A New Framework for Classifying Probability Density Functions

Anis Fradi(✉) and Chafik Samir

University of Clermont Auvergne, LIMOS CNRS(UMR 6158), 63000 Clermont-Ferrand, France
{anis.fradi,chafik.samir}@uca.fr

Abstract. This paper introduces a new framework for classifying probability density functions. The proposed method fits in the class of constrained Gaussian processes indexed by distribution functions. Firstly, instead of classifying observations directly, we consider their isometric transformations which enables us to satisfy both positiveness and unit integral hard constraints. Secondly, we introduce the theoretical proprieties and give numerical details of how to decompose each transformed observation in an appropriate orthonormal basis. As a result, we show that the coefficients are belonging to the unit sphere when equipped with the standard Euclidean metric as a natural metric. Lastly, the proposed methods are illustrated and successfully evaluated in different configurations and with various dataset.

Keywords: Classification · Constrained Gaussian Processes · Distribution Functions · Bayesian Inference

1 Introduction

Supervised learning is a powerful tool for solving many real-world problems in various fields [2]. It has a wide range of applications, including but not limited to, image recognition, natural language processing, sentiment analysis, fraud detection, and prediction in finance and health-care. For example, in image recognition [20], supervised learning algorithms can be trained on large datasets of labeled images to identify objects and classify them into specific categories. In language processing [16], supervised learning can be used for text classification, sentiment analysis, and language translation. In finance [26], supervised learning can be used to predict stock prices. Some popular supervised learning algorithms include linear regression [14], logistic regression [13], decision trees [6], random forests [9] and support vector machines [30]. These algorithms have different strengths and weaknesses and are suitable for different types of problems. The choice of an algorithm depends on the nature of the problem, the amount of labeled data available and the desired level of accuracy.

Nowadays, Gaussian processes are powerful methods for modeling complex data that does not have a simple linear relationship between the input and the

D. Koutra et al. (Eds.): ECML PKDD 2023, LNAI 14169, pp. 507–522, 2023.
https://doi.org/10.1007/978-3-031-43412-9_30

output variables [28,32]. They are particularly useful when data have a high degree of noise or/and uncertainty. A Gaussian process (GP) can also be used for Bayesian optimization and for active learning. In probability and statistics a standard GP is a stochastic process (a collection of random variables indexed by time or space), such that every finite collection has a multivariate normal distribution, i.e., every finite linear combination of them is normally distributed [10]. GP regression models have been extensively developed for statistical machine learning. One of the main advantages of GP regression is that it provides a measure of uncertainty in the predictions. A Gaussian process classifier (GPc) is a machine learning method that adapts GPs for the classification task where the goal is to learn a mapping from input features to a categorical output. The first step of a GPc is to specify a covariance function that defines the covariance between the input features. The covariance function essentially captures the similarity between pairs of data points. Once the covariance function is specified the GPc can be trained on a labeled dataset using a technique called maximum likelihood estimation. This involves finding the values of the covariance hyperparameters that maximize the marginal likelihood of the observed data.

However, standard GPs were limited to data in vector spaces. In fields such as shape analysis [19,31] and diffusion tensor imaging [1] data often lie on a manifold. Therefore, the standard GP model is not straightforwardly applicable to a non-Euclidean space due to hard constraints/limitations imposed by the underlying function [24]. This usually makes the GP model nonviable since the resulting predictive distribution does not live in the correct geometric space. In this context, the linear regression was first generalized to solve the problem of manifold-valued data based on the geodesic regression before being extended for multidimensional covariates [18]. Furthermore, [22] generalized GPs to Riemannian manifolds as wrapped Gaussian processes. Recently, [4] constructed covariance functions in order to obtain GPs indexed by probability measures endowed with the Wasserstein metric. More recently, [29] provided a unified framework of GPs indexed by non-decreasing distribution functions (SNDF) endowed with the Fisher-Rao metric. The closest to our work is that of [11] for which authors have represented functional data by their corresponding probability density functions (PDFs). They also benefited from the connection between the set of PDFs endowed with the Fisher-Rao metric and the set of square-root density functions (SRDFs) endowed with the $\mathbb{L}^2$ metric resulting to be the Hilbert upper-hemisphere with many advantageous geometric tools [12].

In general, functional data analysis (FDA) is about the analysis of information on univariate functions, multidimensional curves, surfaces, etc [27]. Some of commonly used techniques in FDA include functional principal component analysis [33]. A relevant reference on this topic includes the classification of functional data with a segmentation approach [7] and FDA via neural networks [21]. In particular, a PDF is a type of functional data that describes the probability distribution of a continuous random variable. In other words, the PDF of a continuous random variable is a function that maps each realization of the random variable to the relative probability of that value occurring. The set of

PDFs is a constrained functional space that has been applied in many real-world applications [5]. Indeed, PDFs are most commonly preferred as a representation of functional data thanks to their ability to improve the local distributions and explore the skewness of original data [15]. Contrariwise, such representations even their ability to describe functional data prevent the linearity of transformed data due to both positiveness and unit integral constraints [3]. To overcome such issue one should define a metric on the set of PDFs which matches the mentioned constraints. In particular, the consistency of regression and classification with PDFs as inputs was established in [23,25].

One of the main disadvantages of GPs indexed by Riemannian manifolds is that they can be computationally expensive especially for large datasets. In fact, the distance should be evaluated in functional spaces. However, several approximate methods can be used to make this class of GPs more computationally efficient. Keeping the same idea, in this paper, we will develops GPs indexed by PDFs as a measure of divergence between them based on the well-defined covariance function. In contrast to [11] we consider the formal expansion of a SRDF in terms of a $\mathbb{L}^2$ basis yielding from the convergent orthogonal series expansion [8]. We then exploit the fact that the set of SRDFs endowed with the $\mathbb{L}^2$ functional metric resulting to be the Hilbert hemi-sphere is isometric to the Euclidean sphere endowed with the l^2 square-summable metric generated by the set of coefficients resulting from the expansion at hand. Given a finite set of $\mathbb{L}^2$ basis assumed to maintain most information of the SRDF the restriction to the (uncountably) infinite-dimensional Hilbert sphere translates to a restriction to the (countably) finite-dimensional sphere endowed with the l^2 Euclidean metric.

