C (magenta, with γ constant). Bottom-left: geodesic grid between A and B. Right: two geodesic spheres of radius 1 centered on A and B;

or, in (κ, ν) *parameterization by:*

$$\forall \gamma_1, \gamma_2 > 0,\ d(\nu_1, \nu_2) = \sqrt{\kappa} \left| \log \frac{\nu_1}{\nu_2} \right|.$$

This result, proved in the appendix, was expected, at least for integer values of κ. Consider one Gamma process as the sum of κ i.i.d exponential processes. Because the processes are independent, the Fisher information for the Gamma distribution is κ times as big as that of the exponential distribution. Consequently, the length of a geodesic on the Gamma manifold observes a $\sqrt{\kappa}$ coefficient.

The sectional curvature of the Gamma manifold, which is plotted in Figure 7, is everywhere negative, bounded and depends only on the κ parameter. Since it is also simply connected and geodesically complete, the following result holds.

Proposition 3.4.1.5 (Curvature of the Gamma manifold [10])**.** *The sectional curvature of the Gamma manifold at each point* $(\gamma, \kappa) \in \left(\mathbb{R}_+^*\right)^2$ *verifies*

$$-\frac{1}{2} < K(\gamma, \kappa) = K(\kappa) = \frac{\psi'(\kappa) + \kappa\psi''(\kappa)}{4(-1 + \kappa\psi'(\kappa))^2} < -\frac{1}{4}.$$

In particular, any two gamma distributions can be linked by a unique geodesic in the parameter space, the length of which gives the Fisher-Rao distance.

We implement the information geometry of the Gamma distributions reviewed here within the Python classes `GammaDistributions` and `GammaMetric` shown in Figure 1. Let us mention that the Fisher-Rao geometry of generalized Gamma

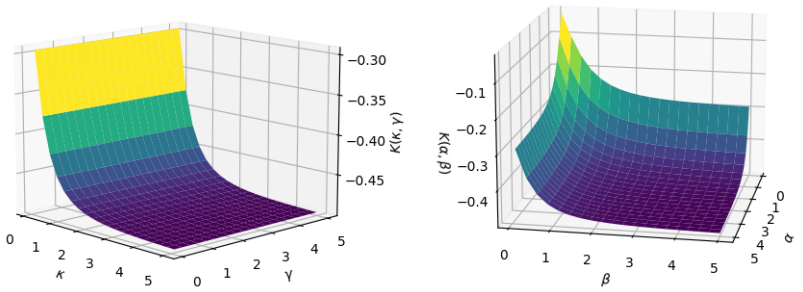

FIGURE 7. Sectional curvature of the Fisher-Rao manifolds of gamma (left) and beta (right) distributions.

distributions have also been studied in the literature [12, 1, 34], and will be the object of future implementation in the proposed information geometry module.

3.4.2. *Applications.* Information geometry of both the standard Gamma and the generalized Gamma manifolds have been used in the literature. Most often, the goal is to implement a "natural" (geodesic) distance between distributions. In that aspect, a geometric reasoning of Gamma distributions finds purposes in many fields, ranging from performance improvement in classification methods in medical imaging [34] to texture retrieval [1].

3.4.3. *`Geomstats` example.* In the following example, we compute the sectional curvature of the Gamma manifold at a given point. The sectional curvature is computed for the subspace spanned by two tangent vectors, but since the gamma manifold is two dimensional, the result does not depend on the chosen vectors.

```
import geomstats.backend as gs
from geomstats.information_geometry import GammaDistributions

dim = 2
manifold = GammaDistributions()
point = gs.array([1., 2.])

vec_a = manifold.to_tangent(gs.random.rand(dim))
vec_b = manifold.to_tangent(gs.random.rand(dim))
vec_c = manifold.to_tangent(gs.random.rand(dim))

print(manifold.metric.curvature(vec_a, vec_b, point))
print(manifold.metric.curvature(vec_a, vec_c, point))

>>> -0.45630369144018423
>>> -0.4563036914401915
```

A comprehensive example using information geometry of the Gamma manifold in the context of traffic optimization in São Paulo can be found in this notebook.

