A Bayesian approach to functional regression: theory and computation

Preprint

José R. Berrendero[*][1,2], **Antonio Coín**[†][1], and **Antonio Cuevas**[‡][1,2]
[1]Departamento de Matemáticas, Universidad Autónoma de Madrid (UAM), Madrid, Spain
[2]Instituto de Ciencias Matemáticas ICMAT (CSIC-UAM-UC3M-UCM), Madrid, Spain

December 21, 2023

Abstract

We propose a novel Bayesian methodology for inference in functional linear and logistic regression models based on the theory of reproducing kernel Hilbert spaces (RKHS's). These models build upon the RKHS associated with the covariance function of the underlying stochastic process, and can be viewed as a finite-dimensional approximation to the classical functional regression paradigm. The corresponding functional model is determined by a function living on a dense subspace of the RKHS of interest, which has a tractable parametric form based on linear combinations of the kernel. By imposing a suitable prior distribution on this functional space, we can naturally perform data-driven inference via standard Bayes methodology, estimating the posterior distribution through Markov chain Monte Carlo (MCMC) methods. In this context, our contribution is two-fold. First, we derive a theoretical result that guarantees posterior consistency in these models, based on an application of a classic theorem of Doob to our RKHS setting. Second, we show that several prediction strategies stemming from our Bayesian formulation are competitive against other usual alternatives in both simulations and real data sets, including a Bayesian-motivated variable selection procedure.

Keywords functional data · linear regression · logistic regression · reproducing kernel Hilbert space · Bayesian inference · posterior consistency

1 Introduction

The problem of predicting a scalar response from a functional covariate is one that has gained traction over the last few decades, as more and more data is being generated with an ever-increasing level of granularity in the measurements. While in principle the functional data could simply be regarded as a discretized vector in a very high dimension, there are often many advantages in taking into account the functional nature of the data, ranging from modeling the possibly high correlation among points that are close in the domain, to extracting information that may be hidden in the derivatives of the function in question. As a consequence, numerous proposals have arisen on how to suitably deal with functional data, all of them encompassed under the term Functional Data Analysis (FDA), which essentially explores statistical techniques to process, model and make inference on data varying over a continuum. A partial survey on such methods is Cuevas (2014) or Goia and Vieu (2016), while a more detailed exposition of the theory and applications can be found for example in Ramsay and Silverman (2005) or Hsing and Eubank (2015).

FDA is undoubtedly an active area of research, which finds applications in a wide variety of fields, such as biomedicine, finance, meteorology or chemistry (see for example Ullah and Finch, 2013). Accordingly,

[*] joser.berrendero@uam.es
[†] antonio.coin@uam.es (corresponding author)
[‡] antonio.cuevas@uam.es

there are many recent contributions on how to tackle functional data problems, both from a theoretical and practical standpoint. Chief among them is the approach of reducing the problem to a finite-dimensional one, for example using a truncated basis expansion or spline interpolation methods (e.g. Müller and Stadtmüller, 2005; Aguilera and Aguilera-Morillo, 2013). At the same time, much effort has also been put into the task of building a sound theoretical basis for FDA, generalizing different concepts to the infinite-dimensional framework. Examples of this endeavor include the definition of centrality measures and depth-based notions for functional data (e.g. Fraiman and Muniz, 2001; Cuevas et al., 2007; López-Pintado and Romo, 2009), an ANOVA test for functional data (Cuevas et al., 2004), a purely functional partial least squares algorithm (Delaigle and Hall, 2012b), a functional Mahalanobis distance (e.g. Galeano et al., 2015; Berrendero et al., 2020a), or an extension of Fisher's discriminant analysis for random functions (e.g. James and Hastie, 2001; Shin, 2008), among many others. Moreover, additional non-parametric methods for functional prediction and classification were notably explored in Ferraty and Vieu (2006). As the name suggests, FDA techniques are heavily inspired by functional analysis tools and methods: Hilbert spaces, orthonormal systems, linear operators, and so on. Incidentally, a notion that also intersects with the classical theory of machine learning and pattern recognition, and that has gained popularity in recent years, is that of reproducing kernel Hilbert spaces (RKHS's) and their applications in functional data problems (see for example Yuan and Cai, 2010 or Berrendero et al., 2018, 2020b).

On the other hand, Bayesian inference methods are ubiquitous in the realm of statistics, and their usual non-parametric approach also makes use of random functions, though in a slightly different manner than in the FDA context. However, while there are recent works in the literature that offer a Bayesian treatment of functional data (e.g. Scarpa and Dunson, 2009; Crainiceanu and Goldsmith, 2010; Shi and Choi, 2011; Kang et al., 2023), there is still no systematic approach to Bayesian methodologies within the FDA framework. It is precisely at this relatively unexplored intersection between FDA and Bayesian methods that our work is aimed. In particular, our goal is to study functional regression problems, which are the infinite-dimensional equivalents of the usual regression problems that appear in statistics and machine learning. We follow the path started by Ferguson (1974) of setting a prior distribution on a functional space, but in our case we take a particular RKHS as the ambient space, obtaining functional regression models that allow for a simple yet efficient Bayesian treatment of functional data. Moreover, we also study the basic theoretical question of posterior consistency in these RKHS models within the proposed Bayesian framework. The concepts of consistency and posterior concentration are a kind of frequentist validation that have arguably been an active point of research in the last few decades, particularly in infinite-dimensional settings (see Amewou-Atisso et al., 2003; Ghosh and Ramamoorthi, 2003; Choi and Ramamoorthi, 2008), and also in the functional regression case (e.g. Lian et al., 2016; Abraham and Grollemund, 2020). To put it simply, posterior consistency ensures that with enough samples, the Bayesian updating mechanism works as intended and the posterior distribution eventually concentrates around the true value of the parameters, supposing the model is well specified. We leverage the properties of RKHS's and existing techniques, mainly by Nobile (1994) and Miller (2023), to show that posterior consistency holds in our RKHS models under some mild identifiability conditions, thus providing a strong and coherent background to our Bayesian approach. Finally, this theoretical side is complemented by extensive experimentation that demonstrates the predictive performance of the proposed functional regression models, especially when compared with other usual frequentist methods.

L^2-models, shortcomings and alternatives

In this work we are concerned with functional linear and logistic regression models, that is, situations where the goal is to predict a continuous or dichotomous variable from functional observations. Even though these problems can be formally stated with almost no differences from their finite-dimensional counterparts, there are some fundamental challenges as well as some subtle drawbacks that emerge as a result of working in infinite dimensions. To set a common framework, we will consider throughout a scalar response variable Y (either continuous or binary) which has some dependence on a stochastic L^2-process $X = X(t) = X(t, \omega)$ with trajectories in $L^2[0, 1]$. We will further suppose that X is centered, that is, its mean function $m(t) = \mathbb{E}[X(t)]$ vanishes for all $t \in [0, 1]$. In addition, when prediction is our ultimate objective, we will tacitly assume the existence of a *labeled* data set $\mathcal{D}_n = \{(X_i, Y_i) : i = 1, \dots, n\}$ of independent observations from (X, Y), and our aim will be to accurately predict the response corresponding to unlabeled samples from X.

The most common scalar-on-function linear regression model is the classical L^2-model, widely popularized since the first edition (1997) of the monograph by Ramsay and Silverman (2005). It can be seen as a generalization of the usual finite-dimensional model, replacing the scalar product in $\mathbb{R}^d$ for that of the functional space $L^2[0,1]$ (henceforth denoted by $\langle\cdot,\cdot\rangle$):

$$Y = \alpha_0 + \langle X, \beta\rangle + \varepsilon = \alpha_0 + \int_0^1 X(t)\beta(t)\,dt + \varepsilon, \tag{1.1}$$

where $\alpha_0 \in \mathbb{R}$, ε is a random error term independent from X with $\mathbb{E}[\varepsilon] = 0$, and the functional slope parameter $\beta = \beta(\cdot)$ is assumed to be a member of the infinite-dimensional space $L^2[0,1]$. In this case, the inference on β is hampered by the fact that $L^2[0,1]$ is an extremely wide space that contains many non-smooth or ill-behaved functions, so that any estimation procedure involving optimization on it would typically be hard. In spite of this, model (1.1) is not flexible enough to include "simple" finite-dimensional models based on linear combinations of the marginals, such as $Y = \alpha_0 + \beta_1 X(t_1) + \cdots + \beta_p X(t_p) + \varepsilon$ for some constants $\beta_j \in \mathbb{R}$ and instants $t_j \in [0,1]$; see Berrendero et al. (2020b) for additional details on this. Moreover, the non-invertibility of the covariance operator associated with X, which plays the role of the covariance matrix in the infinite case, invalidates the usual least squares theory (see e.g. Cardot and Sarda, 2018). Thus, some regularization or dimensionality reduction technique is needed for parameter estimation; see Reiss et al. (2017) for a summary of several widespread methods.

A similar L^2-based functional logistic equation can be derived for the binary classification problem via the logistic function:

$$\mathbb{P}(Y = 1 \mid X) = \frac{1}{1 + \exp\{-\alpha_0 - \langle X, \beta\rangle\}}, \tag{1.2}$$

where $\alpha_0 \in \mathbb{R}$ and $\beta \in L^2[0,1]$. In this situation, the most common way of estimating the slope function β is via its Maximum Likelihood Estimator (MLE). However, not only do the same complications as in the linear regression model apply in this situation, but there is also the additional problem that in functional settings the MLE does not exist with probability one under fairly general conditions (see Berrendero et al., 2023).

It turns out that in both scenarios a natural alternative to the L^2-model is the so-called reproducing kernel Hilbert space (RKHS) model, which instead assumes the unknown functional parameter to be a member of the RKHS associated with the covariance function of the process X, making use of the scalar product of that space. As we will show later on, not only is this model simpler and arguably easier to interpret, but it also constrains the parameter space to smoother and more manageable functions. In fact, it does include a model based on finite linear combinations of the marginals of X as a particular case, which is especially appealing to practitioners confronted with functional data problems due to its simplicity. These RKHS-based models and their idiosyncrasies have been explored in Berrendero et al. (2019, 2020b) in the functional linear regression setting, and in Berrendero et al. (2023) for the case of functional logistic regression. Incidentally, these models also shed light on the near-perfect classification phenomenon for functional data, described by Delaigle and Hall (2012a) and further examined for example in the works of Berrendero et al. (2018) or Torrecilla et al. (2020).

A major aim of this work is to motivate the aforementioned RKHS models inside the functional framework, while also providing efficient techniques to apply them in practice. Our main contribution is the proposal of a Bayesian approach to parameter estimation and prediction within these models, in which a prior distribution is imposed on the unknown functional parameter and the posterior distribution is used as a basis for several prediction techniques. Following recent trends in Bayesian computation techniques, this posterior distribution is approximated via generic Markov chain Monte Carlo (MCMC) methods (e.g. Brooks et al., 2011). Although setting a prior distribution on a function space is generally a hard task, the specific parametric formulation of the RKHS models we propose greatly facilitates this (see Section 2 for details). Similar Bayesian schemes have recently been explored in Grollemund et al. (2019) and Abraham (2023), albeit not within a RKHS framework. Another set of techniques extensively studied in this context are variable selection methods, which aim to select the marginals $\{X(t_j)\}$ of the process that better summarize it according to some optimality criterion. As it happens, some variable selection methods have already been proposed in the RKHS framework (e.g. Berrendero et al., 2019; Bueno-Larraz and Klepsch, 2019), but in general they have their own dedicated algorithms and procedures. As will become apparent in the forthcoming sections, given the nature of our suggested Bayesian model we can easily isolate the marginal posterior distribution corresponding to a finite set of points $\{t_j\}$, and thus provide a Bayesian-motivated variable selection process along with the other prediction methods that naturally arise within our model. These points-of-impact selection models for functional predictors have also been considered in the literature; see Ferraty et al. (2010), Berrendero et al. (2016) or Poß et al. (2020) by way of illustration. Another example of a related strategy is the work of James et al. (2009), in which the authors propose a method based on variable selection to estimate the functional parameter $\beta(t)$ in such a way that it is exactly zero over some regions in the domain.

Some essentials on RKHS's and notation

The methodology proposed in this work relies heavily on the use of RKHS's, so before diving into it, we will briefly describe the main characteristics of these spaces from a probabilistic point of view (for a more detailed account, see for example Berlinet and Thomas-Agnan, 2004). Let us denote by $K(t,s) = \mathbb{E}[X(t)X(s)]$ the covariance function of the centered process X, and in what follows suppose that it is continuous. To construct the RKHS $\mathcal{H}(K)$ associated with the covariance function, we start by defining the functional space $\mathcal{H}_0(K)$ of all finite linear combinations of evaluations of K, that is,

$$\mathcal{H}_0(K) = \left\{ f \in L^2[0,1] : \ f(\cdot) = \sum_{i=1}^{p} a_i K(t_i, \cdot), \ p \in \mathbb{N}, \ a_i \in \mathbb{R}, \ t_i \in [0,1] \right\}. \qquad (1.3)$$

This space is endowed with the inner product $\langle f, g\rangle_K = \sum_{i,j} a_i b_j K(t_i, s_j)$, where $f(\cdot) = \sum_i a_i K(t_i, \cdot)$ and $g(\cdot) = \sum_j b_j K(s_j, \cdot)$. Then, $\mathcal{H}(K)$ is defined to be the completion of $\mathcal{H}_0(K)$ under the norm induced by the scalar product $\langle \cdot, \cdot \rangle_K$. As it turns out, functions in this space satisfy the so-called *reproducing property* $\langle K(t,\cdot), f\rangle_K = f(t)$, for all $f \in \mathcal{H}(K)$ and $t \in [0,1]$. An important consequence of this is that $\mathcal{H}(K)$ is a space of genuine functions and not of equivalence classes, since the values of the functions at particular points are in fact relevant, unlike in L^2-spaces.

Now, a particularly useful approach in statistics is to regard $\mathcal{H}(K)$ as an isometric copy of a well-known space. Specifically, via *Loève's isometry* (Loève, 1948) one can establish a congruence Ψ_X between $\mathcal{H}(K)$ and the linear span of the process, $\mathcal{L}(X)$, in the space of all random variables with finite second moment, $L^2(\Omega)$ (see Lemma 1.1 in Lukić and Beder, 2001). This isometry is essentially the completion of the correspondence

$$\sum_{i=1}^{p} a_i X(t_i) \longleftrightarrow \sum_{i=1}^{p} a_i K(t_i, \cdot), \qquad (1.4)$$

and can be formally defined, in terms of its inverse, as $\Psi_X^{-1}(U)(t) = \mathbb{E}[UX(t)]$ for $U \in \mathcal{L}(X)$. Despite the close connection between the process X and the space $\mathcal{H}(K)$, special care must be taken when dealing with concrete realizations of the process, since under rather general conditions the trajectories of X do not belong to the corresponding RKHS with probability one (see for example Lukić and Beder, 2001, Corollary 7.1). As a consequence, the expression $\langle x, f\rangle_K$ is ill-defined and lacks meaning when x is a realization of X. However, following Parzen's approach in his seminal work (e.g. Parzen, 1961, Theorem 4E), we can leverage Loève's isometry and identify $\langle x, f\rangle_K$ with the image $\Psi_x(f) := \Psi_X(f)(\omega)$, for $x = X(\omega)$ and $f \in \mathcal{H}(K)$. This notation, viewed as a formal extension of the inner product, often proves to be useful and convenient.