The rest of the paper is organized as follows. In Sect. 2, we review the GPc model, inference, learning, and prediction. Section 3 presents how to move from a PDF to a vector of coefficients belonging to the tangent space of the Euclidean sphere when dealing with the convergent orthogonal series expansion. Section 4 introduces the GPc indexed by the set of PDFs thanks to the isometry with the tangent space of the Euclidean sphere. Empirical results are presented and discussed in Sect. 5. Finally, Sect. 6 concludes the paper.

2 Standard Gaussian Process Classifier

We are given N observations $(x_1, y_1), \ldots, (x_N, y_N)$ with $x_i \in \mathbb{R}^d$ are the d-dimensional inputs (predictors) and y_i are the associated responses ($i = 1, \ldots, N$). In this paper, we consider the binary classification where y_i takes values in $\{-1, +1\}$ for which a GP becomes a GPc. A GPc is a probabilistic model that makes predictions by learning a mapping from inputs to class probabilities. In particular, we are interested in finding the probability of the target class "+1" satisfying: $\pi(x) = \mathbb{P}(y = +1|f(x)) = \sigma(f(x))$, depending on an activation function $\sigma : \mathbb{R} \to [0, 1]$ and usually referring to the sigmoid $\sigma(t) = 1/(1+\exp(-t))$. In a Bayesian framework, we model f with a zero mean GPc of a covariance function $c(.,.)$ controlling its underlying structure, i.e., $f(x) \sim \mathcal{GP}(0, c(x, x'))$. Note that, in this context, our formulation is different from kernel-based methods [17]

and all predictions are guaranteed to be PDFs. Since y_i is of binary values then $y_i|f(x_i)$ follows a Bernoulli law $\sim \mathcal{B}(\sigma(f(x_i)))$. The standard GPc model is

$$\begin{cases} f \sim \mathcal{GP}(0, c) \\ y_i|f(x_i) \sim \mathcal{B}(\sigma(f(x_i))) \end{cases}$$

In this paper, the covariance function $c(.,.)$ is supposed to be homogeneous which means that it is associated with a stationary parametrized kernel $K_\theta : \mathbb{R} \to \mathbb{R}$ such that $c(x, x') = K_\theta(||x - x'||_2)$.

Likelihood. Let $\mathbf{x} = (x_1, ..., x_N)^T$, $\mathbf{y} = (y_1, ..., y_N)^T$ and $\mathbf{f} = (f_1, ..., f_N)^T = (f(x_1), ..., f(x_N))^T$. The likelihood term is the product of individual likelihoods

$$\mathbb{P}(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{N} \mathbb{P}(y_i|f_i) = \prod_{i=1}^{N} \sigma(y_i f_i) \tag{1}$$

Prior. Since $f \sim \mathcal{GP}(0, c)$ then $\mathbf{f}|\mathbf{x}$ follows a multivariate Gaussian law

$$\mathbb{P}(\mathbf{f}|\mathbf{x}) = \mathcal{N}(\mathbf{f}|0, \mathbf{C}); \quad \mathbf{C} = c(\mathbf{x}, \mathbf{x}) \tag{2}$$

Posterior. From the Bayes' rule we write the posterior distribution as

$$\mathbb{P}(\mathbf{f}|\mathbf{x}, \mathbf{y}) = \frac{\mathbb{P}(\mathbf{f}|\mathbf{x}) \times \mathbb{P}(\mathbf{y}|\mathbf{f})}{\mathbb{P}(\mathbf{y}|\mathbf{x})} \propto \mathbb{P}(\mathbf{f}|\mathbf{x}) \times \mathbb{P}(\mathbf{y}|\mathbf{f}) \tag{3}$$

where $\mathbb{P}(\mathbf{y}|\mathbf{x})$ refers to the marginal likelihood. The posterior is analytically intractable and need to be approximated due to the likelihood term. To handle this issue one can introduce the Laplace approximation by finding the maximum a posteriori (MAP) estimator of $\mathbf{f}$ denoted $\hat{\mathbf{f}} = (\hat{f}_1, ..., \hat{f}_N)^T$ from the Newton-Raphson method, iteratively

$$\mathbf{f}^{k+1} = \left(\mathbf{C}^{-1} + \mathbf{W}^k\right)^{-1} \left(\mathbf{W}^k \mathbf{f}^k + \nabla \log \mathbb{P}(\mathbf{y}|\mathbf{f})|_{\mathbf{f}=\mathbf{f}^k}\right); \quad k = 1, 2, \ldots \tag{4}$$

$\mathbf{W}^k$ is the negative Hessian matrix of the likelihood at $\mathbf{f}^k$: $\mathbf{W}^k = -\nabla^2 \log \mathbb{P}(\mathbf{y}|\mathbf{f})\big|_{\mathbf{f}=\mathbf{f}^k}$. Once we estimate $\hat{\mathbf{f}}$ and $\hat{\mathbf{W}} = -\nabla^2 \log \mathbb{P}(\mathbf{y}|\mathbf{f})\big|_{\mathbf{f}=\hat{\mathbf{f}}}$ yields a posterior approximation from a second order Taylor expansion of $\log \mathbb{P}(\mathbf{f}|\mathbf{x}, \mathbf{y})$ around $\hat{\mathbf{f}}$ as

$$\hat{\mathbb{P}}(\mathbf{f}|\mathbf{y}, \mathbf{x}) = \mathcal{N}(\mathbf{f}|\hat{\mathbf{f}}, (\mathbf{C}^{-1} + \hat{\mathbf{W}})^{-1}) \tag{5}$$

Given a test input x^* the predictive distribution at $f^* = f(x^*)$ is then

$$\hat{\mathbb{P}}(f^*|\mathbf{x}, \mathbf{y}, x^*) = \mathcal{N}(f^*|\mu(x^*), \sigma^2(x^*)) \tag{6}$$

with

$$\begin{cases} \mu(x^*) = \mathbf{C}_*^T \mathbf{C}^{-1} \hat{\mathbf{f}} \\ \sigma^2(x^*) = \mathbf{C}_{**} - \mathbf{C}_*^T (\mathbf{C} + \hat{\mathbf{W}}^{-1})^{-1} \mathbf{C}_* \end{cases} \tag{7}$$

where $\mathbf{C}_* = c(\mathbf{x}, x^*)$ and $\mathbf{C}_{**} = c(x^*, x^*)$. Using the moments of prediction the predictor of $y^* = +1$ satisfies

$$\bar{\pi}(x^*) = \mathbb{P}(y^* = 1|x^*) \approx \int_{\mathbb{R}} \sigma(f^*)\hat{\mathbb{P}}(f^*|\mathbf{x}, \mathbf{y}, x^*)df^* \tag{8}$$