3.5. **Beta and Dirichlet distributions.**

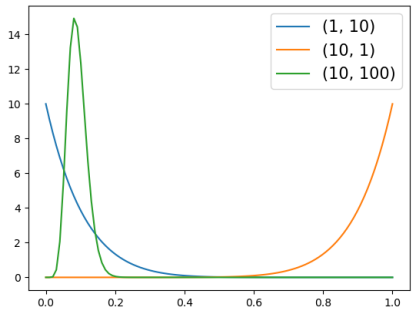

FIGURE 8. P.d.f.s of the beta distributions plotted in Geomstats example 3.5.3. The Fisher-Rao geodesic distance between the parameters of the blue and green distributions is larger than the one between the blue and orange, while the converse is true for the Euclidean distance.

3.5.1. *Main results.* Beta distributions form a 2-parameter family of probability measures defined on the unit interval and often used to define a probability distribution on probabilities. In Bayesian statistics, it is the conjugate prior to the binomial distribution, meaning that if the prior on the probability of success in a binomial experiment belongs to the family of beta distributions, then so does the posterior distribution. This allows users to estimate the distribution of the probability of success by iteratively updating the parameters of the beta prior. Beta and Dirichlet distributions are defined as follows:

Definition 3.5.1.1 (Probability density function of Beta distributions)**.** The p.d.f. of beta distributions is parameterized by two shape parameters $\alpha, \beta > 0$ and given by:

$$f(x|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}, \quad \forall x \in [0, 1].$$

Figure 8 shows examples of p.d.f. of beta distributions, which can take a wide variety of shapes. The distribution has a unique mode in $]0, 1[$ when $\alpha, \beta > 1$, and a mode in 0 or 1 otherwise.

Beta distributions can be seen as a sub-family of the Dirichlet distributions, defined on the $(n-1)$-dimensional probability simplex Δ_{n-1} of n-tuples composed of non-negative components that sum up to one. Similarly to the beta distribution, the Dirichlet distribution is used in Bayesian statistics as the conjugate prior to themultinomial distribution. It is a multivariate generalization of the beta distribution in the sense that if X is a random variable following a beta distribution of parameters α_1, α_2, then $(X, 1-X)$ follows a Dirichlet distribution of same parameters on Δ_1.

Definition 3.5.1.2 (Probability density function of Dirichlet distributions)**.** The p.d.f. of Dirichlet distributions is parametrized by n positive reals $\alpha_1, \ldots, \alpha_n > 0$

and given by:

$$f(x|\alpha_1,\dots,\alpha_n) = \frac{\Gamma(\sum_{i=1}^n \alpha_i)}{\prod_{i=1}^n \Gamma(\alpha_i)} \prod_{i=1}^n {x_i}^{\alpha_i - 1}, \quad \forall (x_1,\dots,x_n) \in \Delta_{n-1}.$$

Proposition 3.5.1.1 (The Fisher-Rao metric on the Dirichlet manifold [24])**.** *The Fisher-Rao metric on the parameter manifold* $\Theta = (\mathbb{R}_+^*)^n$ *of Dirichlet distributions is*

$$ds^2 = \sum_{i=1}^n \psi'(\alpha_i) d\alpha_i^2 - \psi'(\bar{\alpha}) d\bar{\alpha}^2,$$

where $\bar{\alpha} = \sum_{i=1}^n \alpha_i$.

No closed form are known for the geodesics of the beta and Dirichlet manifold. Therefore, our `information_geometry` module solves the geodesic equations numerically. Figure 9 shows 3 geodesics, 2 geodesic sphere and 1 geodesic grid for the beta manifold, and Figure-10 shows geodesic spheres in the 3-Dirichlet manifold. In the beta manifold, the oval shape of the geodesic spheres suggest that the cost to go from one point to another is less important along the lines of equation $\alpha_2/\alpha_1 = \text{cst}$. This seems natural since these are the lines of constant distribution mean.