Organization of the paper

The rest of the paper is organized as follows. In Section 2 we explain the Bayesian methodology and the functional regression models we propose. Section 3 is devoted to presenting a positive posterior consistency result, along with an overview of the proof. The empirical results of the experimentation are contained in Section 4, which includes a short discussion of computational details. Lastly, the conclusions drawn from this work are presented in Section 5.

2 A Bayesian methodology for RKHS-based functional regression models

In this section we present the precise functional models and Bayesian methodologies explored in this work. The RKHS-based functional models under consideration are those obtained by taking a functional parameter $\alpha \in \mathcal{H}(K)$ and replacing the scalar product for $\langle X, \alpha\rangle_K$ in the L^2-models (1.1) and (1.2). However, to further simplify things we will follow a parametric approach and suppose that α is in fact a member of the dense subspace $\mathcal{H}_0(K)$ defined in (1.3). The general idea will be to impose a prior distribution on this functional parameter to derive an approximate posterior model (via MCMC methods) after incorporating the available sample information. Moreover, as we said before, with a slight abuse of notation we will understand the expression $\langle x, \alpha\rangle_K$ as $\Psi_x(\alpha)$, where $x = X(\omega)$ and Ψ_x is Loève's isometry. Hence, taking into account that $\alpha \in \mathcal{H}_0(K)$ and that $\Psi_X(K(t,\cdot)) = X(t)$ by definition, we can write $\langle x, \alpha\rangle_K \equiv \sum_j \beta_j x(t_j)$ when $\alpha(\cdot) = \sum_j \beta_j K(t_j, \cdot)$. In this way we get a simpler, finite-dimensional approximation of the functional RKHS model, which we argue reduces the overall complexity of the model while still capturing most of the relevant information. Moreover, the model remains "truly functional", in the sense that we are exploiting the RKHS perspective to give a functional nature to discretized models.

In view of (1.3) and Loève's isometry, to set a prior distribution on the unknown function α (that is, a prior distribution on the functional space $\mathcal{H}_0(K)$) it suffices to consider a discrete distribution on p, and then impose p-dimensional continuous prior distributions on the coefficients β_j and the times t_j given p. Thanks to this parametric approach, the challenging task of setting a prior distribution on a space of functions is considerably simplified, while simultaneously not constraining the model to any specific distribution (in contrast to, say, Gaussian process regression methods). Moreover, note that starting from a probability distribution $\mathbb{P}_0$ on $\mathcal{H}_0(K)$ we can obtain a probability distribution $\mathbb{P}$ on $\mathcal{H}(K)$ merely by defining $\mathbb{P}(B) = \mathbb{P}_0(B \cap \mathcal{H}_0(K))$ for all Borel sets B. Consequently, our simplifying assumption on α is not very restrictive, since any prior distribution on $\mathcal{H}_0(K)$ can be directly extended to a prior distribution on $\mathcal{H}(K)$.

However, after some initial experimentation we found that, for practical and computational reasons, the value of $p \in \mathbb{N}$ (the dimensionality of the model) is best fixed beforehand in a suitable way; see Appendix A.2 for details. Thus, we will regard only the β_j and t_j as free parameters, and search for our functional parameter in the space

$$\mathcal{H}_{0,p}(K) = \left\{ \sum_{j=1}^{p} \beta_j K(t_j, \cdot) : \ \beta_j \in \mathbb{R}, \ t_j \in [0,1] \right\}. \tag{2.1}$$

Even though we actually work on $\mathcal{H}_{0,p}(K)$, the discrete parameter p can still be selected in several meaningful ways that make use of the available data (such as cross-validation or some information criteria), and the set of feasible values is not very large in practice: we advocate for simple, parsimonious models. Moreover, we could think of this approach as imposing a degenerate prior distribution on p, so it is in a way a particular case of the more general model discussed above.

2.1 Functional linear regression

In the case of functional linear regression, the simplified RKHS model considered is

$$Y = \alpha_0 + \langle X, \alpha \rangle_K + \varepsilon = \alpha_0 + \sum_{j=1}^{p} \beta_j X(t_j) + \varepsilon, \tag{2.2}$$

where $\alpha(\cdot) = \sum_{j=1}^{p} \beta_j K(t_j, \cdot) \in \mathcal{H}_{0,p}(K)$, $\alpha_0 \in \mathbb{R}$, and $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ is an error term independent from X. This model is essentially a finite-dimensional approximation from a functional perspective to the more general RKHS model that assumes $\alpha \in \mathcal{H}(K)$, proposed in Berrendero et al. (2019). When p is fixed, the parameter space of dimension $2p+2$ becomes $\Theta_p = \mathbb{R}^p \times [0,1]^p \times \mathbb{R} \times \mathbb{R}^+$, and in the sequel a generic element of this space will be denoted by $\theta = (\beta_1, \ldots, \beta_p, t_1, \ldots, t_p, \alpha_0, \sigma^2) \equiv (b, \tau, \alpha_0, \sigma^2)$. Before proceeding any further, observe that we can rewrite model (2.2) in a more explicit and practical fashion in terms of the available sample information in $\mathcal{D}_n$. For $\theta \in \Theta_p$, the reinterpreted model assumes the form

$$Y_i \mid X_i, \theta \overset{\text{ind.}}{\sim} \mathcal{N}\left(\alpha_0 + \sum_{j=1}^{p} \beta_j X_i(t_j), \ \sigma^2 \right), \quad i = 1, \ldots, n. \tag{2.3}$$

It is worth mentioning that the model remains linear in the sense that it fundamentally involves a random variable $\langle X, \alpha \rangle_K = \Psi_X(\alpha)$ belonging to the linear span of the process X in $L^2(\Omega)$. Also, note that given the time instants t_j, the model becomes a multiple linear model with the $X(t_j)$ as scalar covariates. As a matter of fact, this RKHS model is particularly suited as a basis for variable selection methods, and furthermore the general RKHS model entails the classical L^2-model (1.1) under certain conditions (see Berrendero et al., 2020b, Section 3). In addition, this model could be easily extended to the case of several covariates via an expression of type $Y = \alpha_0 + \Psi_{X^1}(\alpha_1) + \cdots + \Psi_{X^q}(\alpha_q) + \varepsilon$. In that case, as argued in Grollemund et al. (2019) for a similar situation, if we were to set a prior distribution on all the parameters involved, we could recover the full posterior by looking alternately at the posterior distribution of each covariate conditional on the rest of them.

The Bayesian approach: prior selection and posterior derivation

A simple, natural prior distribution for the parameter vector $\theta \in \Theta_p$, suggested by the structure of the parameter space and usually employed in similar situations in the Bayesian literature, is given by

$$\begin{aligned} \pi(\alpha_0, \sigma^2) &\propto 1/\sigma^2, \\ \tau &\sim \mathcal{U}([0,1]^p), \\ b \mid \tau, \sigma^2 &\sim \mathcal{N}_p(b_0, g\sigma^2 \underbrace{\left(\mathcal{X}_\tau^T \mathcal{X}_\tau + \eta I\right)^{-1}}_{G_\tau}), \end{aligned} \tag{2.4}$$

where I is the identity matrix, $\mathcal{X}_\tau$ is the data matrix $(X_i(t_j))_{i,j}$, and $b_0 \in \mathbb{R}^p$, $g \in \mathbb{R}$ and $\eta \in \mathbb{R}^+$ are hyperparameters of the model. On the one hand, note the use of a joint prior distribution on α_0 and σ^2, which is a widely used non-informative prior known as Jeffrey's prior (Jeffreys, 1946). In any event, the estimation of $\alpha_0 = \mathbb{E}[Y]$ is straightforward, so it could have been left out of the model altogether. On the other hand, the prior on b is a slight modification of the well-known Zellner's g-prior (Zellner, 1986), in which a regularizing term is added to avoid ill-conditioning problems in the Gram matrix, obtaining a ridge-like Zellner prior controlled by the tuning parameter η (Baragatti and Pommeret, 2012). All in all, with a slight abuse of notation the proposed prior distribution becomes $\pi(\theta) = \pi(b|\tau, \sigma^2)\pi(\tau)\pi(\alpha_0, \sigma^2)$.

As for the posterior distribution, we only compute a function proportional to its log-density, since that is all that is needed for a MCMC algorithm to work. A standard algebraic manipulation in the posterior expression yields the following result:

Proposition 2.1. *Under the linear model (2.3), the prior distribution implied in (2.4) produces the log-posterior distribution*

$$\begin{aligned} \log \pi(\theta \mid \mathcal{D}_n) \propto \frac{1}{2\sigma^2} &\left(\|\boldsymbol{Y} - \alpha_0 \mathbf{1} - \mathcal{X}_\tau b\|^2 + \frac{1}{g}(b - b_0)^T G_\tau (b - b_0) \right) \\ &+ (p + n + 2)\log \sigma - \frac{1}{2}\log |G_\tau|, \end{aligned}$$

where $\boldsymbol{Y} = (Y_1, \dots, Y_n)^T$ and $\mathbf{1}$ is an n-dimensional vector of ones.

Making predictions

In order to generate predictions, we have to take into account that when performing the empirical posterior approximation, a MCMC algorithm is an iterative procedure that produces a chain of M approximate samples $\theta^{(m)*} = (b^{(m)*}, \tau^{(m)*}, \alpha_0^{(m)*}, (\sigma^2)^{(m)*})$ of the posterior distribution $\pi(\theta|\mathcal{D}_n)$. Assuming now a previously unseen test set $\mathcal{D}'_{n'}$ in the same conditions as $\mathcal{D}_n$, we propose to construct three different kinds of predictors:

Summarize-then-predict. If we consider a point-estimate statistic T, we can get the corresponding estimates $\hat{\theta} = (\hat{b}, \hat{\tau}, \hat{\alpha}_0, \hat{\sigma}^2) \equiv (T\{b^{(m)*}\}, T\{\tau^{(m)*}\}, T\{\alpha_0^{(m)*}\}, T\{(\sigma^2)^{(m)*}\})$, and then use these summaries of the marginal posterior distributions to predict the responses in the usual way following model (2.2), i.e.:

$$\hat{Y}_i = \hat{\alpha}_0 + \sum_{j=1}^{p} \hat{\beta}_j X_i(\hat{t}_j), \quad i = 1, \dots, n'. \tag{2.5}$$

Note that in this case the variance σ^2 is treated as a nuisance parameter. Although it contributes to measure the uncertainty in the approximations, its estimates are discarded in the final prediction.

Predict-then-summarize. Alternatively, we can look at the approximate posterior distribution as a whole, and compute the predictive distribution of the simulated responses at each step of the chain following model (2.3):

$$\boldsymbol{Y}^{(m)*} := \left\{ Y_i^{(m)*} \equiv Y_i \mid X_i, \theta^{(m)*} : \, i = 1, \dots, n' \right\}, \quad m = 1, \dots, M. \tag{2.6}$$

Then, we can take the mean of all such simulated responses as a proxy for each response variable, that is,

$$\hat{Y}_i = \frac{1}{M} \sum_{m=1}^{M} Y_i^{(m)*}, \quad i = 1, \dots, n'.$$

This method differs from the previous one in that it takes into account the full approximate posterior distribution instead of summarizing it directly.

Variable selection. Lastly, we can focus only on the marginal posterior distribution of $\tau|\mathcal{D}_n$ and select p time instants using a point-estimate statistic T as in our first strategy, but discarding the rest of the parameters. Specifically, we can consider the times $\hat{t}_j = T\{t_j^{(m)*}\}$ and reduce the original data set to just the $n \times p$ real matrix $\{X_i(\hat{t}_j) : i = 1, \dots, n,\ j = 1, \dots, p\}$. After this variable selection has been carried out, we can tackle the problem using a finite-dimensional linear regression model and apply any of the well-known prediction algorithms suited for this situation.

Note that these predictors can be obtained all at once after only one round of training (that is, an individual MCMC run to approximate the posterior distribution). As a consequence, what we have in practice is a single algorithm that can produce multiple predictors at the same computational cost, so that any of them can be chosen (or even switched back and forth) depending on the particularities of the problem at hand. Moreover, one could even contemplate an ensemble model in which some kind of aggregation of several of the available prediction methods is performed to produce a final result.

2.2 Functional logistic regression

In the case of functional logistic regression, we regard the binary response variable $Y \in \{0, 1\}$ as a Bernoulli random variable given the regressor $X = x \in L^2[0, 1]$, and as usual suppose that $\log(p(x)/(1 - p(x)))$ is linear in x, where $p(x) = \mathbb{P}(Y = 1|X = x)$. Then, following the approach suggested by Berrendero et al. (2023), a logistic RKHS model might be given, in terms of the correspondence $\langle X, \alpha\rangle_K = \Psi_X(\alpha)$, by the equation

$$\mathbb{P}(Y = 1 \mid X) = \frac{1}{1 + \exp\{-\alpha_0 - \langle X, \alpha\rangle_K\}}, \quad \alpha_0 \in \mathbb{R},\ \alpha \in \mathcal{H}_{0,p}(K). \tag{2.7}$$

Indeed, note that this can be seen as a finite-dimensional approximation (but still with a functional interpretation) to the general RKHS functional logistic model proposed by these authors, which can be obtained by replacing $\mathcal{H}_{0,p}(K)$ with the whole RKHS space $\mathcal{H}(K)$. Now, if we aim at a classification problem, our strategy will be similar to that followed in the functional linear model: after incorporating the sample information, we can rewrite (2.7) as

$$Y_i \mid X_i, \theta \overset{\text{ind.}}{\sim} \text{Bernoulli}(p_i), \quad i = 1, \dots, n, \tag{2.8}$$

with

$$p_i = \mathbb{P}(Y_i = 1 \mid X_i, \theta) = \frac{1}{1 + \exp\left\{-\alpha_0 - \displaystyle\sum_{j=1}^{p} \beta_j X_i(t_j)\right\}}, \quad i = 1, \dots, n, \tag{2.9}$$

where in turn $\alpha_0, \beta_j \in \mathbb{R}$ and $t_j \in [0, 1]$.