For some applications, the hyperparameter θ associated to the kernel K_θ is known a priori and is chosen according to, for example, certain physical properties. However, in many applied environments the kernel's hyperparameter is learned from data for instance when maximizing the approximate log marginal likelihood satisfying

$$\log \hat{\mathbb{P}}(\mathbf{y}|\mathbf{x}) = -\frac{1}{2}\hat{\mathbf{f}}^T\mathbf{C}^{-1}\hat{\mathbf{f}} + \log \mathbb{P}(\mathbf{y}|\hat{\mathbf{f}}) - \frac{1}{2}\log\left|\mathcal{I}_N + \hat{\mathbf{W}}^{\frac{1}{2}}\mathbf{C}\hat{\mathbf{W}}^{\frac{1}{2}}\right| \tag{9}$$

where $\mathcal{I}_N$ refers to the $N \times N$ diagonal matrix. At this stage, it becomes possible to fit the kernel hyperparameters, for instance, by a gradient-descent algorithm. Inferring the predictive distribution or learning the hyperparameters from the log approximate marginal likelihood is dominated by the inversion of the $N \times N$ covariance matrix $\mathbf{C}$, which incurs a computational cost of $O(N^3)$. Additionally, the memory requirements for GPc scale with a computational complexity of $O(N^2)$.

3 Manifold Structure

Let p be a PDF of a real-valued random variable with respect to the Lebesgue measure. The set of all PDFs defined on $I = [0, 1]$ is a simplex satisfying

$$\mathcal{P} = \left\{p : I \to \mathbb{R} \;\middle|\; p \text{ is nonnegative and } \int_I p(t)dt = 1\right\} \tag{10}$$

$\mathcal{P}$ is a Riemannian manifold when endowed with the Fisher-Rao metric

$$\left\langle g_1, g_2 \right\rangle_p = \int_I \frac{g_1(t)g_2(t)}{p(t)}dt \tag{11}$$

where $g_1, g_2 \in \mathcal{T}_p(\mathcal{P})$ are two tangent vectors at p belonging to

$$\mathcal{T}_p(\mathcal{P}) = \left\{g : I \to \mathbb{R} \;\middle|\; \int_I g(t)dt = 0\right\} \tag{12}$$

As a second representation we introduce the set of SRDFs satisfying

$$\mathcal{H} = \left\{\psi : I \to \mathbb{R} \;\middle|\; \psi \text{ is nonnegative, and } ||\psi||_{\mathbb{L}^2} = \left(\int_I \psi(t)^2 dt\right)^{1/2} = 1\right\} \tag{13}$$

Endowed with the $\mathbb{L}^2$ metric $\mathcal{H}$ results to be the Hilbert upper-hemisphere (nonnegative part). In addition, the tangent space of $\mathcal{H}$ locally at ψ is

$$\mathcal{T}_\psi(\mathcal{H}) = \left\{f : I \to \mathbb{R} \;\middle|\; \left\langle \psi, f \right\rangle_{\mathbb{L}^2} = \int_I \psi(t)f(t)dt = 0\right\} \tag{14}$$

Associated with any $p \in \mathcal{P}$ is a unique $\psi \in \mathcal{H}$ (isometrically) expressed as

$$\psi(t) = \sqrt{p(t)}; \quad t \in I \tag{15}$$

The advantage of representing a PDF $p \in \mathcal{P}$ with $\psi \equiv \sqrt{p} \in \mathcal{H}$ is that it greatly simplifies the underlying geometry of $\mathcal{P}$ with some nice tools on the Hilbert sphere. Since ψ is an element of $\mathbb{L}^2(I, \mathbb{R})$, it can be represented as a convergent orthogonal series expansion

$$\psi(t) = \sum_{l=1}^{\infty} a_l \phi_l(t) \tag{16}$$

where $(\phi_l)_l$ is a complete orthonormal basis in $\mathbb{L}^2(I, \mathbb{R})$. Note that $\psi(t)$ can be re-written as

$$\psi(t) = \Phi(t)^T A \tag{17}$$

for $A = (a_1, a_2, \dots)^T$ and $\Phi(t) = (\phi_1(t), \phi_2(t), \dots)^T$. Consequently, $\psi(t)$ is a SRDF if and only if, in addition to the non-negativity constraint, $A \in \mathcal{S}^{\infty}$ from the following equality

$$||\psi||^2_{\mathbb{L}^2} = \int_I \psi(t)^2 dt = \sum_{l=1}^{\infty} a_l^2 \int_I \phi_l(t)^2 dt = \sum_{l=1}^{\infty} a_l^2 = ||A||_2^2 \tag{18}$$

Here, $\mathcal{S}^{\infty}$ refers to the unit infinite-dimensional Euclidean (square-summable) sphere satisfying

$$\mathcal{S}^{\infty} = \Big\{ A \in l^2 \;\Big|\; ||A||_2 = \Big(\sum_{l=1}^{\infty} a_l^2 \Big)^{1/2} = 1 \Big\} \tag{19}$$

with the corresponding tangent space locally at A as

$$\mathcal{T}_A(\mathcal{S}^{\infty}) = \Big\{ B \in l^2 \;\Big|\; \langle A, B \rangle_2 = \sum_{l=1}^{\infty} a_l b_l = 0 \Big\} \tag{20}$$

Exponential Map. Let A be an element of $\mathcal{S}^{\infty}$ and $B \in \mathcal{T}_A(\mathcal{S}^{\infty})$. We define the exponential map as

$$\exp_A(B) = \cos\big(||B||_2\big) A + \sin\big(||B||_2\big) \frac{B}{||B||_2} \tag{21}$$

The exponential map is a diffeomorphism between the tangent space and the unit finite-dimensional sphere if we restrict B so that $||B||_2 \in [0, \pi[$.

Log Map. For $A_1, A_2 \in \mathcal{S}^\infty$ such that A_1 does not belong to the cut locus of A_2. We define $B \in \mathcal{T}_{A_2}(\mathcal{S}^\infty)$ to be the inverse exponential (log) map of A_1 if $\exp_{A_2}(B) = A_1$. We then use the notation

$$B = \log_{A_2}(A_1) \tag{22}$$

where $B = \frac{\alpha}{||\alpha||_2} d_{\mathcal{S}^\infty}(A_1, A_2)$ and $\alpha = A_2 - \langle A_1, A_2 \rangle_2 A_1$. Here, $d_{\mathcal{S}^\infty}(.,.)$ refers to the geodesic distance on the sphere (the angle of the shortest arc), i.e., $d_{\mathcal{S}^\infty}(A_1, A_2) = \arccos(\langle A_1, A_2 \rangle_2)$.