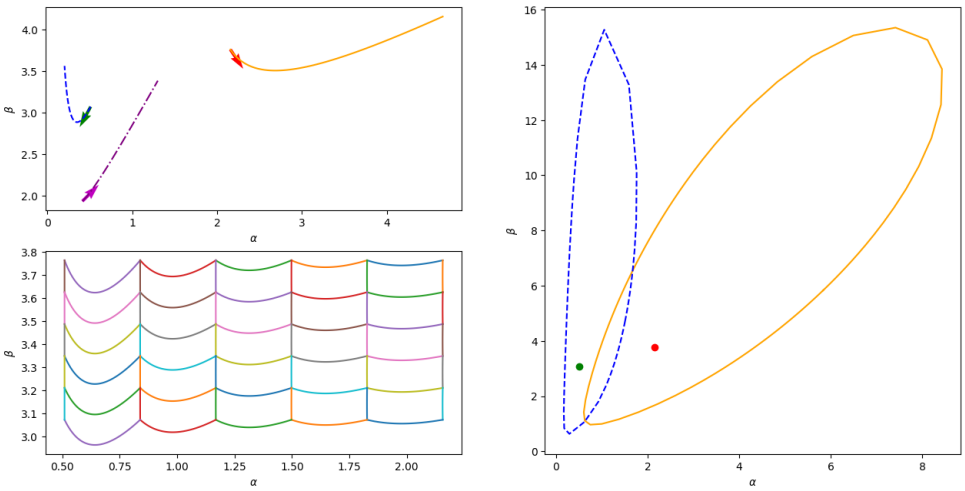

FIGURE 9. Information geometry of the Beta manifold implemented in `BetaDistributions`. Up-left: three geodesics of length 1 departing from three random points A (red) and B (green) and C (magenta, with $\frac{\alpha}{\beta}$ constant). Bottom-left: geodesic grid between A and B. Right: two geodesic spheres of unit radius centered on A and B;

The Dirichlet manifold is isometric to a hypersurface in flat $(n+1)$-dimensional Minkowski space through the transformation

$$(x_1,\dots,x_n) \mapsto (\eta(x_1),\dots,\eta(x_n),\eta(x_1+\dots+x_n)),$$

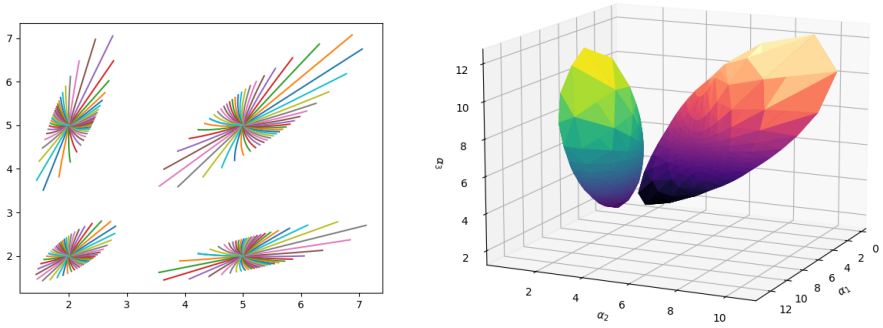

FIGURE 10. Left: rays of four geodesic spheres in the beta manifold, the oval shape of which suggest that the cost to go from one beta distribution to another is less important along the lines of equation $\alpha_2/\alpha_1 = \text{cst}$. This seems natural since these are the lines of constant distribution mean. Right: geodesic spheres of unit radius in the 3-Dirichlet manifold.

where $\eta'(x) = \sqrt{\psi'(x)}$. This allows to show the following result on the curvature, which is plotted in Figure 7 for dimension 2.

Proposition 3.5.1.2 ([24]). *The parameter manifold of Dirichlet distributions endowed with the Fisher-Rao metric is simply connected, geodesically complete and has everywhere negative sectional curvature. In particular, any two Dirichlet distributions can be linked by a unique geodesic, the length of which gives the Fisher-Rao distance.*

The classes `BetaDistributions`, `DirichletDistributions`, and `DirichletMetric` implement the geometries described here. We note that `BetaDistributions` inherits from `DirichletDistributions` and thus inherits the computations coming from its Fisher-Rao metrics as shown in Figure 1.

3.5.2. *Applications.* The Fisher-Rao geometry of beta distributions has received less attention in the literature than the previously described families, although it has been used in [23] to classify histograms of medical data.