In much the same way as the linear regression model described above, this RKHS-based logistic regression model offers some advantages over the L^2-model. First and foremost, it has a more straightforward interpretation and allows for a workable Bayesian approach, as we will demonstrate below. Secondly, it can be shown that under mild conditions the general RKHS logistic functional model holds whenever the conditional distributions $X|Y = i$ $(i = 0, 1)$ are homoscedastic Gaussian processes, and in some cases it also entails the L^2-model (see Section 4 in Berrendero et al., 2023); this arguably provides a solid theoretical motivation for the reduced model. Furthermore, a maximum likelihood approach for parameter estimation, although not considered here, is possible as well. Indeed, the use of a finite-dimensional approximation mitigates the problem of non-existence of the MLE in the functional case. However, let us recall that even in finite-dimensional settings there are cases of quasi-complete separation in which the MLE does not exist (Albert and Anderson, 1984), though this issue could be partially circumvented using, for example, Firth's corrected estimator (Firth, 1993). Nevertheless, there are still cases in high-dimensional logistic regression in which the MLE may not exist, as exemplified in the theory recently developed by Sur and Candès (2019) and Candès and Sur (2020). In any event, we argue that the Bayesian RKHS model presented here is a compelling and feasible approach to functional logistic regression, since it bypasses the main difficulties of the usual maximum likelihood techniques.

The Bayesian approach: prior selection and posterior derivation

As far as prior distributions go, we propose to use the same ones as we did in (2.4) for the linear regression model. Note that in this case the nuisance parameter σ^2 only appears as part of the hierarchical prior distribution, and not in the final model. The posterior distribution is again derived after a routine calculation:

Proposition 2.2. *Under the logistic model* (2.8)*, the prior distribution implied in* (2.4) *produces the log-posterior distribution*

$$\log \pi(\theta \mid \mathcal{D}_n) \propto \sum_{i=1}^{n} \left[(\alpha_0 + \langle X_i, \alpha \rangle_K)\, Y_i - \log\left(1 + \exp\left\{\alpha_0 + \langle X_i, \alpha \rangle_K\right\}\right)\right] + \frac{1}{2}\log|G_\tau| - (p+2)\log\sigma - \frac{1}{2g\sigma^2}(b - b_0)^T G_\tau (b - b_0).$$

Making predictions

Bear in mind that in this case we are essentially approximating probabilities, so we need to transform the predicted values to a binary output in $\{0, 1\}$. According to the usual criterion of minimizing the misclassification probability, the Bayes optimal rule is recovered by predicting $\hat{Y} = 1$ when $\mathbb{P}(Y = 1|X) \geq 1/2$. Nevertheless, for a more general cost function one could consider other criteria that would lead to evaluating whether $\mathbb{P}(Y = 1|X) \geq \gamma$ for some threshold $\gamma \in [0, 1]$. With this last strategy in mind, the *summarize-then-predict* approach is analogous to the linear regression case:

$$\hat{Y}_i = \mathbb{I}\left(\left[1 + \exp\left\{-\hat{\alpha}_0 - \sum_{j=1}^{p} \hat{\beta}_j X_i(\hat{t}_j)\right\}\right]^{-1} \geq \gamma\right), \quad i = 1, \dots, n', \tag{2.10}$$

where $\mathbb{I}$ is the indicator function ($\mathbb{I}(P)$ is 1 if P is true and 0 otherwise). The hat estimates are obtained once again through a summary statistic T of the corresponding marginal posterior distributions. On the other hand, the prediction method that takes into account the entire posterior approximation (i.e. the *predict-then-summarize* approach) is somewhat different now, since there is the question of which response (the Bernoulli variables in (2.8) or the raw probabilities in (2.9)) to consider when averaging the posterior samples. Hence, there are primarily two possible outcomes:

Average approximate probability. Averaging the approximate probabilities $p_i^{(m)*} = \mathbb{P}(Y_i = 1|X_i, \theta^{(m)*})$ computed following (2.9) results in predictors $\hat{Y}_i = \mathbb{I}(\frac{1}{M}\sum_{m=1}^{M} p_i^{(m)*} \geq \gamma)$, for $i = 1, \dots, n'$.

Average approximate response. Averaging the approximate binary responses $Y_i^{(m)*}$ instead (see (2.6)) leads to predictions of the form $\hat{Y}_i = \mathbb{I}(\frac{1}{M}\sum_{m=1}^{M} Y_i^{(m)*} \geq \gamma)$, for $i = 1, \dots, n'$. In this case, each $Y_i^{(m)*}$ follows a Bernoulli distribution with parameter $p_i^{(m)*}$, and note that when $\gamma = 1/2$ this strategy is equivalent to predicting $\hat{Y}_i$ from the majority vote of all the $Y_i^{(m)*}$.

Lastly, the *variable selection* method is essentially the same as in the case of linear regression: we select p time instants from each trajectory based on a summary of the posterior distribution $\tau|\mathcal{D}_n$, and then feed the reduced data set to a finite-dimensional binary classification procedure.

3 Posterior consistency

In this section we will show how the Bayesian methodology in conjunction with the RKHS models provides a strong theoretical background for the prediction procedures derived from them. Firstly, let us briefly recall what we understand by posterior consistency. Note that to avoid confusion, throughout this section we will sometimes use bold letters to represent random variables. Consider a sample space $\mathcal{X}$ and $X_1, \dots, X_n$ an i.i.d. sample of the data X. Let us fix a prior distribution Π for random variables $\boldsymbol{\theta}$ on the parameter space Θ, that is, $\boldsymbol{\theta} \sim \Pi$, and let P_θ represent a sampling model (a distribution on $\mathcal{X}$ indexed by $\theta \in \Theta$) such that $X|\boldsymbol{\theta} \sim P_{\boldsymbol{\theta}}$. Furthermore, assume that the model is well-specified, i.e., there is a true value $\theta_0 \in \Theta$ such that $X \sim P_{\theta_0}$, and denote by P_0^∞ the joint probability measure of $(X_1, X_2, \dots)$ when θ_0 is the true value of the parameter.

Definition 3.1 (Ghosh and Ramamoorthi, 2003)**.** *We say that the posterior distribution is (strongly) consistent at θ_0 if for every neighborhood U of θ_0 it holds that*

$$\lim_{n\to\infty} \Pi(\boldsymbol{\theta} \in U \mid X_1, \dots, X_n) = 1 \quad P_0^\infty - a.s.$$

For a metric space (Θ, d), this is equivalent to

$$\lim_{n\to\infty} \Pi(d(\boldsymbol{\theta}, \theta_0) < \varepsilon \mid X_1, \dots, X_n) = 1 \quad P_0^\infty - a.s, \quad \textit{for all } \varepsilon > 0.$$

Note that the conditional probabilities are computed under the assumed joint distribution of $(\boldsymbol{\theta}, (X_1, X_2, \dots))$. Essentially, we are saying that the posterior concentrates around θ_0 for almost all sequences of data. Thus, if consistency holds, the effect of the prior gets diluted as more and more data is used for the inference.

Doob's theorem

It turns out that, under very general conditions, the posterior distribution is consistent at almost every value of θ_0 with respect to the measure induced by the prior (Doob, 1949). Let the sample space $\mathcal{X}$ and the parameter space Θ be complete separable metric spaces, endowed with their respective Borel sigma-algebras. Suppose that Π is a prior distribution on Θ, and for each $\theta \in \Theta$ let P_θ be a probability distribution on $\mathcal{X}$. Consider the model $\boldsymbol{\theta} \sim \Pi$ and $X_1, X_2, \dots | \boldsymbol{\theta} \sim P_{\boldsymbol{\theta}}$ i.i.d., where $\boldsymbol{\theta}$ is a random variable taking values in Θ. Observe that this induces a posterior distribution $\Pi(\boldsymbol{\theta}|X_1, \dots, X_n)$.

Theorem 3.1 (Doob's consistency theorem)**.** *If $\theta \mapsto P_\theta$ is one-to-one and $\theta \mapsto P_\theta(A)$ is measurable for all measurable sets $A \subseteq \mathcal{X}$, then the posterior distribution is consistent at Π-almost all values of Θ. That is, there exists $\Theta_* \subseteq \Theta$ such that $\Pi(\Theta_*) = 1$ and for all $\theta_0 \in \Theta_*$, if $X_1, X_2, \dots \sim P_{\theta_0}$ i.i.d., then for any neighborhood B of θ_0 we have*

$$\lim_{n\to\infty} \Pi(\boldsymbol{\theta} \in B \mid X_1, \dots, X_n) = 1 \quad P_0^\infty - a.s.$$

See chapter 7.4.1 of Schervish (1995), chapter 10.4 of Van der Vaart (1998), chapter 1.3 of Ghosh and Ramamoorthi (2003), or Miller (2018) for more details on this result. As a side note, in a nonparametric setting where the parameter of interest is a random function (e.g. a probability density), there is also a stronger consistency result by Schwartz (1965) which omits the Π-almost sure qualification under some more restrictive conditions. Moreover, there are some extensions of this result that deal with independent but not identically distributed data, such as Choi and Ramamoorthi (2008).

3.1 Consistency in our RKHS model

Now we study in detail whether Doob's theorem can be applied in our linear RKHS model, but the posterior consistency results we obtain hold *mutatis mutandis* in the case of the RKHS-based logistic model proposed here. In this section we will assume that the covariance function K of the underlying stochastic process X is strictly positive definite. In the linear case the sample space is $\mathcal{X} \times \mathcal{Y} = L^2[0,1] \times \mathbb{R}$, which is already a complete separable metric space. Consider for each $p \in \mathbb{N}$ the subset of the $(2p+2)$-Euclidean space

$$\Theta_p = \{(\beta, \tau, \alpha_0, \sigma^2) : \beta \in \mathbb{R}^p, \tau \in [0,1]^p, \alpha_0 \in \mathbb{R}, \sigma^2 \in \mathbb{R}_0^+\}.$$

Then, we can write our infinite-dimensional parameter space as

$$\Theta = \bigcup_{p=1}^{\infty} \Theta_p. \tag{3.1}$$

At this point we can follow an approach very similar to Miller (2023), in which posterior consistency is established in a mixture model with an infinite-dimensional parameter space that factorizes in the same way as (3.1). Note that given $\theta \in \Theta$ there is a unique $p = p(\theta)$ such that $\theta \in \Theta_p$. Considering that we will only be interested in small balls around the true value of the parameter, we can define a metric for $\theta, \theta' \in \Theta$ by

$$d(\theta, \theta') = \begin{cases} \min\{\|\theta - \theta'\|, 1\}, & \text{if } p(\theta) = p(\theta'), \\ 1, & \text{otherwise.} \end{cases}$$

Since each Θ_p is itself a complete separable metric space with the inherited Euclidean norm, Proposition A.1 in Miller (2023) ensures that (Θ, d) is a complete separable metric space. Further, we equip both $\mathcal{X} \times \mathcal{Y}$ and Θ with their respective Borel sigma-algebras. In terms of $\theta \in \Theta$, the data distribution can be expressed as $P_\theta(X, Y) = P_{\beta,\tau,\alpha_0,\sigma^2}(X, Y)$, which formally factorizes as $P_\theta(X, Y) = P(X)P_\theta(Y|X)$, where in turn $X \sim P(X)$ and, in our RKHS setting, $P_\theta(Y|X) \equiv \mathcal{N}(\alpha_0 + \sum_{j=1}^{p(\theta)} \beta_j X(t_j), \sigma^2)$. Now, let us suppose that $\theta \mapsto P_\theta(X, Y)(A)$ is measurable for all measurable sets $A \subseteq \mathcal{X} \times \mathcal{Y}$ (which holds under mild conditions; see Appendix B). Moreover, for convenience, we will denote the sequences $(X, Y)_{1:n} = (X_1, Y_1), \dots, (X_n, Y_n)$ and $(X, Y)_{1:\infty} = (X_1, Y_1), (X_2, Y_2), \dots$.

Now, the full hierarchical model under consideration is

$$\begin{aligned}
&\text{(no. of components)} && \mathcal{P} \sim \pi,\\
&\text{(component values)} && \boldsymbol{\beta} \mid \mathcal{P} = p \sim F_p,\\
&\text{(component times)} && \boldsymbol{\tau} \mid \mathcal{P} = p \sim G_p,\\
&\text{(intercept)} && \boldsymbol{\alpha}_0 \sim C,\\
&\text{(error variance)} && \boldsymbol{\sigma}^2 \sim D,\\
&\text{(observed data)} && (X,Y)_{1:n} \mid \boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\alpha_0}, \boldsymbol{\sigma^2} \sim P_{\boldsymbol{\beta},\boldsymbol{\tau},\boldsymbol{\alpha_0},\boldsymbol{\sigma^2}}(X,Y) \quad \text{i.i.d.,}
\end{aligned} \tag{3.2}$$

where π, F_p, G_p, C and D are probability measures on $\mathbb{N}$, $\mathbb{R}^p$, $[0,1]^p$, $\mathbb{R}$ and $\mathbb{R}_0^+$, respectively. Note that since $P_\theta(X,Y)$ is invariant under permutations of the component labels β and τ, we can only show consistency up to one such permutation. To that effect, and mirroring the strategy of Miller (2023), let S_p denote the set of permutations of $\{1,\dots,p\}$, and for $\nu \in S_p$ and $\theta \in \Theta_p$, denote by $\theta[\nu]$ the result of applying the permutation ν to the component labels of θ. That is, if $\theta = (\beta_1,\dots,\beta_p,t_1,\dots,t_p,\alpha_0,\sigma^2)$, then $\theta[\nu] = (\beta_{\nu_1},\dots,\beta_{\nu_p},t_{\nu_1},\dots,t_{\nu_p},\alpha_0,\sigma^2)$. Now, for $\theta_0 \in \Theta_p$ and $\varepsilon > 0$ define the neighborhood $\tilde{B}(\theta_0,\varepsilon) = \bigcup_{\nu\in S_p}\{\theta \in \Theta : d(\theta,\theta_0[\nu]) < \varepsilon\}$, which is the set of all parameters that are within ε of some permutation of (the component labels of) θ_0. Lastly, define the random variable $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\alpha_0}, \boldsymbol{\sigma^2})$, which takes values in Θ, and denote by Π the prior distribution on $\boldsymbol{\theta}$ implied by the model in (3.2). In order for identifiability to hold, we need to place some restrictions on the prior:

Condition 3.1 (Identifiability constraints)**.** *Under the model in* (3.2)*, for all* $p \in \mathbb{N}$*:*

1. $\Pi(t_i = t_j|\mathcal{P} = p) = 0$ *for all* $1 \le i < j \le p$*.*
2. *There exists* $\delta > 0$ *such that* $\Pi(|\beta_j| < \delta|\mathcal{P} = p) = 0$ *for all* $1 \le j \le p$*.*

Both assumptions can be interpreted as a way of pursuing parsimony in the model, aiming for as few components as possible. In practical and computational terms, we can think of δ as the *machine precision number*, so that virtually all continuous prior distributions satisfy the associated condition. With this setup in mind, we are now ready to state our main theorem:

Theorem 3.2. *Suppose that Condition 3.1 holds. Then there exists* $\Theta_* \subseteq \Theta$ *such that* $\Pi(\boldsymbol{\theta} \in \Theta_*) = 1$ *and for all* $\theta_0 \in \Theta_*$*, if* $(X,Y)_{1:\infty} \sim P_{\theta_0}(X,Y)$ *i.i.d., then for all* $\varepsilon > 0$

$$\lim_{n\to\infty} \Pi(\boldsymbol{\theta} \in \tilde{B}(\theta_0,\varepsilon) \mid (X,Y)_{1:n}) = 1 \quad P_0^\infty(X,Y) - a.s.$$

and

$$\lim_{n\to\infty} \Pi(\mathcal{P} = p(\theta_0) \mid (X,Y)_{1:n}) = 1 \quad P_0^\infty(X,Y) - a.s.$$

All in all, this result guarantees consistency for almost every parameter in the support of the prior distribution. In addition, the second conclusion is of certain relevance in itself because the estimation of the number of components in mixture-like models is a hard problem in general (see Miller and Harrison, 2018, and references therein). However, even though we can choose the prior in such a way that $\operatorname{supp}(\Pi) = \Theta$, in principle there is no assurance that the Π-null set in which consistency may fail will not be a large set. In fact, when the parameter space is infinite-dimensional there are examples of large inconsistency sets, even for reasonably chosen prior distributions (e.g. Diaconis and Freedman, 1986). Nonetheless, when the parameter space is a countable union of disjoint finite-dimensional sets, we can further refine our almost sure statement. First, note that there is a natural extension of the Lebesgue measure to our parameter space Θ: just consider the genuine Lebesgue measure λ_p on Θ_p, and for all $B \subseteq \Theta$ measurable define $\lambda(B) = \sum_{p=1}^{\infty} \lambda_p(\Theta_p \cap B)$. Then, if we choose a prior distribution with respect to which this measure is absolutely continuous, the consistency set Θ_* in Theorem 3.2 will satisfy $\lambda(\Theta \setminus \Theta_*) = 0$. A similar approach is considered in Nobile (1994) and Miller (2023) to establish "Lebesgue"-almost sure consistency in finite mixture models with a prior on the number of components. In our case, the requirement of absolute continuity can be relaxed so that sets with nonzero Lebesgue measure have nonzero prior probability *for some permutation of the component labels*. We also need to impose the somewhat technical condition that the prior assign positive mass to all $p \in \mathbb{N}$.