4 Gaussian Process Classifier on PDFs

In this section, we focus on constructing a GPc on $\mathcal{P}$ based on the connection to the tangent space of the finite-dimensional sphere. A GPc Z on $\mathcal{P}$ is a random field indexed by $\mathcal{P}$ so that $(Z(p_1), \ldots, Z(p_N))^T$ is a multivariate Gaussian vector for $p_1, \ldots, p_N \in \mathcal{P}$. A zero mean GPc Z is completely specified by its covariance function $c_{\mathcal{P}} : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ defined as

$$c_{\mathcal{P}}(p_i, p_j) = cov(Z(p_i), Z(p_j)) \tag{23}$$

A covariance function $c_{\mathcal{P}}(.,.)$ on $\mathcal{P}$ should satisfy the following condition: for any $N \geq 1$ and $\mathbf{p} = (p_1, \ldots, p_N)^T$ the matrix $\mathbf{C}_{\mathcal{P}} = c_{\mathcal{P}}(\mathbf{p}, \mathbf{p})$ is symmetric nonnegative definite.

Lemma 1. *Given an orthonormal basis for $\mathbb{L}^2$, the set of PDFs equipped with the Fisher-Rao metric $(\mathcal{P}, \langle .,. \rangle_p)$ is isometric to the sphere with its natural Euclidean metric $(\mathcal{S}^\infty, \langle .,. \rangle_2)$.*

Proof. The proof yields by composing two isometric maps in (15) and (18). □

Since $A_1 \mapsto \log_{A_2}(A_1)$ is an isometry between $\mathcal{S}^\infty$ and $\mathcal{T}_{A_2}(\mathcal{S}^\infty)$ for $A_2 \in \mathcal{S}^\infty$ then from Lemma 1 we get an isometry between $\mathcal{P}$ and $\mathcal{T}_{A_2}(\mathcal{S}^\infty)$; $p(.) \equiv (\Phi(.)^T A_1)^2 \mapsto \log_{A_2}(A_1)$ by composition of two isometries. As a special case, let $\mathcal{E} = \mathcal{T}_{\mathbf{1}}(\mathcal{S}^\infty)$ be the tangent space of $\mathcal{S}^\infty$ at the infinite unity pole $\mathbf{1} = (0, \ldots, 0, 1)$. The strategy that we adopt to construct covariance functions is to exploit the isometric map $\log_{\mathbf{1}}$ based on the linear tangent space $\mathcal{E}$. That is, we construct covariance functions with (i, j) component as

$$c_{\mathcal{P}}(p_i, p_j) = K_\theta(\| \log_{\mathbf{1}}(A_i) - \log_{\mathbf{1}}(A_j) \|_2) \tag{24}$$

It seems natural to consider a truncated version of ψ at order d expressed as $\psi^d(t) = \sum_{l=1}^{d} a_l \phi_l(t)$ and consider the rest of the sum as an error approximation: $e^d(t) = \sum_{l=d+1}^{\infty} a_l \phi_l(t)$. The truncation $\psi^d(t)$ is then re-written as $\psi^d(t) = \Phi^d(t)^T A^d$ for $A^d = (a_1 \ldots, a_d)^T \in \mathcal{S}^{d-1}$ and $\Phi^d(t) = (\phi_1(t), \ldots, \phi_d)^T$. The covariance on $\mathcal{P}$ approximately becomes

$$c_{\mathcal{P}}(p_i, p_j) \approx K_\theta(\| \log_{\mathbf{1}^d}(A_i^d) - \log_{\mathbf{1}^d}(A_j^d) \|_2) \tag{25}$$

where $\mathbf{1}^d$ is the d-dimensional unity pole of $\mathcal{S}^{d-1}$.

Proposition 1. *Let $K_\theta : \mathbb{R} \to \mathbb{R}$ be a kernel associated to a homogeneous covariance function $c(x_i, x_j)$ defined on $\mathbb{R}^d \times \mathbb{R}^d$, i.e., $c(x_i, x_j) = K_\theta(\|x_i - x_j\|_2)$ and $c_{\mathcal{P}}(.,.)$ be defined like in (25). Then, $c_{\mathcal{P}}(.,.)$ is approximately a covariance function.*

Proof. Let p_i $(i = 1, \ldots, N)$ be a sample of i.i.d. observations on the PDF p depending on the corresponding finite-dimensional spherical coefficients $A_i^d \in \mathcal{S}^{d-1}$ and $B_i^d = \log_{\mathbf{1}^d}(A_i^d)$. Consider the matrix $\tilde{\mathbf{C}}$ with entries $\tilde{\mathbf{C}}_{ij} \approx \left\langle B_i^d, B_j^d \right\rangle_2$. Then $\tilde{\mathbf{C}}$ is approximately a Gram matrix in $\mathbb{R}^{N \times N}$. Therefore, there exists a $d \times d$ nonnegative diagonal matrix D and a $N \times d$ orthogonal matrix P such that $\tilde{\mathbf{C}} \approx PDP^T$. If $e_1, \ldots, e_N$ denote the canonical basis of $\mathbb{R}^N$ then $e_i^T \tilde{\mathbf{C}} e_j \approx x_i^T x_j$ with $x_i = D^{1/2} P^T e_i \in \mathbb{R}^d$ depending on $p_1, \ldots, p_N$. This implies that $\left\langle B_i^d, B_j^d \right\rangle_2 \approx x_i^T x_j$ and consequently $|| \log_{\mathbf{1}^d}(A_i^d) - \log_{\mathbf{1}^d}(A_j^d)||_2 \approx ||x_i - x_j||_2$. Finally, any matrix with entries $K_\theta(\| \log_{\mathbf{1}^d}(A_i^d) - \log_{\mathbf{1}^d}(A_j^d)\|_2)$ can be approximately seen as a covariance matrix with entries $K_\theta(||x_i - x_j||_2)$ and inherits its properties. □

Let p_i $(i = 1, \ldots, N)$ be a sample of i.i.d. observations on the PDF p depending on the corresponding spherical coefficients $A_i^d \in \mathcal{S}^{d-1}$, respectively.