3.5.3. *`Geomstats` example.* The following example compares the Fisher-Rao distance with the Euclidean distance between the beta distributions shown in Figure 8. The Euclidean distance between the beta distributions with p.d.f.s shown in blue and green is much larger than the one between the blue and orange. This does not seem satisfactory when considering the differences in mean and mass overlap. By contrast, the blue distribution is closer to the green than to the orange distribution according to the Fisher-Rao metric.

```
import matplotlib.pyplot as plt

import geomstats.backend as gs
from geomstats.information_geometry.beta import BetaDistributions

point_a = gs.array([1., 10.])
```

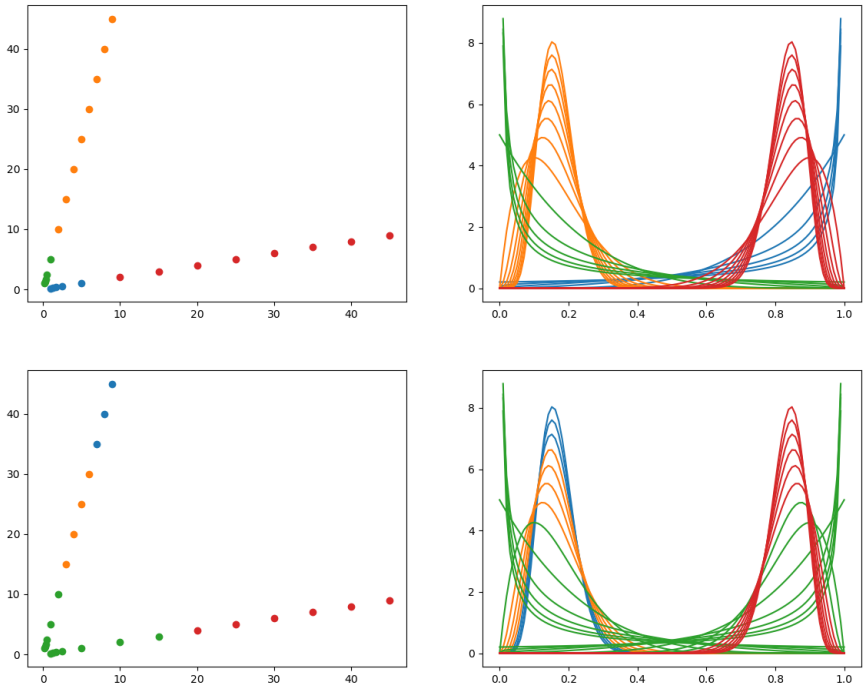

FIGURE 11. Results of K-means clustering of the beta distributions of Geomstats example 3.5.3 using the Fisher-Rao metric (upper row) and the Euclidean distance (lower row), shown in terms of parameters (left column) and p.d.f.s (right column). Contrary to the Euclidean distance, the Fisher-Rao metric regroups the distributions with the same mean, i.e. with parameters aligned on a straight line through the origin, and inside a group of same mean, it regroups the p.d.f.s with similar shape.

```
point_b = gs.array([10., 1.])
point_c = gs.array([10., 100.])

# Plot pdfs
samples = gs.linspace(0., 1., 100)
points = gs.stack([point_a, point_b, point_c])
pdfs = manifold.point_to_pdf(points)(samples)
plt.plot(samples, pdfs)
plt.show()

# Euclidean distances
print(gs.linalg.norm(point_a - point_b))
print(gs.linalg.norm(point_a - point_c))

>>> 12.73
>>> 90.45
```

```
# Fisher-Rao distances
print(manifold.metric.dist(point_a, point_b))
print(manifold.metric.dist(point_a, point_c))

>>> 4.16
>>> 1.76
```

More generally, beta distributions with the same mean are close for the Fisher-Rao metric. Indeed, the oval shape of the geodesic balls shown in Figure 8 suggests that the cost to go from one point to another is less important along the lines of equation $\alpha_2/\alpha_1 = \text{cst}$, which are the lines of constant distribution mean.