Condition 3.2 (Absolute continuity)**.** *Under the model in* (3.2)*, for all* $p \in \mathbb{N}$*:*

1. $\Pi(\mathcal{P} = p) > 0$.
2. $\sum_{\nu\in S_p} \Pi(\boldsymbol{\theta}[\nu] \in B|\mathcal{P} = p) = 0$ *implies* $\lambda_p(B) = 0$*, for all* $B \subseteq \Theta_p$ *measurable.*

The second condition is met, for example, if $\boldsymbol{\theta}|p$ has a density with respect to Lebesgue measure that is invariant to permutations of the component labels and positive on all of Θ_p. Finally, we get the announced result:

Theorem 3.3. *Assume Condition 3.1 and Condition 3.2 hold. Then the conclusion of Theorem 3.2 remains valid with* $\lambda(\Theta \setminus \Theta_*) = 0$*.*

Additionally, the proof of Theorem 3.2 can be easily tweaked to guarantee consistency when the number of components is fixed beforehand (and thus the parameter space is finite-dimensional). Indeed, in this case Doob's theorem applies directly under the sole condition that the times be distinct with prior probability one. The coefficients β_j do not cause a problem for identifiability now, since the dimension of every parameter is the same.

Theorem 3.4. *Assume the model (3.2) in which the value of p is fixed, and hence Θ_p is the finite-dimensional parameter space. Suppose that Condition 3.1-1 holds. Then if θ_0 is the true value of the parameter, the posterior is consistent at θ_0 with Π-probability one. If moreover Condition 3.2-2 holds, then the inconsistency set has Lebesgue measure zero.*

Note that by allowing β_j to be zero we can sometimes circumvent the fact that the true value of the parameter might not have exactly p components, as long as p is larger than the true value $p(\theta_0)$. Indeed, if $\theta_0 \in \Theta$ with $p(\theta_0) < p$ and $(X, Y)_{1:\infty} \sim P_{\theta_0}(X, Y)$ i.i.d., then we can find $\theta_1 \in \Theta_p$ such that P_{θ_0}(X, Y) = $P_{\theta_1}(X, Y)$ and the result holds: just set $p - p(\theta_0)$ components of θ_1 to zero and keep the rest of the values as in θ_0.

3.2 A sketch of the proofs

We now present an overview of the proof of the consistency results. The general strategy will be to apply Doob's theorem to a subset of the parameter space Θ, where permutations are suitably taken into account and full identifiability holds, and then we will extend the results to the whole parameter space. As we will see shortly, save for an eventual permutation, identifiability of the map $\theta \mapsto P_\theta(X, Y)$ is obtained when the covariance function K of the underlying stochastic process $X = X(t)$ is non-degenerate.

Reduced parameter space. Consider the spaces $[0,1]^p_{\text{ord}} = \{(t_1, \dots, t_p) \in [0,1]^p : t_1 < \cdots < t_p\}$ and $\mathbb{R}_\delta = (-\infty, -\delta] \cup [\delta, +\infty)$, with δ the fixed value given in Condition 3.1-2, and define a new finite-dimensional parameter space $\tilde{\Theta}_p = \mathbb{R}^p_\delta \times [0,1]^p_{\text{ord}} \times \mathbb{R} \times \mathbb{R}^+_0$. Now consider $\tilde{\Theta} = \bigcup_{p\geq 1} \tilde{\Theta}_p$, and note that the sets $\tilde{\Theta}_1, \tilde{\Theta}_2, \dots$ are still disjoint. Then, by the same argument as above, we can conclude that $\tilde{\Theta}$ is a complete separable metric space under the metric d. We will henceforth say that a parameter in $\tilde{\Theta}$ is "ordered".

Transformation into ordered form. For $\theta \in \Theta_p$, define $T(\theta) = \theta[\nu]$ if there is $\nu \in S_p$ such that $\theta[\nu] \in \tilde{\Theta}_p$; otherwise set $T(\theta) = \theta$. Note that we can only transform θ to be in $\tilde{\Theta}_p$ if the times t_j are all distinct and the coefficients β_j satisfy $|\beta_j| \geq \delta$. But since the prior distribution assigns probability one to both events by Condition 3.1, we have $\Pi(T(\boldsymbol{\theta}) \in \tilde{\Theta}) = 1$. It will be useful later to observe that for any set $B \subseteq \tilde{\Theta}_p$, if we denote $B[\nu] = \{\theta[\nu] : \theta \in B\}$, we have

$$\bigcup_{\nu \in S_p} B[\nu] = T^{-1}(B). \tag{3.3}$$

Collapsed model. Let $\tilde{Q}$ denote the distribution of $T(\boldsymbol{\theta})$ restricted to $\tilde{\Theta}$, and note the equivalence $P_{T(\theta)}(X, Y) = P_\theta(X, Y)$. Then the following model holds on the reduced space $\tilde{\Theta}$:

$$\begin{aligned} &T(\boldsymbol{\theta}) \sim \tilde{Q}, \\ &(X, Y)_{1:n} \mid T(\boldsymbol{\theta}) \sim P_{T(\boldsymbol{\theta})}(X, Y) \quad \text{i.i.d.} \end{aligned} \tag{3.4}$$

Verifying conditions. We will now show that the conditions of Doob's theorem hold on $\tilde{\Theta}$. First, since $\theta \mapsto P_\theta(X, Y)(A)$ is measurable on Θ, it is also measurable on $\tilde{\Theta}$ for all sets $A \subseteq \mathcal{X} \times \mathcal{Y}$ measurable. For the identifiability part, suppose by contradiction that there are $\theta, \theta' \in \tilde{\Theta}$ such that $\theta \neq \theta'$ and $P_\theta(X, Y) = P_{\theta'}(X, Y)$. Then necessarily $\sum_{j=1}^{p(\theta)} \beta_j X(t_j) = \sum_{j=1}^{p(\theta')} \beta'_j X(t'_j)$, which reordering the terms implies $\sum_{j=1}^{p(\theta)+p(\theta')} \beta^*_j X(t^*_j) = 0$ for some $\beta^*_j \in \mathbb{R}$ and $t^*_j \in [0,1]$. Now observe that all the β^*_j must vanish, since the covariance function of the process X is strictly positive definite. But that can only happen if $\theta' = \theta[\nu]$

for some $\nu \in S_p$, where $p = p(\theta) = p(\theta')$ (because $\beta_j \neq 0$ and $t_i \neq t_j$ on $\tilde{\Theta}_p$). However, $t_1 < \cdots < t_p$ and $t_1' < \cdots < t_p'$, so ν must be the identity permutation, that is, $\theta' = \theta$, contradicting the initial assumption.

Applying Doob's theorem. Next we analyze the conclusions of Doob's theorem applied to the collapsed model (3.4): there exists $\tilde{\Theta}_* \subseteq \tilde{\Theta}$ with $\Pi(T(\boldsymbol{\theta}) \in \tilde{\Theta}_*) = 1$ such that, if $T(\theta_0) \in \tilde{\Theta}_*$ and the data verifies $(X,Y)_{1:\infty} \sim P_{T(\theta_0)}(X,Y)$ i.i.d., then for any neighborhood $B \subseteq \tilde{\Theta}$ of $T(\theta_0)$ it holds that

$$\Pi(T(\boldsymbol{\theta}) \in B \mid (X,Y)_{1:n}) \xrightarrow{n\to\infty} 1 \quad P^{\infty}_{T(\theta_0)} - \text{a.s.} \tag{3.5}$$

Now define Θ_* to be the set of all parameters in Θ that can be obtained by permuting the component labels of a parameter in $\tilde{\Theta}_*$, i.e., $\Theta_* = \bigcup_{p=1}^{\infty} \bigcup_{\nu \in S_p} (\tilde{\Theta}_* \cap \tilde{\Theta}_p)[\nu]$. Then, by (3.3) we have

$$\Pi(\boldsymbol{\theta} \in \Theta_*) = \Pi\left(T(\boldsymbol{\theta}) \in \bigcup_{p=1}^{\infty} (\tilde{\Theta}_* \cap \tilde{\Theta}_p) \right) = \Pi(T(\boldsymbol{\theta}) \in \tilde{\Theta}_*) = 1.$$

Extending the result to Θ. Now, let $\theta_0 \in \Theta_*$, define $p_0 = p(\theta_0)$ and $S_0 = S_{p_0}$, and suppose that $(X,Y)_{1:\infty} \sim P_{\theta_0}(X,Y)$ i.i.d. Fix $\varepsilon \in (0,1)$ and consider the set B of all ordered parameters that are within ε of the ordered version of θ_0, i.e.,

$$B = \left\{ \theta \in \tilde{\Theta} : d(\theta, T(\theta_0)) < \varepsilon \right\}. \tag{3.6}$$

Observe that, since $\varepsilon < 1$, we have $B \subseteq \tilde{\Theta}_{p_0}$ by definition of d. Moreover, $\bigcup_{\nu\in S_0} B[\nu] \subseteq \tilde{B}(\theta_0, \varepsilon)$. Then, again by (3.3), we can write

$$\begin{aligned} \Pi(\boldsymbol{\theta} \in \tilde{B}(\theta_0,\varepsilon) \mid (X,Y)_{1:n}) &\geq \Pi(\boldsymbol{\theta} \in \bigcup_{\nu\in S_0} B[\nu] \mid (X,Y)_{1:n}) \\ &= \Pi(T(\boldsymbol{\theta}) \in B \mid (X,Y)_{1:n}). \end{aligned} \tag{3.7}$$

Now, $T(\theta_0) \in \tilde{\Theta}_*$ because $\theta_0 \in \Theta_*$, and in that case we know that the collapsed model is consistent at $T(\theta_0)$. Note that we also have $(X,Y)_{1:\infty} \sim P_{T(\theta_0)}(X,Y)$ i.i.d. (since $P_{\theta_0} = P_{T(\theta_0)}$), and the set B in (3.6) is a neighborhood of $T(\theta_0)$ in $\tilde{\Theta}$. Then, by (3.5) we can conclude that, $\Pi(T(\boldsymbol{\theta}) \in B | (X,Y)_{1:n}) \xrightarrow{n\to\infty} 1$, $P^{\infty}_{\theta_0}(X,Y) -$ a.s., and this fact together with (3.7) proves consistency for θ_0 in the original model (3.2). Lastly, since $\varepsilon < 1$ implies $\tilde{B}(\theta_0,\varepsilon) \subseteq \Theta_{p_0}$, we have also proved the second assertion of our theorem:

$$\begin{aligned} \Pi(\mathcal{P} = p_0 \mid (X,Y)_{1:n}) &= \Pi(\boldsymbol{\theta} \in \Theta_{p_0} \mid (X,Y)_{1:n}) \\ &\geq \Pi(\boldsymbol{\theta} \in \tilde{B}(\theta_0,\varepsilon) \mid (X,Y)_{1:n}) \\ &\xrightarrow[n\to\infty]{} 1 \quad P^{\infty}_{\theta_0}(X,Y) - \text{a.s.} \end{aligned}$$

□

The proof of Theorem 3.3 is now straightforward. Define Θ_* as in the proof of Theorem 3.2, and observe that $0 = \Pi(\Theta \setminus \Theta_*) = \sum_{p=1}^{\infty} \Pi(\Theta_p \setminus \Theta_* | \mathcal{P} = p)\Pi(\mathcal{P} = p)$, since $\Pi(\Theta_*) = 1$. Now, since $\Pi(\mathcal{P} = p) > 0$ for all $p \in \mathbb{N}$ by Condition 3.2-1, then necessarily $\Pi(\Theta_p \setminus \Theta_* | \mathcal{P} = p) = 0$ for all $p \in \mathbb{N}$. Then, for $\nu \in S_p$, let μ_p^{ν} be the distribution of $\boldsymbol{\theta}[\nu] | \mathcal{P} = p$ under the model. Given that $(\Theta_p \setminus \Theta_*)[\nu] = \Theta_p \setminus \Theta_*$ by definition of Θ_*, we have that for all $\nu \in S_p$,

$$\mu_p^{\nu}(\Theta_p \setminus \Theta_*) = \mu_p^{\text{id}}(\Theta_p \setminus \Theta_*) = \Pi(\Theta_p \setminus \Theta_* \mid \mathcal{P} = p) = 0. \tag{3.8}$$

Lastly, note that Condition 3.2-2 means that $\lambda_p \ll \sum_{\nu \in S_p} \mu_p^{\nu}$, where $\ll$ denotes absolute continuity, and this together with (3.8) implies that $\lambda_p(\Theta_p \setminus \Theta_*) = 0$. But this is valid for all $p \in \mathbb{N}$, so we conclude that $\lambda(\Theta \setminus \Theta_*) = \sum_{p=1}^{\infty} \lambda_p(\Theta_p \setminus \Theta_*) = 0$. □

As a final comment, it is worth reiterating that in this theoretical aspect of our work we have closely followed the techniques recently developed in Miller (2023), where the author provides a simplification of the work by Nobile (1994) in studying posterior consistency in finite-dimensional mixture models with a prior on the number of components. While the methods are quite similar, we have succeeded in extending this theory to a fundamentally different situation, namely functional (i.e. infinite-dimensional) regression models, which, thanks to the RKHS formulation, share the key properties that allow for a treatment parallel to the finite-dimensional mixture case.