Corollary 1. *If Z is a GPc indexed by PDFs such that*

$$\begin{cases} Z \sim \mathcal{GP}(0, c_{\mathcal{P}}) \\ y_i | Z(p_i) \sim \mathcal{B}(\sigma(Z(p_i))) \end{cases}$$

then there is an approximated standard GPc f on $\mathcal{E}_d = \mathcal{T}_{\mathbf{1}^d}(\mathcal{S}^{d-1})$ satisfying

$$\begin{cases} f \sim \mathcal{GP}(0, c) \\ y_i | f(B_i^d) \sim \mathcal{B}(\sigma(f(B_i^d))) \\ B_i^d = \log_{\mathbf{1}^d}(A_i^d) \end{cases}$$

5 Experimental Results

In this section, we evaluate the proposed model on various datasets and compare it to other state-of-the-art methods. We consider the squared exponential (SE) kernel satisfying

$$K(\tau) = \sigma^2 \exp(-0.5\tau^2/\gamma^2); \quad \tau = ||x - x'||_2 \tag{26}$$

Functions drawn from a GP with this kernel are infinitely differentiable, and can display long-range trends. GPs with a SE kernel are well-suited for modeling functions that exhibit smoothness and continuity properties, such as classification problems. The covariance structures that can be learned from data are the variance σ^2 and the length-scale γ. The orthonormal basis in $\mathbb{L}^2(I, \mathbb{R})$ is set to $\phi_l(t) = \sqrt{2}\sin(l\pi t)$ and the truncation order is fixed to $d = 30$, see more details in [12]. Note that all the methods tested in this section have been carefully implemented in Python programming language on a standard desktop machine running linux.

5.1 Illustrative and Challenging Datasets

Synthetic PDFs. We consider two datasets of simulated PDFs: beta and inverse gamma distributions. They have been applied to model randomness on intervals of finite length and have been widely used in simulation studies for a variety of disciplines. We performed this experiment by simulating 1000 PDFs slightly different for two classes in each dataset. Each observation p_i represents a PDF when we add a random uniform noise to initial parameters. For beta dataset we take $\mathbb{P}(p_i|y_i = +1) = \mathcal{B}(2 + \epsilon_i; 2)$ for the first class and $\mathbb{P}(p_i|y_i = -1) = B(1.8+\epsilon_i; 2)$ for the second one where $\epsilon_i \sim \mathcal{U}([-0.2; 0.2])$ is a realization of the uniform law. For inverse gamma dataset we take $\mathbb{P}(p_i|y_i = +1) = \mathcal{IG}(3+\epsilon_i)$ for the first class and $\mathbb{P}(p_i|y_i = -1) = \mathcal{IG}(2.8+\epsilon_i)$ for the second one. We show some examples of p_i in Fig. 1 (top) with different colors (blue and red) for the two classes.

Real PDFs. In this part, a real study was conducted with two datasets of PDFs. The first dataset consists of 1500 observations giving the segmented and preprocessed electrocardiogram (ECG) signals for Heartbeat (500 normal and 1000 abnormal) ECG Heartbeat Categorization Dataset. This dataset contains a collection of ECG recordings with a sampling frequency: 125 Hz, where the goal is to classify each heartbeat into normal or abnormal when the human was affected by different arrhythmias and myocardial infarction. Each signal includes information about the symptoms during a short period. The information in this dataset

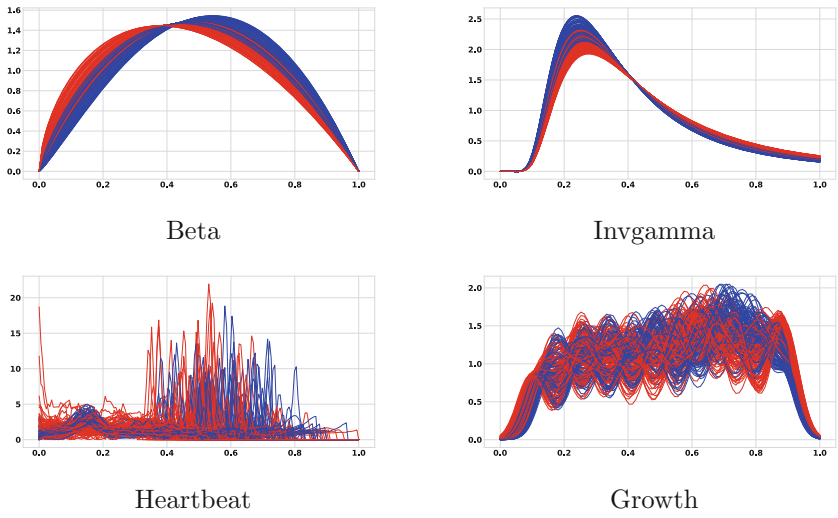

Fig. 1. Some examples of PDFs with first class (blue) and second class (red). For Growth: boys (blue) and girls (red) and Heartbeat: normal (blue) and abnormal (red). (Color figure online)

could be used to develop strategies to control this problem. It could also be used to develop better treatments for other similar problems. We display some examples of signals represented by their PDFs registered on $I = [0, 1]$ and normalized to admit an unit integral in Fig. 1 (bottom-left). Moreover, the second dataset used in this analysis consists of monthly clinical growth charts for children from 1 to 12 years (100 girls and 100 boys) National Center for Health Statistics. It is a typical example of biological dynamics observed over months. Each growth chart represents the size (the increase in centimeters) of a child during 132 months. In this context, all growth charts were represented by PDFs of child sizes registered on $I = [0, 1]$, see some examples in Fig. 1 (bottom-right) for which we make the use of nonparametric kernel method with an automatic bandwidth.

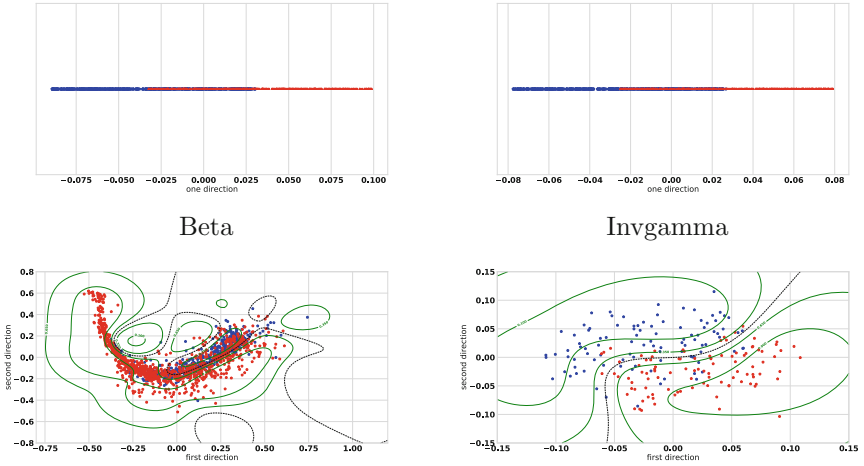

Fig. 2. Top: TPCA of projected coefficients into the tangent space of the sphere with first class (blue) and second class (red). Bottom: The predicted class "1" probabilities are shown in the contour plots. The black dashed line represents the decision boundary at $\bar{\pi}(Colorfigureonline)(C_i^{d,k}) = \frac{1}{2}$.