The next example performs K-means clustering , using either the Euclidean distance or the Fisher-Rao distance. We consider a set of beta distributions whose means take only two distinct values, which translates into the alignment of the parameters on two straight lines going through the origin, see Figure 11. The clustering based on the Fisher-Rao metric (top row of the figure) distinguishes these two classes, and can further separate the distributions according to the shape of their p.d.f. The Euclidean distance on the other hand (bottom row of the figure) does not distinguish between the two different means.

```
import geomstats.backend as gs
from geomstats.geometry.euclidean import Euclidean
from geomstats.information_geometry.beta import BetaDistributions
from geomstats.learning.kmeans import RiemannianKMeans

# Data
values = gs.array([1/i for i in range(1, 6)] + [i for i in range(2, 10)])

factor = 5
cluster_1 = gs.stack((values, factor * values)).T
cluster_2 = gs.stack((factor * values, values)).T

points = gs.vstack((cluster_1, cluster_2))

n_points = points.shape[0]
n_clusters = 4

# KMeans with the Euclidean distance
r2 = Euclidean(dim=2)

kmeans = RiemannianKMeans(metric=r2.metric, n_clusters=n_clusters, verbose=1)
centroids_eucl = kmeans.fit(points)
labels_eucl = kmeans.predict(points)

# KMeans with the Fisher Rao distance
beta = BetaDistributions()

kmeans = RiemannianKMeans(metric=beta.metric, n_clusters=n_clusters, verbose=1)
centroids_riem = kmeans.fit(points)
labels_riem = kmeans.predict(points)
```

4. Application to text classification

This section presents a comprehensive usecase of the proposed Geomstats module `information_geometry` for text classification using the information manifold of Dirichlet distributions.

We use the Latent Dirichlet Allocation (LDA) model to represent documents in the parameter manifold of Dirichlet distributions. LDA is a generative model for text, where each document is seen as a random mixture of topics, and each topic as a categorical distribution over words [9]. Specifically, consider a corpus with several documents composed of words from a dictionary of size V, and K topics represented by a $K \times V$ matrix β where the i-th line $\beta_{i\bullet}$ gives the discrete probability distribution of the i-th topic over the vocabulary. Given a Dirichlet parameter α in Δ_{K-1} the $(K-1)$-dimensional simplex, each document of N words is generated as follows. First, we sample mixing coefficients $\theta = (\theta_1, \dots, \theta_K) \sim \text{Dirichlet}(\alpha)$. Next, in order to generate each word, we sample the i-th topic from Categorical(θ). Finally, we sample a word from Categorical$(\beta_{i\bullet})$. In other words, for each document the following two steps are iterated for $n = 1, \dots, N$:

(1) select a topic z_n according to $\mathbb{P}(z_n = i|\theta) = \theta_i$, $1 \leq i \leq K$
(2) select a word w_n according to $\mathbb{P}(w_n = j|z_n, \beta) = \beta_{ij}$, $1 \leq j \leq V$.

Here "$z_n = i$" means that the i^{th} topic is selected among the K possible topics, and this is encoded as a vector of size K full of zeros except for a 1 in i^{th} position. Similarly, "$w_n = j$" means that the j^{th} word of the dictionary is selected and is encoded by a vector of size V full of zeros except for a 1 in j^{th} position.

The Dirichlet parameter $\alpha \in \Delta_{K-1}$ and the word-topic distributions $\beta \in \mathbb{R}^{k \times V}$ are the parameters of the model, which need to be estimated from data. Unfortunately, the likelihood of the LDA model cannot be computed and therefore cannot be maximized directly to estimate these parameters. In the seminal paper [9], the authors introduce variational parameters that are document-specific, as well as a lower bound of the likelihood that involves these parameters. This bound can serve as a substitute for the true likelihood when estimating the parameters α and β. In binary classification experiments, the authors use the variational Dirichlet parameters to represent documents of the Reuters-21578 dataset and perform Euclidean support vector machine (SVM) in this low-dimensional representation space.

Here we also use the parameter space of Dirichlet distributions to represent documents. However, we use the Fisher-Rao metric instead of the Euclidean metric for comparison. We extract 140 documents from the 20Newsgroups dataset, a collection of news articles labeled according to their main topic. We select documents from 4 different classes: 'alt.atheism', 'comp.graphics', 'comp.os.ms-windows.misc', 'soc.religion.christian'. We then perform LDA on the obtained corpus, estimate the corresponding variational Dirichlet parameters on a space of $K = 10$ topics, and use these to represent the documents in the 10-dimensional parameter manifold of Dirichlet distributions. The pairwise distances between these parameters, regrouped by classes, for the Euclidean distance and the Fisher-Rao geodesic distance are shown in Figure 12. While the 4 classes structure does not appear clearly, one can see 2 classes appear —one corresponding to religion and the other to computers —more distinctly with the Fisher-Rao metric than with the Euclidean metric. We use these distance matrices to perform K-nearest neighbors classification ($K = 10$) after splitting the dataset into training and testing sets, and show the evolution of