4 Experimental results

In this section we present the results of the experiments carried out to test the performance of our models in different scenarios, both simulated and with real data. More details on them (such as simulation parameters in data sets, hyperparameters or implementation decisions), as well as additional experiments, figures and tables are available on Appendix C, while the code itself is available at `https://github.com/antcc/rk-bfr` (see also Appendix D). For the purposes of computation we consider the point statistics *mean*, *median* and *mode* for our summarize-then-predict approach (see (2.5) and (2.10)). As a result, in linear regression we have 4 prediction methods (one for each statistic and one for the predict-then-summarize approach) and 3 variable selection methods (one for each statistic), while in logistic regression there are similarly 5 prediction methods and 3 variable selection procedures. We will refer to the predict-then-summarize methods as *posterior_mean* (with the additional *posterior_vote* in logistic regression). In each case, after variable selection is performed, we use an l^2-penalized multiple linear/logistic regression method to generate the corresponding predictions. All in all, we are looking at 7 (8 in the case of logistic regression) prediction methods, and although all of them are derived from a single MCMC run, we will treat them as separate in the experimentation.

For the experimental setting we take $n = 150$ training samples and $n' = 100$ testing samples on an equispaced grid of $N = 100$ points on $[0, 1]$ for the simulated data sets, and we do a 66%/33% train/test split on the real data sets. We then perform 5-fold cross validation (CV) on the training set to select the best values of p and η for each model (the other hyperparameters in (2.4) are fixed for simplicity), and after refitting the best model in each case on the whole training set, we evaluate it and measure the predictive performance on the test set. Since the initial experiments carried out indicate that low values of p provide sufficient flexibility in most scenarios, we look for p in the set $\{1, 2, \ldots, 10\}$, while the possible values of η are $\{10^{-4}, 10^{-3}, \ldots, 10^2\}$. Lastly, we independently repeat the whole process 10 times (each with a different train/test configuration) to account for the stochasticity in the prediction procedure, and average the results across these executions. The metrics used to evaluate the performance of the models are the Root Mean Square Error (RMSE) for linear regression and the accuracy (rate of correctly predicted samples) for logistic regression.

The Python library used to perform the MCMC approximation is *emcee* (Foreman-Mackey et al., 2013), and thus our methods inherit that name as a prefix in their denomination (see Appendices A.3 and A.4 for more information). Because of execution time constraints, the hyperparameters of the MCMC method are not part of the CV process, and are selected manually based on an initial set of experiments, as well as recommendations from the original article. Moreover, a small adjustment is needed to mitigate the well-known *label switching* phenomenon that occurs in MCMC approximations of mixture-like models (e.g. Stephens, 2000); see Appendix A.1 for details.

Data sets

We consider a set of functional regressors common to linear and logistic regression problems. They are four Gaussian processes (GPs), each with a different covariance function. In particular, we consider a Brownian motion, a fractional Brownian motion, an Ornstein-Uhlenbeck process, and a GP with a Gaussian kernel. Also, when applicable, we fix a variance $\sigma^2 = 0.5$ for the error terms ε.

Linear regression data sets. We employ two different types of simulated data sets, all with a common value of $\alpha_0 = 5$.

- A finite-dimensional RKHS response with three components for each of the four GP regressors mentioned above: $Y = 5 - 5X(0.1) + X(0.4) + 10X(0.8) + \varepsilon$.
- A "component-less" response generated by an L^2-model with a smooth underlying coefficient function, namely $\beta(t) = \log(1 + 4t)$, again for the same four GPs.

As for the real data sets, we use the (twice-differentiated) Tecator data set (Borggaard and Thodberg, 1992) to predict fat content based on near-infrared absorbance curves of 193 meat samples, as well as what we call the Moisture (Kalivas, 1997) and Sugar (Bro, 1999) data sets. The first consists of near-infrared spectra of 100 wheat samples and the objective is to predict the samples' moisture content, whereas the second contains 268 samples of sugar fluorescence data in order to predict ash content. The three data sets are measured on a grid of 100, 101 and 115 equispaced points on $[0, 1]$, respectively.

Logistic regression data sets. Again we consider two different types of simulated data sets, with a common value of $\alpha_0 = -0.5$. In this case we randomly permute 10% of the labels to introduce some noise in the simulations.

- Four logistic finite-dimensional RKHS responses with the same functional parameter as in the linear regression case (one for each GP). Specifically,

$$\mathbb{P}(Y = 1 \mid X) = \frac{1}{1 + \exp\{0.5 + 5X(0.1) - X(0.4) - 10X(0.8)\}}.$$

- Four logistic responses following an L^2-model with the same coefficient function as in the linear regression case, i.e., $\beta(t) = \log(1 + 4t)$.

Additionally, we use three real data sets well known in the literature. The first one is a subset of the Medflies data set (Carey et al., 1998), consisting on samples of the number of eggs laid daily by 534 flies over 30 days, to predict whether their longevity is high or low. The second one is the Berkeley Growth Study data set (Tuddenham and Snyder, 1954), which records the height of 54 girls and 39 boys over 31 different points in their lives. Finally, we selected a subset of the Phoneme data set (Hastie et al., 1995), based on 200 digitized speech frames over 128 equispaced points to predict the phonemes “aa” and “ao”.

Comparison algorithms

We have included a fairly comprehensive suite of comparison algorithms, chosen among the most common methods used in machine learning and FDA, and following a standard choice of implementation and hyperparameters. There are purely functional methods (such as the usual L^2 regression that follows models (1.1) and (1.2)), finite-dimensional models that work on the discretized data (e.g. penalized finite-dimensional regression), and variable selection/dimension reduction procedures (like PCA or PLS). The main parameters of all these algorithms are selected by cross-validation, using the same 5 folds as our proposed models so that the comparisons are fair. A detailed account of these algorithms is available in Appendix C.1.

Results display

We have adopted a visual approach to presenting the experimentation results, using colored graphs instead of tables to help visualize them, since we felt that this was a better way of summarizing a large empirical study such as the one we have carried out. In each case, the mean and standard deviation of the score obtained across the 10 random runs is shown, depicting our models in orange and the comparison algorithms in blue. We also show the global mean of all the comparison algorithms with a dashed vertical line, excluding extreme negative results from this mean to avoid distortion. Moreover, we separate complete prediction algorithms from two-stage methods, the latter being the ones that perform variable selection or dimension reduction prior to a multiple linear/logistic regression method.

4.1 Functional linear regression

Simulated data sets

In Figure 1 we see the results for the four GP regressors considered in the RKHS case. This is the most favorable case for us, as the underlying model coincides with our assumed model. Indeed, we can see that in most cases our algorithms are the ones with lower RMSE, save for a few exceptions, notably the Gaussian kernel. A subsequent analysis showed that this particular data set is especially sensitive to the value of the hyperparameter η in our prior distribution, so a more customized approach would be needed to obtain better results.

In Figure 2 we see the results for the case with an underlying L^2-model, which would be our most direct competitor. In this case the outcome is satisfactory, since for the most part our models are on a par with the rest, even beating other methods that were designed with the L^2-model in mind. Moreover, whenever one of our models has a higher RMSE, the difference is pretty small in comparison. Note that some of our Bayesian models have a higher standard deviation, partly because there is an intrinsic randomness in the methods, and it can be the cause of the occasional worse performance. In relation to this, we observe that the methods that use the mean as a summary statistic tend to perform much worse. Since the mean is very sensitive to outliers, if at some point a MCMC chain randomly deviates from the rest, the average of the posterior samples is greatly impacted.

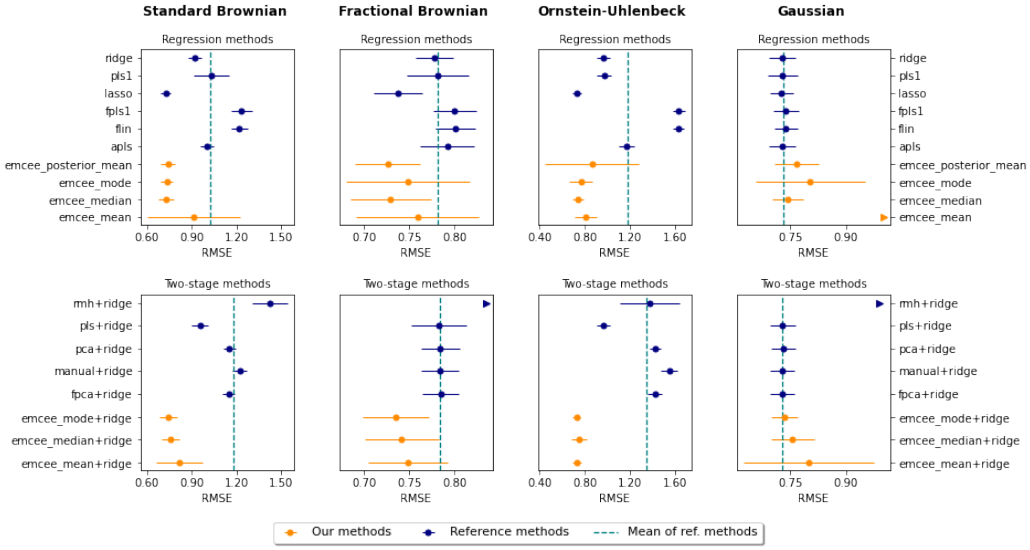

Figure 1: Mean and standard error of RMSE of predictors (lower is better) for 10 runs with GP regressors, one in each column, that obey an underlying linear RKHS model. The first row are direct methods and the second are dimensionality reduction methods.

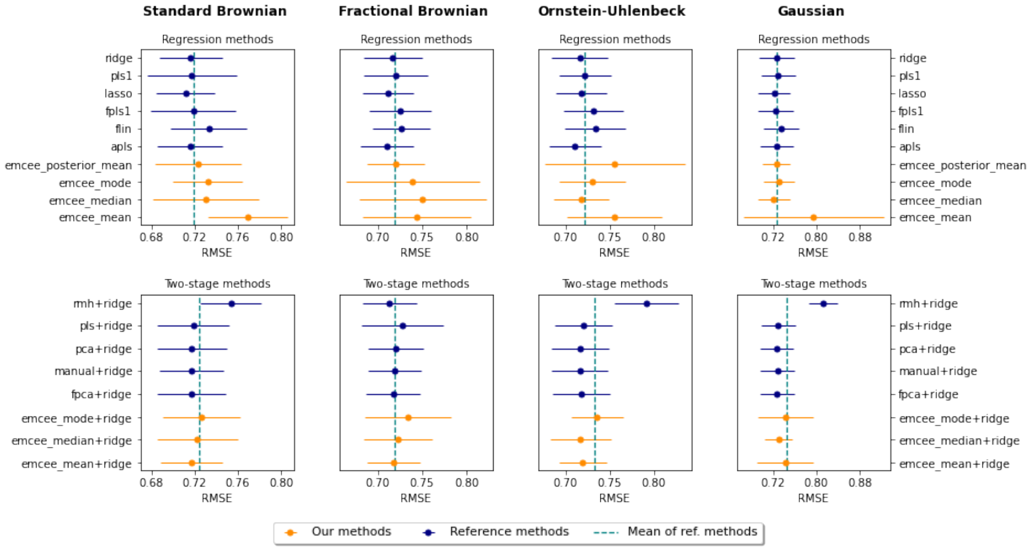

Figure 2: Mean and standard error of RMSE of predictors (lower is better) for 10 runs with GP regressors, one in each column, that obey an underlying linear L^2-model. The first row are direct methods and the second are dimensionality reduction methods.

Real data

Figure 3 shows the results for the real data sets, where we can see that there is sometimes a substantial difference in performance between some of our methods and the reference algorithms. However, our predict-then-summarize approach (*emcee_posterior_mean*) seems to work quite well, always scoring near the mean RMSE of all the comparison algorithms. Moreover, our two-stage methods seem to outperform the summarize-then-predict methods in the Moisture and Sugar data sets, scoring again very close to the mean of the reference models. We have to bear in mind that real data is more complex and noisy than simulated data, and it is possible that after a suitable pre-preprocessing we would obtain better results. However, our goal was to perform a general comparison without focusing too much on the specifics of any particular data set.

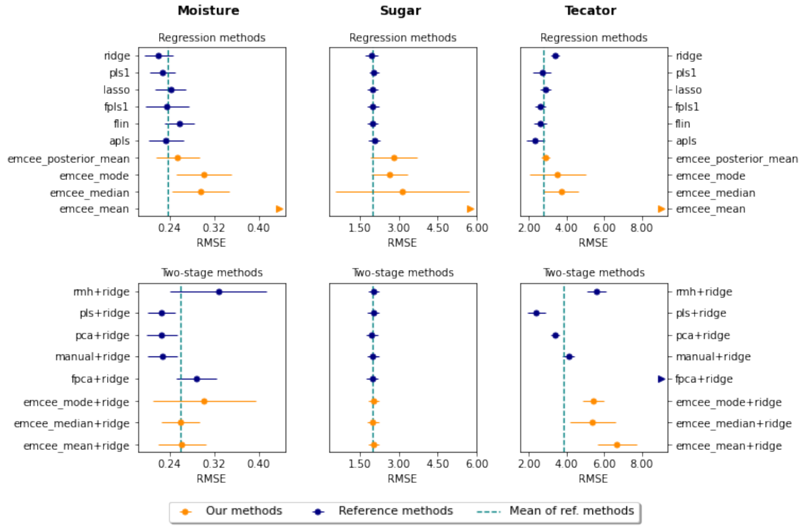

Figure 3: Mean and standard error of RMSE of predictors (lower is better) for 10 runs with real data sets, one in each column. The first row are direct methods and the second are dimensionality reduction methods.

4.2 Functional logistic regression

Simulated data sets

In Figure 4 we see the results for the GP regressors in the logistic RKHS case. Our models perform fairly well in this advantageous case, although they are not always better than the comparison methods. However, in most cases the differences observed account for only one or two misclassified samples.