5.2 Tangent Principal Component Analysis

Tangent principal component analysis (TPCA) is a mathematical technique, also called Geodesic Component Analysis, used for dimensionality reduction and feature extraction in machine learning and data analysis. It is particularly useful for data embedded on curved manifolds. According to our case, this technique involves first computing the tangent space at each point on the finite-dimensional sphere $\mathcal{S}^{d-1}$ then performs to obtain a set of orthogonal basis vectors that capture the most important variations in data. If some point movements B_i^d were to be totally correlated manifold learning methods including: t-SNE, Isomap,

LLE, and MDS are useful for nonlinear dimensionality reduction. Since our vector data B_i^d are not of high-dimension ($d = 30$) and belong to the Euclidean tangent space we establish the TPCA. The central idea of TPCA is to reduce the dimensionality of projected vectors into the tangent space of $\mathcal{S}^{d-1}$ belonging to a linear sub-space of $\mathbb{R}^d$ by keeping one ($k = 1$), two ($k = 2$) or three ($k = 3$) dimensions in $\mathbb{R}^k$. This is achieved when transforming to a new set of variables, known as principal components (PCs) so that the first directions retain most of the variation presented in the original variables. First, we find the eigenvectors of the covariance matrix of the whole dataset B_i^d. Second, we sort the eigenvectors by decreasing the corresponding eigenvalues and choose k eigenvectors of the largest eigenvalues to be the principal directions. Finally, we transform the original data B_i^d into the new sub-space of reduced dimension $\mathbb{R}^k$. Let $C_i^{d,k}$ ($i = 1, \ldots, N$) be the resulting coefficients in $\mathbb{R}^k$. Generally, the variance ratio indicates the proportion of the total variance that is accounted by each principal component. Specifically, principal components with high variance ratios are considered to be more important and should be retained, while those with low variance ratios may be discarded. In Fig. 2 (top) we show results of the coefficients projected into one principal direction for Beta and Invgamma datasets. Indeed, only one principal component ($k = 1$) accounts for the largest proportion of the variance, with a variance ratio of 0.99 for both. Figure 2 (bottom) shows a scatter plot of the data with the first two principal components ($k = 2$) for the Heartbeat and Growth datasets. The first principal component (which explains 67% of the variance for Heartbeat and 38% for Growth) separates the two classes along the x-axis, while the second principal component (which explains 19% of the variance for Heartbeat and 15% for Growth) separates the classes along the y-axis. Although the action of TPCA is not by isometry but only a dimensionality reduction technique that finds the directions of maximum variance we add the contour plot in each region associated with the predicted class "1" probability that shows how GPc can be successfully performed in low-dimensional tangent spaces mainly when real data are not linearly separable.

5.3 Results and Comparison

To evaluate the performance of the proposed method, we split the labeled dataset into two subsets: training and test. The training set (75% of the dataset: 50% for training and 25% for validation) is used to train the model, while the test set (25% of the dataset) is used to evaluate its performance. Some commonly used metrics for evaluating the performance of a classification model include:

- Accuracy: The proportion of correctly classified instances in the test set.
- AUC: The measure of the overall performance of the model based on the ROC curve.
- LOSS: The measure of the logarithmic (also known as cross-entropy loss) between the predicted probability distribution and the true label.

In order to get an accurate estimate of the model's performance, we perform multiple random splits of the dataset into training and test sets, and train and

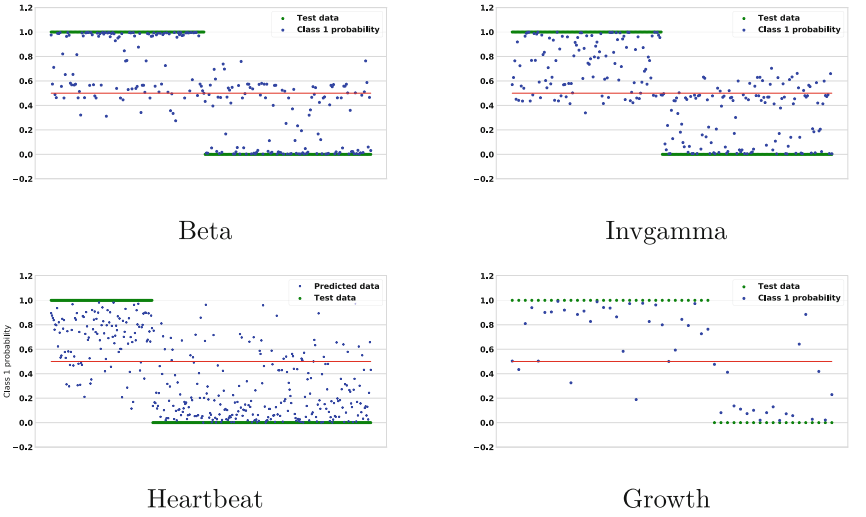

Fig. 3. The classification results with first class (label 1) and second class (label 0).