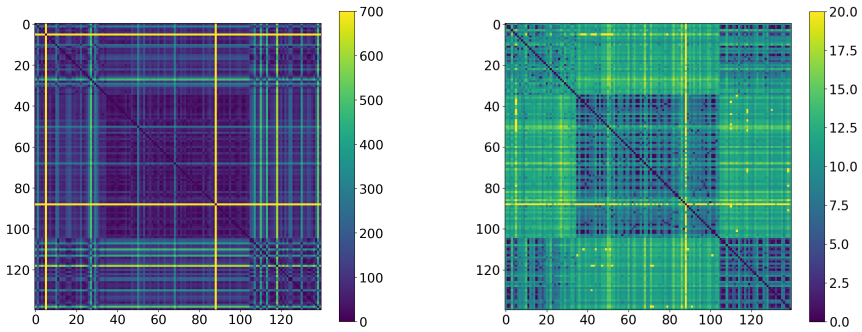

FIGURE 12. Distance matrices between the variational Dirichlet parameters of 140 documents from 4 classes of the 20NewsGroup dataset, for the Euclidean distance (left) and the Fisher-Rao geodesic distance (right). The indices are regrouped by classes, which are 'alt.atheism', 'comp.graphics', 'comp.os.ms-windows.misc', 'soc.religion.christian'.

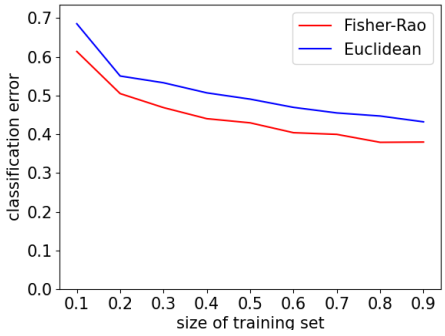

FIGURE 13. Classification error of K-nearest neighbors algorithm applied to 140 documents from 4 classes of the 20Newsgroups dataset, using the Euclidean distance and the Fisher-Rao geodesic distance, plotted with respect to the percentage of the data chosen for the training set.

the classification error with respect to the percentage of data chosen for the training set in Figure 13. We observe that the classification error is consistently lower for the Fisher-Rao metric compared to the Euclidean metric.

CONCLUSION

In this paper, we presented a Python implementation of information geometry integrated in the software Geomstats. We showed that our module `information_geometry` contains the essential building blocks to perform statistics and machine learning on probability distributions data. As we have described the formulas and mathematical

structures implemented in our module, we have also reviewed the main analytical results of the field and the main areas of applications. We also demonstrated a clear usecase of information geometry for text classification, where the geometry of the probability space helps improve the data analysis. We hope that our implementation will inspire researchers to use, and contribute to, information geometry with the Geomstats library.

Appendix

4.1. **Proof of geodesic distance for geometric distributions.** A geometric distribution of parameter $p \in [0,1]$ has a p.m.f. :

$$\forall k \geq 1,\ P(k|p) = f(k|p) = (1-p)^{k-1}p.$$

Then, for $0 < p < 1$, as $\frac{\partial^2 \log f}{\partial p^2} = \frac{1-k}{(1-p)^2} - \frac{1}{p^2}$, we have:

$$I(p) = -\mathbb{E}_p\left[\frac{\partial^2 \log f(X)}{\partial p^2}\right] = \frac{1}{p^2} + \frac{\mathbb{E}(X)-1}{(1-p)^2} = \frac{1}{p^2} + \frac{1}{p(1-p)} = \frac{1}{p^2(1-p)}$$