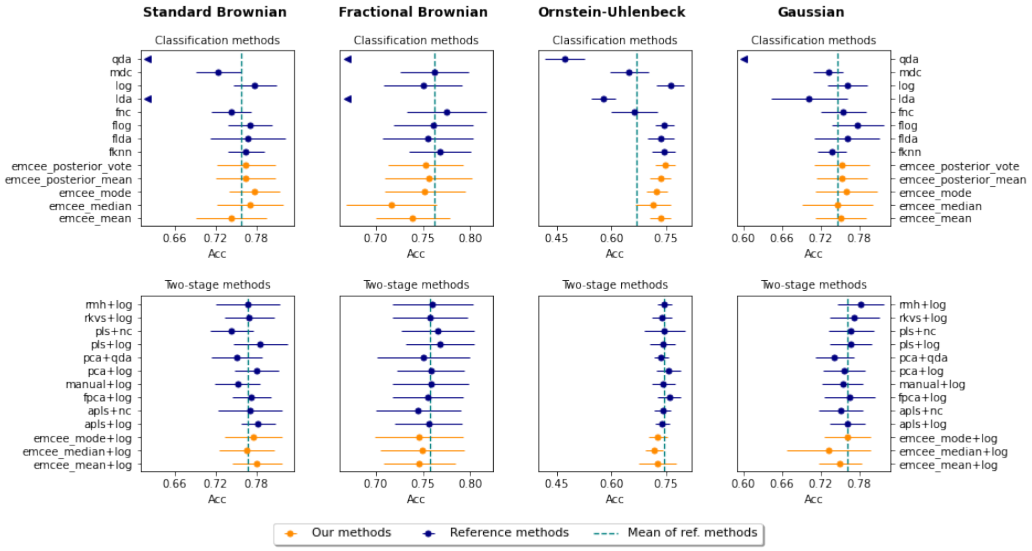

Figure 4: Mean and standard error of accuracy of classifiers (higher is better) for 10 runs with GP regressors, one in each column, that obey an underlying logistic RKHS model. The first row are direct methods and the second are dimensionality reduction methods.

Moreover, Figure 5 shows that the results in the L^2 case are again promising, since our models score consistently on or above the mean of the reference models, and in many instances surpass most of them. The predict-then-summarize approaches (*emcee_posterior_mean* and *emcee_posterior_vote*) are particularly good in this case, and in general have low standard errors. Moreover, the overall accuracy of all methods is poor (below 60%), so this is indeed a difficult problem in which even small increases in accuracy are relevant.

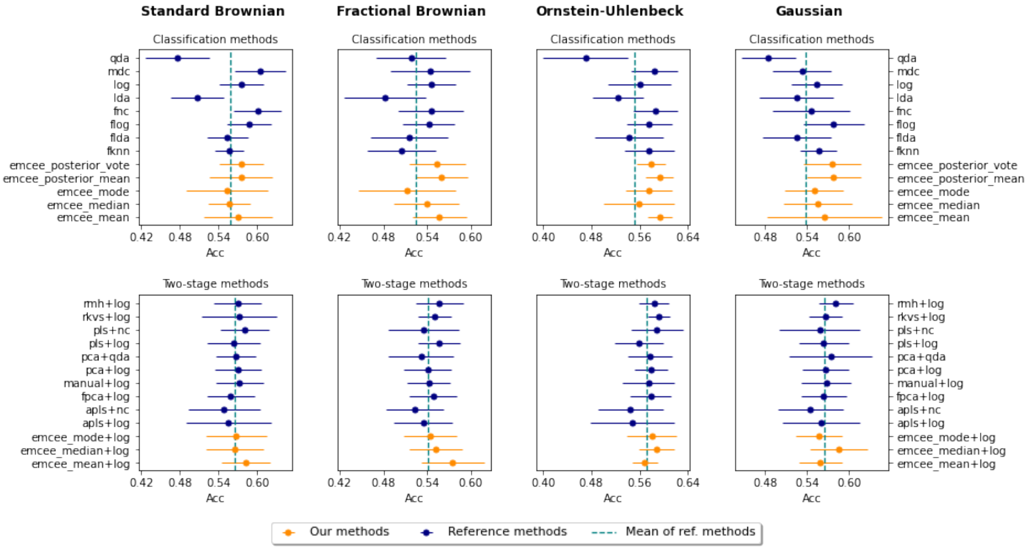

Figure 5: Mean and standard error of accuracy of classifiers (higher is better) for 10 runs with GP regressors, one in each column, that obey an underlying logistic L^2-model. The first row are direct methods and the second are dimensionality reduction methods.

Real data

As for the real data sets, in Figure 6 we see positive results in general, obtaining in most cases accuracies well above the mean of the reference models, and sometimes above most of them. In particular, the predict-then-summarize methods tend to have a good performance and achieve a lower standard error across executions, which is a trend that we also saw in the simulated data sets. However, as we have been seeing almost invariably, the models that use *emcee_mean* are the exception: in all these data sets they perform steadily worse than the rest of our Bayesian models. Moreover, we can appreciate that for the most part our logistic models perform better in comparison with our linear regression models.

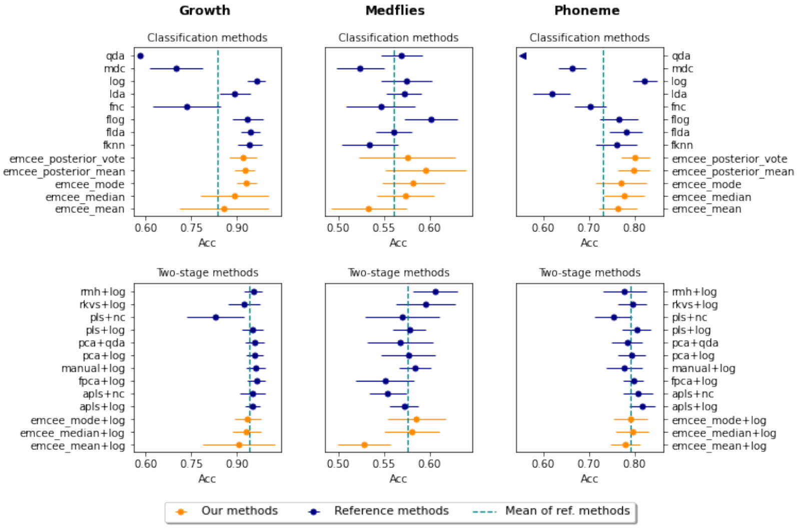

Figure 6: Mean and standard error of accuracy of classifiers (higher is better) for 10 runs with real data sets, one in each column. The first row are direct methods and the second are dimensionality reduction methods.

5 Conclusion

In this work we have introduced a natural and computationally feasible way of integrating Bayesian inference into functional regression models, by means of a RKHS approach that simplifies the usually hard task of setting a prior distribution on a functional space. Moreover, the RKHS framework motivates models based on a linear combination of marginals of the underlying stochastic process, which in a way gives a theoretical background to these popular models, but still retains the functional point of view. Our finite-dimensional approximation has the advantage of working with simpler functional parameters, thus increasing the interpretability and ease of implementation. In addition, it is worth pointing out that our approach works especially well in the logistic case, bypassing the difficulties associated with maximum likelihood techniques and providing a tractable alternative to the more studied methods in the literature.

We have also proved a posterior consistency result that ensures the coherence and correctness of the Bayesian methods we developed. These kinds of results have in other contexts more intricate and restrictive conditions to arrive at essentially the same conclusions as we did, but again the introduction of RKHS's is the key point to greatly simplifying them. In the end, we can regard our derivations as yet another way of applying the abstract posterior consistency theorem of Doob to a concrete situation, giving a positive answer to a problem which was not originally envisioned by this result. Lastly, we have presented numerical evidence that supports the proposed RKHS-based Bayesian methodology and its predictive performance with simulated and real data. This practical side of our work goes to show that the prediction methods we constructed from the posterior distribution are competitive against several non-cherry-picked frequentist alternatives, while still remaining relatively simple, interpretable and viable implementation-wise.

Acknowledgments

This research was partially supported by grants PID2019-109387GB-I00 and PRE2020-095147 of the Spanish Ministry of Science and Innovation (MCIN), co-financed by the European Social Fund (ESF). J. R. Berrendero and A. Cuevas acknowledge financial support from grant CEX2019-000904-S funded by MCIN/AEI/ 10.13039/501100011033. The authors also wish to acknowledge the computational resources provided by the Centro de Computación Científica-Universidad Autónoma de Madrid (CCC-UAM).

References

Abdi, H. (2010). "Partial least squares regression and projection on latent structure regression (PLS Regression)." *WIREs Computational Statistics*, 2(1): 97–106. doi: https://doi.org/10.1002/wics.51. 27

Abraham, C. (2023). "An informative prior distribution on functions with application to functional regression." *Statistica Neerlandica*, Early View: 1–17. doi: https://doi.org/10.1111/stan.12322. 3

Abraham, C. and Grollemund, P.-M. (2020). "Posterior concentration for a misspecified Bayesian regression model with functional covariates." *Journal of Statistical Planning and Inference*, 208: 58–65. doi: https://doi.org/10.1016/j.jspi.2020.01.008. 2

Aguilera, A. M. and Aguilera-Morillo, M. (2013). "Comparative study of different B-spline approaches for functional data." *Mathematical and Computer Modelling*, 58(7-8): 1568–1579. doi: https://doi.org/10.1016/j.mcm.2013.04.007. 2

Aguilera, A. M., Escabias, M., Preda, C., and Saporta, G. (2010). "Using basis expansions for estimating functional PLS regression: applications with chemometric data." *Chemometrics and Intelligent Laboratory Systems*, 104(2): 289–305. doi: https://doi.org/10.1016/j.chemolab.2010.09.007. 27

Albert, A. and Anderson, J. A. (1984). "On the existence of maximum likelihood estimates in logistic regression models." *Biometrika*, 71(1): 1–10. doi: https://doi.org/10.1093/biomet/71.1.1. 7

Amewou-Atisso, M., Ghosal, S., Ghosh, J. K., and Ramamoorthi, R. V. (2003). "Posterior consistency for semi-parametric regression problems." *Bernoulli*, 9(2): 291–312. doi: https://doi.org/10.3150/bj/1068128979. 2

Baragatti, M. and Pommeret, D. (2012). "A study of variable selection using g-prior distribution with ridge parameter." *Computational Statistics & Data Analysis*, 56(6): 1920–1934. doi: https://doi.org/10.1016/j.csda.2011.11.017. 6

Berlinet, A. and Thomas-Agnan, C. (2004). *Reproducing kernel Hilbert spaces in probability and statistics*. Springer. doi: https://doi.org/10.1007/978-1-4419-9096-9. 4

Berrendero, J. R., Bueno-Larraz, B., and Cuevas, A. (2019). "An RKHS model for variable selection in functional linear regression." *Journal of Multivariate Analysis*, 170: 25–45. doi: https://doi.org/10.1016/j.jmva.2018.04.008. 3, 5

— (2020a). "On Mahalanobis Distance in Functional Settings." *Journal of Machine Learning Research*, 21(9): 1–33. url: http://jmlr.org/papers/v21/18-156.html. 2

— (2023). "On functional logistic regression: some conceptual issues." *Test*, 32: 321–349. doi: https://doi.org/10.1007/s11749-022-00836-9. 3, 7, 27

Berrendero, J. R., Cholaquidis, A., and Cuevas, A. (2020b). "On a general definition of the functional linear model." *Preprint arXiv:2011.05441*. 2, 3, 5

Berrendero, J. R., Cuevas, A., and Torrecilla, J. L. (2016). "Variable selection in functional data classification: a maxima-hunting proposal." *Statistica Sinica*, 619–638. doi: https://doi.org/10.5705/ss.202014.0014. 3

— (2018). "On the use of reproducing kernel Hilbert spaces in functional classification." *Journal of the American Statistical Association*, 113(523): 1210–1218. doi: https://doi.org/10.1080/01621459.2017.1320287. 2, 3, 27

Borggaard, C. and Thodberg, H. H. (1992). "Optimal minimal neural interpretation of spectra." *Analytical Chemistry*, 64(5): 545–551. doi: https://doi.org/10.1021/ac00029a018. 13

Bro, R. (1999). "Exploratory study of sugar production using fluorescence spectroscopy and multi-way analysis." *Chemometrics and Intelligent Laboratory Systems*, 46(2): 133–147. doi: https://doi.org/10.1016/s0169-7439(98)00181-6. 13

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011). *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC. doi: https://doi.org/10.1201/b10905. 3

Bueno-Larraz, B. and Klepsch, J. (2019). "Variable Selection for the Prediction of $C[0,1]$-Valued Autoregressive Processes using Reproducing Kernel Hilbert Spaces." *Technometrics*, 61(2): 139–153. doi: https://doi.org/10.1080/00401706.2018.1505660. 3

Candès, E. J. and Sur, P. (2020). "The phase transition for the existence of the maximum likelihood estimate in high-dimensional logistic regression." *The Annals of Statistics*, 48(1): 27 – 42. doi: https://doi.org/10.1214/18-AOS1789. 7

Cardot, H. and Sarda, P. (2018). "Functional Linear Regression." In Ferraty, F. and Romain, Y. (eds.), *The Oxford Handbook of Functional Data Analysis*, 21–46. Oxford Handbooks. doi: https://doi.org/10.1093/oxfordhb/9780199568444.013.2. 3

Carey, J. R., Liedo, P., Müller, H.-G., Wang, J.-L., and Chiou, J.-M. (1998). "Relationship of Age Patterns of Fecundity to Mortality, Longevity, and Lifetime Reproduction in a Large Cohort of Mediterranean Fruit Fly Females." *The Journals of Gerontology: Series A*, 53(4): B245–B251. doi: https://doi.org/10.1093/gerona/53a.4.b245. 14

Celeux, G., Hurn, M., and Robert, C. P. (2000). "Computational and Inferential Difficulties with Mixture Posterior Distributions." *Journal of the American Statistical Association*, 95(451): 957–970. doi: https://doi.org/10.1080/01621459.2000.10474285. 23

Choi, T. and Ramamoorthi, R. V. (2008). "Remarks on consistency of posterior distributions." In Clarke, B. and Ghosal, S. (eds.), *Pushing the limits of contemporary statistics: contributions in honor of Jayanta K. Ghosh*, 170–186. Institute of Mathematical Statistics Collections. doi: https://doi.org/10.1214/074921708000000138. 2, 9

Crainiceanu, C. M. and Goldsmith, A. J. (2010). "Bayesian Functional Data Analysis Using WinBUGS." *Journal of Statistical Software*, 32(11): 1–33. doi: https://doi.org/10.18637/jss.v032.i11. 2

Cuevas, A. (2014). "A partial overview of the theory of statistics with functional data." *Journal of Statistical Planning and Inference*, 147: 1–23. doi: https://doi.org/10.1016/j.jspi.2013.04.002. 1

Cuevas, A., Febrero, M., and Fraiman, R. (2004). "An anova test for functional data." *Computational statistics & data analysis*, 47(1): 111–122. doi: https://doi.org/10.1016/j.csda.2003.10.021. 2

— (2007). "Robust estimation and classification for functional data via projection-based depth notions." *Computational Statistics*, 22(3): 481–496. doi: https://doi.org/10.1007/s00180-007-0053-0. 2

Delaigle, A. and Hall, P. (2012a). "Achieving near Perfect Classification for Functional Data." *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 74(2): 267–286. doi: https://doi.org/10.1111/j.1467-9868.2011.01003.x. 3, 27

— (2012b). "Methodology and theory for partial least squares applied to functional data." *The Annals of Statistics*, 40(1): 322–352. doi: https://doi.org/10.1214/11-aos958. 2, 27