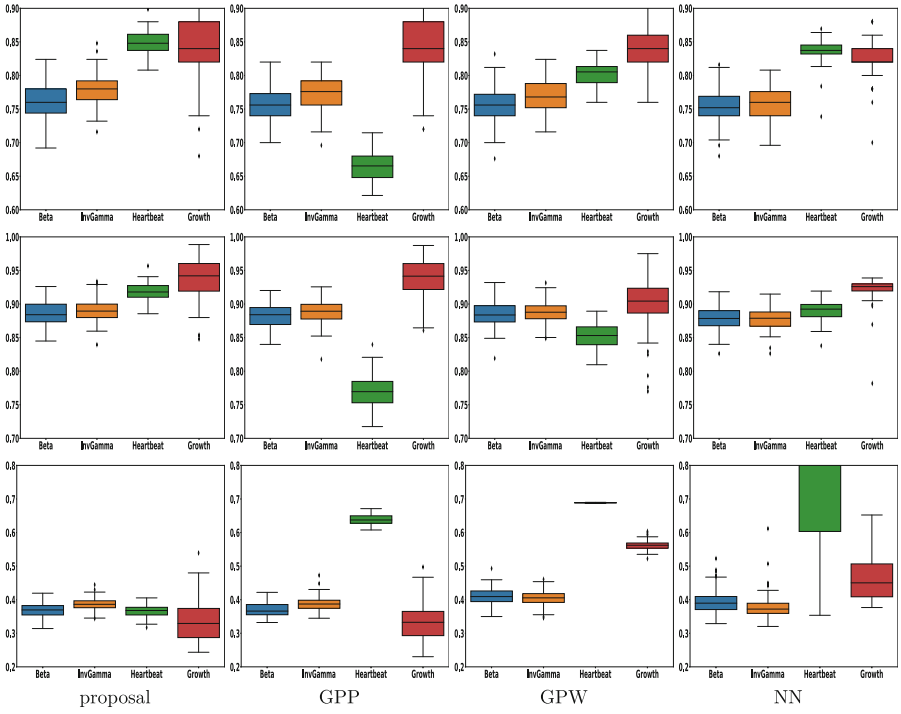

Fig. 4. The boxplots of different metrics: Accuracy score (top), AUC score (middle), and LOSS measure (bottom).

test the model on each split which could reduce the variance in the performance metrics obtained from a single split. The predicted class probabilities in our model provide a measure of uncertainty in the model's predictions and used to make informed decisions based on the level of confidence in the classification result. Now, we show results of the predicted class "1" probabilities of one among 100 runs in Fig. 3. The observed values involve computing the mean and variance of the conditional distribution of the output labels given the input data and the model parameters, and then using them to compute the predicted class probabilities. We state that most well classified test data are far from the decision boundary at $\bar{\pi}(B_i^d) = \frac{1}{2}$, which gives a good precision to our method.

At this stage, we will compare the results of our approach with some baseline methods: i) GPs indexed by PDFs (GPP), ii) GPs based on the Wasserstein distance (GPW), and iii) neural network (NN) model for classifying univariate functional data to determine whether the differences in performance are significant. We remind that standard classification models are not suitable for curved spaces and can not be applied in this context. For an attempt to show it is different, we provide some details about the NN model architecture. We first define the NN model using Keras' Sequential function in Python. The model has an input layer equal to the number of time instances of each observation. The first hidden layer a fully connected layer with 32 neurons and a ReLU activation function, followed by a dropout (regularization) layer that randomly sets 50% of the input units to 0 during training to prevent overfitting. Then, we add a second hidden layer with 16 neurons and ReLU activation, followed by another dropout layer with a dropout rate of 50%. Finally, the model has an output layer with one neuron and sigmoid activation. This produces a scalar output between 0 and 1, representing the model's prediction for the binary classification problem. We compile the model with binary cross-entropy loss and Adam optimizer.

In Fig. 4 we illustrate boxplots of the accuracy, AUC and LOSS metrics for the binary classification problem across the 100 runs of the model. The boxplots of most dataset are relatively narrow for the Accuracy and AUC scores, indicating that these metrics are consistent across different runs. However, we also see a few outliers with other datasets that are in somewhat lower/higher than the rest, which may indicate that there are some runs where the model is performing poorly or exceptionally well. Since most criteria values are sometimes very close for different methods which rends comparison nontrivial we also summarize the mean and the standard deviations (std) values in Table 1. Accordingly, our proposed method achieved a mean accuracy of 0.761, 0.779, 0.849 and 0.847 for Beta, InvGamma, Heartbeat and Growth, which is significantly better than the baseline GPP, GPW and NN. However, our proposed method outperformed the same methods in terms of AUC, achieving a score of 0.885, 0.891, 0.918 and 0.938, respectively, see Table 2. Regarding the LOSS measure in Table 3, our proposed method achieves a lower value for three among four datasets: Beta, Heartbeat and Growth. Overall, our proposed method showed promising results and outperformed the baseline methods on all datasets in terms of accuracy and AUC, while it still competitive in terms of LOSS measure. Our method, on the

other hand, is designed to be computationally efficient. This is because it considers some coefficients instead of PDFs directly that are optimized for an efficient training. This allows our method to achieve comparable or better accuracy than traditional methods, while requiring less computational resources. To illustrate this, let's compare the computational time of our method against the baseline methods, particularly for the Beta dataset. We assume that all programs run on a desktop machine with 32 GB memory and CPU Xeon(R) 3.60 GHz. Note that the elapsed times for predicting all the Beta test set are 10^{-3}, 8.9×10^{-4} and 3×10^{-2} seconds using GPW, GPP, and NN respectively while it takes 5.7×10^{-4} seconds for our proposal.

Table 1. Accuracy score.

Dataset	Proposal		GPP		GPW		NN	
	mean	std	mean	std	mean	std	mean	std
Beta	**0.761**	0.027	0.757	0.025	0.757	0.026	0.755	0.025
InvGamma	**0.779**	0.023	0.773	0.024	0.77	0.024	0.757	0.025
Heartbeat	**0.849**	0.017	0.666	0.022	0.802	0.018	0.837	0.015
Growth	**0.847**	0.046	0.841	0.047	0.844	0.036	0.825	0.025

Table 2. AUC score.

Dataset	Proposal		GPP		GPW		NN	
	mean	std	mean	std	mean	std	mean	std
Beta	**0.885**	0.018	0.882	0.018	0.884	0.018	0.878	0.017
InvGamma	**0.891**	0.016	0.887	0.017	0.888	0.016	0.878	0.016
Heartbeat	**0.918**	0.004	0.768	0.024	0.853	0.018	0.89	0.014
Growth	**0.938**	0.03	0.923	0.029	0.901	0.037	0.923	0.017

Table 3. LOSS measure.