Then, with ds the infinitesimal distance on the geometric manifold, we get:

$$ds^2 = \frac{1}{p^2(1-p)}dp^2.$$

Therefore the distance between p_1 and $p_2 \geq p_1$ writes:

$$d(p_1,p_2) = \int_{p_1}^{p_2} \frac{1}{p}\frac{1}{\sqrt{1-p}}dp.$$

With the change of variable $u = \sqrt{p}$, we eventually draw:

$$d(p_1,p_2) = 2\int_{\sqrt{p_1}}^{\sqrt{p_2}} \frac{du}{u\sqrt{1-u^2}} = 2\left[\tanh^{-1}\left(\sqrt{1-u^2}\right)\right]_{\sqrt{p_1}}^{\sqrt{p_2}}.$$

Finally:

$$d(p_1,p_2) = 2\left(\tanh^{-1}\left(\sqrt{1-p_2}\right) - \tanh^{-1}\left(\sqrt{1-p_1}\right)\right).$$

4.2. **Proof of geodesic distance for on the Gamma manifold with fixed κ.** From the Fisher information matrix obtained in 3.4.1.2., we derive here:

$$ds^2 = \frac{\kappa}{\gamma^2}d\gamma^2,$$

and then for $\gamma_1 \leq \gamma_2$:

$$d(\gamma_1,\gamma_2) = \sqrt{\kappa}\int_{\gamma_1}^{\gamma_2} \frac{d\gamma}{\gamma} = \sqrt{\kappa}\log\left(\frac{\gamma_2}{\gamma_1}\right).$$

References

[1] Zakariae Abbad, El Maliani, Ahmed Drissi, Said Ouatik Alaoui, and Mohammed El Hassouni. Rao-geodesic distance on the generalized gamma manifold: Study of three sub-manifolds and application in the texture retrieval domain. *Note di Matematica*, 37(supp1):1–18, 2017.

[2] Shun-ichi Amari. *Information geometry and its applications*, volume 194. Springer, 2016.

[3] Luigi Ambrosio and Nicola Gigli. A user's guide to optimal transport. In *Modelling and optimisation of flows on networks*, pages 1–155. Springer, 2013.

[4] Jesus Angulo and Santiago Velasco-Forero. Morphological processing of univariate gaussian distribution-valued images based on poincaré upper-half plane representation. In *Geometric Theory of Information*, pages 331–366. Springer, 2014.

[5] Vincent Arsigny. *Processing Data in Lie Groups: An Algebraic Approach. Application to Non-Linear Registration and Diffusion Tensor MRI*. PhD thesis, École polytechnique, 11 2006.
[6] Rafael Arutjunjan. Informationgeometry.jl, 2021.
[7] Khadiga Arwini and Christopher TJ Dodson. *Information geometry: near randomness and near independence*. Springer Science & Business Media, 2008.
[8] Colin Atkinson and Ann FS Mitchell. Rao's distance measure. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 345–365, 1981.
[9] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
[10] J Burbea, JM Oller, and F Reverter. Some remarks on the information geometry of the gamma distribution. 2002.
[11] Nikolai Nikolaevich Cencov. Statistical decision rules and optimal inference. transl. math. *Monographs, American Mathematical Society, Providence, RI*, 1982.
[12] William WS Chen and Samuel Kotz. The riemannian structure of the three-parameter gamma distribution. 2013.
[13] Sueli IR Costa, Sandra A Santos, and Joao E Strapasson. Fisher information distance: A geometrical reading. *Discrete Applied Mathematics*, 197:59–69, 2015.
[14] Manfredo Perdigao Do Carmo and J Flaherty Francis. *Riemannian geometry*, volume 6. Springer, 1992.
[15] Ian Dryden and Kanti Mardia. *Statistical shape analysis, with Applications in R*. John Wiley & Sons, New York, 1998.
[16] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.
[17] Thomas Friedrich. Die fisher-information und symplektische strukturen. *Mathematische Nachrichten*, 153(1):273–296, 1991.
[18] Nicolas Guigui, Nina Miolane, and Xavier Pennec. Introduction to riemannian geometry and geometric statistics: from basic theory to implementation with geomstats. (in press)., 2023.
[19] Marc Harper. Information geometry and evolutionary game theory. *arXiv preprint arXiv:0911.1383*, 2009.
[20] Gregory J Husak, Joel Michaelsen, and Chris Funk. Use of the gamma distribution to represent monthly rainfall in africa for drought monitoring applications. *International Journal of Climatology: A Journal of the Royal Meteorological Society*, 27(7):935–944, 2007.
[21] Robert E Kass. The geometry of asymptotic inference. *Statistical Science*, pages 188–219, 1989.
[22] Stefan L Lauritzen. Statistical manifolds. *Differential geometry in statistical inference*, 10:163–216, 1987.
[23] Alice Le Brigant, Nicolas Guigui, Sana Rebbah, and Stéphane Puechmorel. Classifying histograms of medical data using information geometry of beta distributions. *IFAC-PapersOnLine*, 54(9):514–520, 2021.
[24] Alice Le Brigant, Stephen C Preston, and Stéphane Puechmorel. Fisher-rao geometry of dirichlet distributions. *Differential Geometry and its Applications*, 74:101702, 2021.
[25] Guy Lebanon. Learning riemannian metrics. *arXiv preprint arXiv:1212.2474*, 2012.
[26] Erich L Lehmann and George Casella. *Theory of point estimation*. Springer Science & Business Media, 2006.
[27] Christophe Lenglet, Mikaël Rousson, Rachid Deriche, and Olivier Faugeras. Statistics on the manifold of multivariate normal distributions: Theory and application to diffusion tensor mri processing. *Journal of Mathematical Imaging and Vision*, 25(3):423–444, 2006.
[28] Stephen J Maybank. Detection of image structures using the fisher information and the rao metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(12):1579–1589, 2004.
[29] Nina Miolane, Nicolas Guigui, Alice Le Brigant, Johan Mathe, Benjamin Hou, Yann Thanwerdas, Stefan Heyder, Olivier Peltre, Niklas Koep, Hadi Zaatiti, et al. Geomstats: a python