Diaconis, P. and Freedman, D. (1986). "On the Consistency of Bayes Estimates." *The Annals of Statistics*, 14(1): 1–26. doi: https://doi.org/10.1214/aos/1176349830. 10

Doob, J. L. (1949). "Application of the theory of martingales." *Le calcul des probabilites et ses applications. Colloques Internationaux*, 13: 23–27. url: https://www.jehps.net/juin2009/Locker.pdf [at the end]. 9

Dudley, R. M. (2002). *Real Analysis and Probability*. Cambridge University Press. doi: https://doi.org/10.1017/CBO9780511755347. 26

Ferguson, T. S. (1974). "Prior Distributions on Spaces of Probability Measures." *The Annals of Statistics*, 2(4): 615 – 629. doi: https://doi.org/10.1214/aos/1176342752. 2

Ferraty, F., Hall, P., and Vieu, P. (2010). "Most-predictive design points for functional data predictors." *Biometrika*, 97(4): 807–824. doi: https://doi.org/10.1093/biomet/asq058. 3

Ferraty, F. and Vieu, P. (2006). *Nonparametric functional data analysis: theory and practice*. Springer. doi: https://doi.org/10.1007/0-387-36620-2. 2

Firth, D. (1993). "Bias reduction of maximum likelihood estimates." *Biometrika*, 80(1): 27–38. doi: https://doi.org/10.1093/biomet/80.1.27. 7

Folland, G. B. (1999). *Real Analysis: Modern Techniques and Their Applications*. John Wiley & Sons. 26

Foreman-Mackey, D., Hogg, D. W., Lang, D., and Goodman, J. (2013). "emcee: the MCMC hammer." *Publications of the Astronomical Society of the Pacific*, 125(925): 306–312. doi: https://doi.org/10.1086/670067. 13, 24, 25

Fraiman, R. and Muniz, G. (2001). "Trimmed means for functional data." *Test*, 10(2): 419–440. doi: https://doi.org/10.1007/bf02595706. 2

Galeano, P., Joseph, E., and Lillo, R. E. (2015). "The Mahalanobis Distance for Functional Data With Applications to Classification." *Technometrics*, 57(2): 281–291. doi: https://doi.org/10.1080/00401706.2014.902774. 2

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian Data Analysis*. Chapman and Hall/CRC. doi: https://doi.org/10.1201/b16018. 23

Ghosh, A. K. and Chaudhuri, P. (2005). "On Maximum Depth and Related Classifiers." *Scandinavian Journal of Statistics*, 32(2): 327–350. doi: https://doi.org/10.1111/j.1467-9469.2005.00423.x. 27

Ghosh, J. and Ramamoorthi, R. V. (2003). *Bayesian Nonparametrics*. Springer. doi: https://doi.org/10.1007/b97842. 2, 8, 9

Goia, A. and Vieu, P. (2016). "An introduction to recent advances in high/infinite dimensional statistics." *Journal of Multivariate Analysis*, 146: 1–6. doi: https://doi.org/10.1016/j.jmva.2015.12.001. 1

Goodman, J. and Weare, J. (2010). "Ensemble samplers with affine invariance." *Communications in Applied Mathematics and Computational Science*, 5(1): 65–80. doi: https://doi.org/10.2140/camcos.2010.5.65. 24

Green, P. J. (1995). "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination." *Biometrika*, 82(4): 711–732. doi: https://doi.org/10.1093/biomet/82.4.711. 23

Grollemund, P.-M., Abraham, C., Baragatti, M., and Pudlo, P. (2019). "Bayesian Functional Linear Regression with Sparse Step Functions." *Bayesian Analysis*, 14(1): 111 – 135. doi: https://doi.org/10.1214/18-BA1095. 3, 5, 23, 24

Hastie, T., Buja, A., and Tibshirani, R. (1995). "Penalized discriminant analysis." *The Annals of Statistics*, 23(1): 73–102. doi: https://doi.org/10.1214/aos/1176324456. 14

Hsing, T. and Eubank, R. (2015). *Theoretical Foundations of Functional Data Analysis, with an Introduction to Linear Operators*. John Wiley & Sons. doi: https://doi.org/10.1002/9781118762547. 1

James, G. M. and Hastie, T. (2001). "Functional Linear Discriminant Analysis for Irregularly Sampled Curves." *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 63(3): 533–550. doi: https://doi.org/10.1111/1467-9868.00297. 2

James, G. M., Wang, J., and Zhu, J. (2009). "Functional linear regression that's interpretable." *The Annals of Statistics*, 37(5A): 2083–2108. doi: https://doi.org/10.1214/08-aos641. 3

Jasra, A., Holmes, C. C., and Stephens, D. A. (2005). "Markov Chain Monte Carlo Methods and the Label Switching Problem in Bayesian Mixture Modeling." *Statistical Science*, 20(1): 50–67. doi: https://doi.org/10.1214/088342305000000016. 23

Jeffreys, H. (1946). "An invariant form for the prior probability in estimation problems." *Proceedings of the Royal Society of London Series A: Mathematical and Physical Sciences*, 186(1007): 453–461. doi: https://doi.org/10.1098/rspa.1946.0056. 6

Kalivas, J. H. (1997). "Two data sets of near infrared spectra." *Chemometrics and Intelligent Laboratory Systems*, 37(2): 255–259. doi: https://doi.org/10.1016/s0169-7439(97)00038-5. 13

Kang, H. B., Jung, Y. J., and Park, J. (2023). "Fast Bayesian Functional Regression for Non-Gaussian Spatial Data." *Bayesian Analysis*, Advance Publication: 1–32. doi: https://doi.org/10.1214/22-ba1354. 2

Lian, H., Choi, T., Meng, J., and Jo, S. (2016). "Posterior convergence for Bayesian functional linear regression." *Journal of Multivariate Analysis*, 150: 27–41. doi: https://doi.org/10.1016/j.jmva.2016.04.008. 2

Loève, M. (1948). "Fonctions aléatoires du second ordre." In Lévy, P. (ed.), *Processus stochastiques et mouvement Brownien*, 299–352. Gauthier-Villars. 4

López-Pintado, S. and Romo, J. (2009). "On the Concept of Depth for Functional Data." *Journal of the American statistical Association*, 104(486): 718–734. doi: https://doi.org/10.1198/jasa.2009.0108. 2

Lukić, M. N. and Beder, J. H. (2001). "Stochastic processes with sample paths in reproducing kernel Hilbert spaces." *Transactions of the American Mathematical Society*, 353(10): 3945–3969. doi: https://doi.org/10.1090/s0002-9947-01-02852-5. 4

Miller, J. W. (2018). "A detailed treatment of Doob's theorem." *Preprint arXiv:1801.03122*. 9

— (2023). "Consistency of mixture models with a prior on the number of components." *Dependence Modeling*, 11(1): 20220150. doi: https://doi.org/10.1515/demo-2022-0150. 2, 9, 10, 12

Miller, J. W. and Harrison, M. T. (2018). "Mixture Models with a Prior on the Number of Components." *Journal of the American Statistical Association*, 113(521): 340–356. doi: https://doi.org/10.1080/01621459.2016.1255636. 10

Müller, H.-G. and Stadtmüller, U. (2005). "Generalized functional linear models." *The Annals of Statistics*, 33(2): 774–805. doi: https://doi.org/10.1214/009053604000001156. 2

Nobile, A. (1994). "Bayesian analysis of finite mixture distributions." Ph.D. thesis, Carnegie Mellon University. 2, 10, 12

Papastamoulis, P. (2016). "label.switching: An R Package for Dealing with the Label Switching Problem in MCMC Outputs." *Journal of Statistical Software*, 69(Code Snippet 1): 1–24. doi: https://doi.org/10.18637/jss.v069.c01. 23

Parzen, E. (1961). "An Approach to Time Series Analysis." *The Annals of Mathematical Statistics*, 32(4): 951–989. doi: https://doi.org/10.1214/aoms/1177704840. 4

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É. (2011). "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research*, 12(85): 2825–2830. url: http://jmlr.org/papers/v12/pedregosa11a.html. 28

Piironen, J. and Vehtari, A. (2017). "Comparison of Bayesian predictive methods for model selection." *Statistics and Computing*, 27(3): 711–735. doi: https://doi.org/10.1007/s11222-016-9649-y. 23

Poß, D., Liebl, D., Kneip, A., Eisenbarth, H., Wager, T. D., and Barrett, L. F. (2020). "Superconsistent Estimation of Points of Impact in Non-Parametric Regression with Functional Predictors." *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 82(4): 1115–1140. doi: https://doi.org/10.1111/rssb.12386. 3

Preda, C., Saporta, G., and Lévéder, C. (2007). "PLS classification of functional data." *Computational Statistics*, 22(2): 223–235. doi: https://doi.org/10.1007/s00180-007-0041-4. 27

Ramos-Carreño, C., Torrecilla, J. L., Carbajo-Berrocal, M., Marcos, P., and Suárez, A. (2023). "scikit-fda: A Python Package for Functional Data Analysis." *Preprint arXiv:2211.02566*. 28

Ramsay, J. O. and Silverman, B. W. (2005). *Functional Data Analysis*. Springer. doi: https://doi.org/10.1007/b98888. 1, 2

Reiss, P. T., Goldsmith, J., Shang, H. L., and Ogden, R. T. (2017). "Methods for Scalar-on-Function Regression." *International Statistical Review*, 85(2): 228–249. doi: https://doi.org/10.1111/insr.12163. 3

Rodríguez, C. E. and Walker, S. G. (2014). "Label Switching in Bayesian Mixture Models: Deterministic Relabeling Strategies." *Journal of Computational and Graphical Statistics*, 23(1): 25–45. doi: https://doi.org/10.1080/10618600.2012.735624. 23

Scarpa, B. and Dunson, D. B. (2009). "Bayesian Hierarchical Functional Data Analysis Via Contaminated Informative Priors." *Biometrics*, 65(3): 772–780. doi: https://doi.org/10.1111/j.1541-0420.2008.01163.x. 2

Schervish, M. J. (1995). *Theory of Statistics*. Springer. doi: https://doi.org/10.1007/978-1-4612-4250-5. 9

Schwartz, L. (1965). "On Bayes procedures." *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 4(1): 10–26. doi: https://doi.org/10.1007/bf00535479. 9

Shi, J. Q. and Choi, T. (2011). *Gaussian Process Regression Analysis for Functional Data*. Chapman and Hall/CRC. doi: https://doi.org/10.1201/b11038. 2

Shin, H. (2008). "An extension of Fisher's discriminant analysis for stochastic processes." *Journal of Multivariate Analysis*, 99(6): 1191–1216. doi: https://doi.org/10.1016/j.jmva.2007.08.001. 2

Simola, U., Cisewski-Kehe, J., and Wolpert, R. L. (2021). "Approximate Bayesian computation for finite mixture models." *Journal of Statistical Computation and Simulation*, 91(6): 1155–1174. doi: https://doi.org/10.1080/00949655.2020.1843169. 23

Stephens, M. (2000). "Dealing With Label Switching in Mixture Models." *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 62(4): 795–809. doi: https://doi.org/10.1111/1467-9868.00265. 13, 23

Sur, P. and Candès, E. J. (2019). "A modern maximum-likelihood theory for high-dimensional logistic regression." *Proceedings of the National Academy of Sciences*, 116(29): 14516–14525. doi: https://doi.org/10.1073/pnas.1810420116. 7

Torrecilla, J. L., Ramos-Carreño, C., Sánchez-Montañés, M., and Suárez, A. (2020). "Optimal classification of Gaussian processes in homo- and heteroscedastic settings." *Statistics and Computing*, 30(4): 1091–1111. doi: https://doi.org/10.1007/s11222-020-09937-7. 3, 28

Torrecilla, J. L. and Suárez, A. (2016). "Feature selection in functional data classification with recursive maxima hunting." In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29 (NIPS 2016)*. url: https://proceedings.neurips.cc/paper/2016/hash/28b60a16b55fd531047c0c958ce14b95-Abstract.html. 27

Tuddenham, R. D. and Snyder, M. M. (1954). "Physical growth of California boys and girls from birth to eighteen years." *University of California Publications in Child Development*, 1(2): 183–364. url: https://pubmed.ncbi.nlm.nih.gov/13217130/. 14

Ullah, S. and Finch, C. F. (2013). "Applications of functional data analysis: A systematic review." *BMC Medical Research Methodology*, 13:43: 1–12. doi: https://doi.org/10.1186/1471-2288-13-43. 1

Van der Vaart, A. W. (1998). *Asymptotic Statistics*. Cambridge University Press. doi: https://doi.org/10.1017/CBO9780511802256. 9

Wales, D. J. and Doye, J. P. K. (1997). "Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms." *The Journal of Physical Chemistry A*, 101(28): 5111–5116. doi: https://doi.org/10.1021/jp970984n. 25

Yuan, M. and Cai, T. T. (2010). "A reproducing kernel Hilbert space approach to functional linear regression." *The Annals of Statistics*, 38(6): 3412–3444. doi: https://doi.org/10.1214/09-AOS772. 2

Zellner, A. (1986). "On Assessing Prior Distributions and Bayesian Regression Analysis with g-Prior Distributions." In Goel, P. and Zellner, A. (eds.), *Bayesian Inference and Decision Techniques: Essays in Honor of Bruno de Finetti*, volume 6, 233–243. Elsevier. 6

A Model choice and implementation details

A.1 Label switching

A well-known issue that arises when using MCMC methods in mixture-like models such as the one proposed in this work is *label switching*, which in short refers to the non-identifiability of the components of the model caused by their interchangeability. In our case, this happens because the likelihood is symmetric with respect to the ordering of the component parameters b and τ, that is, $\pi(Y|X,\theta) = \pi(Y|X,\nu(\theta))$ for any permutation ν that rearranges the indices $j = 1,\ldots,p$. Thus, since the components are arbitrarily ordered, they may be inadvertently exchanged from one iteration to the next in a MCMC algorithm. This can cause nonsensical answers when summarizing the marginal posterior distributions to perform inference, as different labelings might be mixed on each component (Stephens, 2000). However, this phenomenon is perhaps surprisingly a condition for the convergence of the MCMC method: as pointed out by many authors (e.g. Celeux et al., 2000), a lack of switching would indicate that not all modes of the posterior distribution were being explored by the sampler. For this reason, many ad-hoc solutions revolve around post-processing and relabeling the samples to eliminate the switching effect, but they generally do not prevent it from happening in the first place.

The most straightforward solutions consist on imposing an artificial identifiability constraint on the parameters to break the symmetry of their posterior distributions; see Jasra et al. (2005) and references therein. A common approach that seems to work well is to simply enforce an ordering in the parameters in question, which in our case would mean requiring for example that $\beta_i < \beta_j$ for $i < j$, or the analogous with the times in τ. We have implemented a variation of this method described in Simola et al. (2021), which works by post-processing the samples and relabeling the components to satisfy the order constraint mentioned above, choosing either b or τ depending on which set of ordered parameters would produce the largest separation between any two of them (suitably averaged across all iterations of the chains). This is an area of ongoing research, and thus there are other, more complex relabeling strategies, both deterministic and probabilistic. A summary of several such methods can be found for example in Rodríguez and Walker (2014) and Papastamoulis (2016).