Dataset	Proposal		GPP		GPW		NN	
	mean	std	mean	std	mean	std	mean	std
Beta	**0.368**	0.021	0.371	0.021	0.41	0.024	0.393	0.035
InvGamma	0.387	0.017	0.388	0.021	0.406	0.022	**0.378**	0.038
Heartbeat	**0.367**	0.018	0.638	0.015	0.689	0.001	0.919	0.343
Growth	**0.334**	0.058	0.337	0.056	0.563	0.014	0.464	0.063

6 Conclusion

In this paper, we have introduced a novel approach for classifying probability density functions with a Gaussian process classifier model. Our methodology benefits from the use of functions decomposed with coefficients projected into the tangent space of the sphere, which can perform inference on PDFs. The theoretical foundation detailed in this paper exploits the simple geometry implied the nonparametric Fisher-Rao metric. The experimental evaluation has demonstrated that this new model is competitive on several challenging datasets. Furthermore, the problem formulation can be extended to many other supervised and unsupervised areas of statistical machine learning. Nevertheless, it would be very interesting to further investigate substantial impacts on the computational costs.

References

1. Alexander, A., Lee, J., Lazar, M., Field, A.: Diffusion tensor imaging of the brain. Neurother. J. Am. Soc. Exp. NeuroTher **4**, 316–29 (2007)
2. Alpaydin, E.: Introduction to Machine Learning, 2nd edn. MIT Press, Cambridge, MA (2010)
3. Amari, Si.: Differential geometry of statistical inference. In: Prokhorov, J.V., Itô, K. (eds.) Probability Theory and Mathematical Statistics. Lecture Notes in Mathematics. vol 1021. Springer, Berlin, Heidelberg (1983). https://doi.org/10.1007/BFb0072900
4. Bachoc, F., Gamboa, F., Loubes, J.M., Venet, N.: A gaussian process regression model for distribution inputs. IEEE Trans. Inf. Theor. **64**, 6620–6637 (2018)
5. Botev, Z.I., Grotowski, J.F., Kroese, D.P.: Kernel density estimation via diffusion. Ann. Stat. **38**, 2916–2957 (2010)
6. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth International Group, Belmont, CA (1984)
7. Cardot, H., Ferraty, F., Sarda, P.: Classification of functional data: a segmentation approach. Comput. Stat. Data Anal. **44**, 315–337 (2003)
8. Cencov, N.N.: Evaluation of an unknown distribution density from observations. Doklady **3**, 1559–1562 (1962)
9. Chen, L., Li, J.: Fraud detection for credit cards using random forest. J. Financ. Data Science **1**, 83–94 (2018)
10. Djolonga, J., Krause, A., Cevher, V.: High-dimensional Gaussian process bandits, pp. 1025–1033. NIPS2013, Curran Associates Inc., Red Hook, NY, USA (2013)
11. Fradi, A., Feunteun, Y., Samir, C., Baklouti, M., Bachoc, F., Loubes, J.M.: Bayesian regression and classification using gaussian process priors indexed by probability density functions. Inf. Sci. **548**, 56–68 (2021)
12. Holbrook, A., Lan, S., Streets, J., Shahbaba, B.: Nonparametric fisher geometry with application to density estimation. In: Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI), pp. 101–110. Proceedings of Machine Learning Research (2020)
13. Hosmer, D.W., Lemeshow, S., Sturdivant, R.X.: Applied Logistic Regression, 3rd edn. Wiley, Hoboken, NJ (2013)
14. James, G., Witten, D., Hastie, T., Tibshirani, R.: An Introduction to Statistical Learning. STS, vol. 103. Springer, New York (2013). https://doi.org/10.1007/978-1-4614-7138-7

15. Julian, P.R., Murphy, A.H.: Probability and statistics in meteorology: a review of some recent developments. Bull. Am. Meteorol. Soc. **53**, 957–965 (1972)
16. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 3rd edn. Pearson Education, Harlow, England (2020)
17. Kanagawa, M., Hennig, P., Sejdinovic, D., Sriperumbudur, B.K.: Gaussian processes and kernel methods: A review on connections and equivalences (2018)
18. Kim, H.J., et al.: Multivariate general linear models (MGLM) on Riemannian manifolds with applications to statistical analysis of diffusion weighted images. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2705–2712. IEEE Computer Society, Los Alamitos, CA, USA (2014)
19. Kneip, A., Ramsay, J.O.: Combining registration and fitting for functional models. J. Am. Stat. Assoc. **103**, 1155–1165 (2008)
20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. **25**, 1097–1105 (2012)
21. Lagani, V., Fotiadis, D.I., Likas, A.: Functional data analysis via neural networks: an application to speaker identification. Expert Syst. Appl. **39**, 9188–9194 (2012)
22. Mallasto, A., Feragen, A.: Wrapped Gaussian process regression on Riemannian manifolds. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5580–5588. IEEE Computer Society, Los Alamitos, CA, USA (2018)
23. Oliva, J.B., Neiswanger, W., Póczos, B., Schneider, J.G., Xing, E.P.: Fast distribution to real regression. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, pp. 706–714 (2014)
24. Patrangenaru, V., Ellingson, L.: Nonparametric Statistics on Manifolds and their Applications to Object Data Analysis, 1st edn. Chapman & Hall/CRC Monographs on Statistics & Applied Probability, CRC Press Inc, USA (2015)
25. Póczos, B., Singh, A., Rinaldo, A., Wasserman, L.: Distribution-free distribution regression. In: Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, pp. 507–515. Proceedings of Machine Learning Research, Scottsdale, Arizona, USA (2013)
26. Lopez de Prado, M.: Advances in financial machine learning. Wiley, Hoboken, New Jersey (2018)
27. Ramsay, J.O., Silverman, B.W.: Functional Data Analysis. Springer, New York (2005). https://doi.org/10.1007/b98888
28. Rasmussen, C.E., Williams, C.: Gaussian Processes for Machine Learning. The MIT Press, Cambridge, London (2006)
29. Samir, C., Loubes, J.-M., Yao, A.-F., Bachoc, F.: Learning a gaussian process model on the Riemannian manifold of non-decreasing distribution functions. In: Nayak, A.C., Sharma, A. (eds.) PRICAI 2019. LNCS (LNAI), vol. 11671, pp. 107–120. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29911-8_9
30. Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Adaptive computation and machine learning (2002)
31. Srivastava, A., Klassen, E., Joshi, S.H., Jermyn, I.H.: Shape analysis of Elastic curves in Euclidean spaces. IEEE Trans. Pattern Anal. Mach. Intell. **33**, 1415–1428 (2011)
32. Terenin, A.: Gaussian processes and statistical decision-making in non-Euclidean spaces. arXiv (2022). https://arxiv.org/abs/2202.10613
33. Yao, F., Müller, H.G., ling Wang, J.: Functional data analysis for sparse longitudinal data. J. Am. Stat. Assoc. **100**, 577–590 (2005)