package for riemannian geometry in machine learning. *Journal of Machine Learning Research*, 21(223):1–9, 2020.

[30] Felix Otto. The geometry of dissipative evolution equations: the porous medium equation. 2001.

[31] Xavier Pennec, Pierre Fillard, and Nicholas Ayache. A riemannian framework for tensor computing. *International Journal of computer vision*, 66(1):41–66, 2006.

[32] Julianna Pinele, Sueli IR Costa, and João E Strapasson. On the fisher-rao information metric in the space of normal distributions. In *International Conference on Geometric Science of Information*, pages 676–684. Springer, 2019.

[33] C Radhakrishna Rao. Information and the accuracy attainable in the estimation of statistical parameters. *Bull. Calcutta Math. Soc.*, 37, 01 1945.

[34] Sana Rebbah, Florence Nicol, and Stéphane Puechmorel. The geometry of the generalized gamma manifold and an application to medical imaging. *Mathematics*, 7(8):674, 2019.

[35] Yoshiharu Sato, Kazuaki Sugawa, and Michiaki Kawaguchi. The geometrical structure of the parameter space of the two-dimensional normal distribution. *Reports on Mathematical Physics*, 16(1):111–119, 1979.

[36] Vadim Semenikhine, Edward Furman, and Jianxi Su. On a multiplicative multivariate gamma distribution with applications in insurance. *Risks*, 6(3):79, 2018.

[37] Hiroshi Shinmoto, Koichi Oshio, Chiharu Tamura, Shigeyoshi Soga, Teppei Okamura, Kentaro Yamada, Tastumi Kaji, and Robert V Mulkern. Diffusion-weighted imaging of prostate cancer using a statistical model based on the gamma distribution. *Journal of Magnetic Resonance Imaging*, 42(1):56–62, 2015.

[38] Lene Theil Skovgaard. A Riemannian geometry of the multivariate normal model. *Scandinavian Journal of Statistics*, pages 211–223, 1984.

[39] João E Strapasson, Julianna Pinele, and Sueli IR Costa. Clustering using the fisher-rao distance. In *2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pages 1–5. IEEE, 2016.

[40] Geert Verdoolaege and Paul Scheunders. Geodesics on the manifold of multivariate generalized gaussian distributions with an application to multicomponent texture discrimination. *International Journal of Computer Vision*, 95(3):265–286, 2011.

[41] Laurent Younes. Spaces and manifolds of shapes in computer vision: An overview. *Image and Vision Computing*, 30(6-7):389–397, 2012.