A.2 Selection of hyperparameters

One of the key decisions in our Bayesian modeling scheme was whether to consider the number of components p as a member of the parameter space and integrate it into the model. While theoretically we could impose a prior distribution on p as well (e.g. a categorical distribution with a fixed maximum value), we found that this would have some unwanted practical implications. For instance, it would make the implementation more complex, since the dimensionality of the parameters b and τ would need to be fixed at a certain maximum value beforehand, but the working value of p within the MCMC algorithm would vary from one iteration to the next. In this case we would have no immediate way of tracking down which set of parameters is "active" at any given time. A simple approach would be to always consider the first p parameters and ignore the rest, and we did indeed try this technique, but it gave rise to new difficulties and the results obtained were not good. In fact, the label switching issue is accentuated when p is allowed to vary (c.f. Grollemund et al., 2019, Section 2.3), and on top of that, the interpretation of, say, the first coefficient β_1 in a model with 3 components is different from the interpretation of the same coefficient in a model with only 2 components.

This inconsistency in the interpretation of the components when the dimensionality of the model increases or decreases could be mitigated using a particular type of MCMC method known as reversible-jump MCMC (Green, 1995). Theoretically, these algorithms are specifically designed to approximate the posterior distribution in mixture-like models when the number of components is unknown, allowing the underlying dimensionality to change between iterations. However, since they are not yet widely adopted in practice and a reference implementation is not available, we decided against using them in our applications. Another possibility would be to adapt a purely Bayesian model selection technique to our framework (see Piironen and Vehtari, 2017; Gelman et al., 2013), or even derive some model aggregation methods to combine the posterior distributions obtained for different-sized models. These methods are usually based in computing a quantity known as the *Bayes factor*, which in turn requires the specific value of the normalizing integral constant we have been trying to avoid all along. In the end, for the sake of simplicity we decided to let p be a hyperparameter, so that we could use any model selection criteria (e.g. BIC, DIC or cross-validation) to select its optimal value.

As for the default values of the rest of hyperparameters in the prior distributions we propose, several comments are in order:

- For the expected value b_0 we propose to use the MLE of b. Although the likelihood function is rather involved, an approximation of the optimal value is enough for our purposes. Our numerical studies suggest that the results are much better with this choice than, say, with a random or null vector.
- We found that the parameter g does not have as much influence on the final result, and the experimentation indicates that $g = 5$ is a good enough value.
- Lastly, we observed that the choice of η can have a considerable impact on the final predictors. This is why, in an effort to normalize its scale, we consider a compound parameter $\eta = \tilde{\eta}\lambda_{\max}(\mathcal{X}_\tau^T\mathcal{X}_\tau)$, where $\lambda_{\max}(\mathcal{X}_\tau^T\mathcal{X}_\tau)$ is the largest eigenvalue of the matrix $\mathcal{X}_\tau^T\mathcal{X}_\tau$, and $\tilde{\eta} > 0$ is the actual tuning parameter (which can be selected for instance by cross-validation strategies). This standardization technique has been used previously in the literature; see for example Grollemund et al. (2019).

A.3 Affine-invariant ensemble sampler

An interesting and often desirable property of MCMC sampling algorithms is that they be *affine-invariant*, which means that they regard two distributions that differ in an affine transformation, say $\pi(x)$ and $\pi_{A,b}(Ax+b)$, as equally difficult to sample from. This is useful when one is working with very asymmetrical or skewed distributions, for an affine transformation can turn them into ones with simpler shapes. Generally speaking, a MCMC algorithm can be described through a function R as $\Lambda(t+1) = R(\Lambda(t), \xi(t), \pi)$, where $\Lambda(t)$ is the state of the chain at instant t, π is the objective distribution, and $\xi(t)$ is a sequence of i.i.d. random variables that represent the random behavior of the chain. With this notation, the affine-invariance property can be characterized as $R(A\lambda + b, \xi(t), \pi_{A,b}) = AR(\lambda, \xi(t), \pi) + b$, for all A, b and λ, and almost all $\xi(t)$. This means that if we fix a random generator and run the algorithm twice, one time using π and starting in $\Lambda(0)$ and a second time using $\pi_{A,b}$ with initial point $\Gamma(0) = A\Lambda(0) + b$, then $\Gamma(t) = A\Lambda(t) + b$ for all t. In Goodman and Weare (2010) the authors consider an ensemble of samplers with the affine invariance property. Specifically, they work with a set $\Lambda = (\Lambda_1, \dots, \Lambda_L)$ of *walkers*, where $\Lambda_l(t)$ represents an individual chain at time t. At each iteration, an affine-invariant transformation is used to find the next point, which is constructed using the current values of the rest of the walkers (similar to Gibb's algorithm), namely the *complementary ensemble*

$$\Lambda_{-l}(t) = \{\Lambda_1(t+1), \dots, \Lambda_{l-1}(t+1), \Lambda_{l+1}(t), \dots, \Lambda_L(t)\}, \quad l = 1, \dots, L.$$

To maintain the affine invariance and the joint distribution of the ensemble, the walkers are advanced one by one following a Metropolis-Hastings acceptance scheme. There are mainly two types of moves:

Stretch move. For each walker $1 \leq l \leq L$ another walker $\Lambda_j \in \Lambda_{-l}(t)$ is chosen at random, and the proposal is constructed as

$$\Lambda_l(t) \to \Gamma = \Lambda_j + Z(\Lambda_l(t) - \Lambda_j),$$

where $Z \overset{i.i.d.}{\sim} g(z)$ satisfying the symmetry condition $g(z^{-1}) = zg(z)$. In particular, the suggested density is

$$g_a(z) \propto \begin{cases} \frac{1}{\sqrt{z}}, & \text{if } z \in [a^{-1}, a], \\ 0, & \text{otherwise.} \end{cases}, \quad a > 1.$$

Supposing $\mathbb{R}^p$ is the sample space, the corresponding acceptance probability (chosen so that the detailed balance equations are satisfied) is:

$$\alpha = \min\left\{1,\ Z^{p-1}\frac{\pi(\Gamma)}{\pi(\Lambda_l(t))}\right\}.$$

Walk move. For each walker $1 \leq l \leq L$ a random subset $S_l \subseteq \Lambda_{-l}(t)$ with $|S_l| \geq 2$ is selected, and the proposed move is

$$\Lambda_l(t) \to \Gamma = \Lambda_l(t) + W,$$

where W is a normal distribution with mean 0 and the same covariance as the sample covariance of all walkers in S_l. The acceptance probability in this case is just the Metropolis ratio, that is, $\alpha = \min\{1, \pi(\Gamma)/\pi(\Lambda_l(t))\}$.

From a computational perspective, the Python library *emcee* (Foreman-Mackey et al., 2013) provides a parallel implementation of this algorithm. The idea is to divide the ensemble Λ into two equally-sized subsets $\Lambda^{(0)}$ and $\Lambda^{(1)}$, and then proceed on each iteration in the following alternate fashion:

1. Update *all* walkers in $\Lambda^{(0)}$ through one of the available moves explained above, using $\Lambda^{(1)}$ as the complementary ensemble.
2. Use the new values in $\Lambda^{(0)}$ to update $\Lambda^{(1)}$.

In this way the detailed balance equations are still satisfied, and each of the steps can benefit from the computing power of an arbitrary number of processors (up to $L/2$).

A.4 MCMC implementation

The MCMC method chosen for approximating the posterior distribution in our models is the affine-invariant ensemble sampler described in Appendix A.3. As mentioned there, we utilize the computational implementation in the Python library *emcee*, which is both reliable and easy to use; it aims to be a general-purpose package that performs well in a wide class of problems. One advantage of this method, apart from the property of affine-invariance, is that it only requires us to specify a few hyperparameters, irrespective of the underlying dimension. This contrasts to, say, the $\mathcal{O}(N^2)$ degrees of freedom corresponding to the covariance matrix of an N-dimensional jump distribution in Metropolis-Hastings. After selecting the number of iterations and the number of chains we want, we need to specify the initial points for each of them. As pointed out in Foreman-Mackey et al. (2013), a good initial choice is a Gaussian ball around a point in Θ_p that is expected to have a high probability with respect to the objective distribution. In our implementation we adopt this method, choosing an approximation of the MLE of θ as the central point in each case. To perform this approximation we employ the Basin-hopping optimization algorithm (Wales and Doye, 1997). This is a two-phase stochastic method that combines global steps with local optimization, in the hope of avoiding getting stuck too quickly in local maxima. To reduce the effects of randomness, we run the algorithm a few times and retain the point with the highest likelihood, and to avoid biasing the sampler too much towards the specific point selected, we let a fraction of the initial points be random (within reasonable bounds). This approximation is also used to specify the hyperparameter b_0.

Other less relevant hyperparameters include the burn-in period for the chains, which is the number of initial samples discarded, or the amount of thinning performed, which is the number of consecutive samples discarded to reduce the correlation among them. We use 64 chains and run them for 900 iterations in total, discarding the first 500 iterations as burn-in. Moreover, we use a weighted mixture of *walk* and *stretch* moves in the *emcee* sampler to advance the chains in each iteration, selecting the stretch move (the default) with probability 0.7 or the walk move with probability 0.3. Another computational decision we made is working with $\log \sigma$ instead of σ^2, so that the domain of this parameter is an unconstrained space, which is a widespread recommendation that helps increase the efficiency of the method.

B Measure-theoretic subtleties

There are some technicalities to take into account in the theoretical exposition in Section 3, especially pertaining to measure theory. For example, to justify the existence of regular conditional distributions such as $\boldsymbol{\theta}|X_1, \ldots, X_n$, one should see Theorem 10.2.1 and Theorem 10.2.2 in Dudley (2002), which guarantee they are well-defined provided that the underlying spaces are sufficiently regular. Another issue is the measurability of the mapping $\theta \mapsto P_\theta(X, Y)(A)$, which is assumed in the proof of the consistency results. We illustrate how this can be proved for example in the linear case, under the additional condition of sample continuity, which is arguably not a very restrictive condition in real-life scenarios.

Proposition B.1. *If the process X is sample-continuous (i.e. the trajectories are continuous functions), then the mapping $\theta \mapsto P_\theta(X, Y)(A)$ is measurable for every measurable set $A \subseteq \mathcal{X} \times \mathcal{Y}$.*

Proof. We start by checking that $\theta \mapsto P_\theta(Y|X)(A_1)$ is measurable for every measurable set $A_1 \subseteq \mathcal{Y}$. Indeed, consider the function $F(y, \theta) = f(y|X, \theta)\mathbf{1}_{A_1}(y)$, where $f(\cdot|X, \theta)$ is the density of the normal distribution $\mathcal{N}(\alpha_0 + \sum_j \beta_j X(t_j), \sigma^2)$ and $\mathbf{1}_A$ is the indicator function of a set A. It is easy to see that $F(y, \theta)$ is jointly measurable (it is in fact continuous) if X has continuous sample paths. Then, by Tonelli's theorem (e.g. Folland, 1999, Theorem 2.37), the function

$$\theta \mapsto \int_{\mathcal{Y}} f(y|X, \theta)\mathbf{1}_{A_1}(y)\, dy = \mathbb{E}_{P_\theta(Y|X)}\left[\mathbf{1}_{A_1}(Y)\right] = P_\theta(Y|X)(A_1)$$

is measurable. Now, if A is a measurable subset of $\mathcal{X} \times \mathcal{Y}$, we have

$$P_\theta(X, Y)(A) = \int_{\mathcal{X}} P_\theta(Y|X = x)(A_x) dP(X)(x),$$

where $A_x = \{y \in \mathcal{Y} : (x, y) \in A\}$ is the x-section of A. We just saw that $\theta \mapsto P_\theta(Y|X = x)(A_x)$ is measurable, and thus we conclude that $\theta \mapsto P_\theta(X, Y)(A)$ is measurable, since integration respects measurability. □

C Experimentation

C.1 Overview of data sets and comparison algorithms

To generate the simulated data sets for the comparison experiments of Section 4, we used four types of Gaussian process regressors commonly employed in the literature, each with a different covariance function:

BM. A Brownian motion, with kernel $K_1(t,s) = \min\{t,s\}$.

fBM. A fractional Brownian motion, with kernel $K_2(t,s) = 1/2(s^{2H} + t^{2H} - |t-s|^{2H})$ and Hurst parameter $H = 0.8$.

O-U. An Ornstein-Uhlenbeck process, with kernel $K_3(t,s) = e^{-|t-s|}$.

Gaussian. A Gaussian process with Gaussian kernel $K_4(t,s) = e^{-(t-s)^2/2\nu^2}$, where $\nu = 0.2$.

For the comparison algorithms themselves, we considered several frequentist methods which were selected among popular ones in FDA and machine learning in general. As was specified in Section 4, variable selection and dimensionality reduction methods are part of a pipeline followed by a standard multiple regression technique. In the linear regression case, we chose the following algorithms:

Manual. Dummy variable selection method with a pre-specified number of components (equispaced on $[0,1]$).

Lasso Linear least squares with l^1 regularization.

Ridge. Linear least squares with l^2 regularization.

PLS. Partial least squares for dimension reduction.

PCA. Principal component analysis for dimension reduction.

PLS1. Partial least squares regression (e.g. Abdi, 2010).

APLS. Functional partial least squares regression proposed by Delaigle and Hall (2012b).

RMH. Recursive maxima hunting variable selection method proposed by Torrecilla and Suárez (2016).

FLin. Functional L^2 linear regression model with fixed basis expansion.

FPCA. Functional principal component analysis.

FPLS1. Functional PLS regression through basis expansion, implemented as in Aguilera et al. (2010).

In the logistic regression case, all the variable selection and dimension reduction techniques from above were also considered, with the addition of the following classification methods:

Log. Standard multiple logistic regression with l^2 regularization.

LDA. Linear discriminant analysis.

QDA. Quadratic discriminant analysis.

RKVS. RKHS-based variable selection and classification method proposed in Berrendero et al. (2018).

APLS. Functional PLS used as a dimension reduction method, as proposed in Delaigle and Hall (2012a) in combination with the nearest centroid (NC) algorithm.

FLog. Functional RKHS-based logistic regression algorithm proposed in Berrendero et al. (2023).

FLDA. Implementation of the functional version of linear discriminant analysis proposed in Preda et al. (2007).

MDC. Maximum depth classifier (e.g. Ghosh and Chaudhuri, 2005).

FKNN. Functional K-nearest neighbors classifier with the L^2-distance.

FNC. Functional nearest centroid classifier with the L^2-distance.

The main parameters of all these algorithms are selected by cross-validation, using the same 5 folds as our proposed models so that the comparisons are fair. In particular, regularization parameters are searched among 20 values in the logarithmic space $[10^{-4}, 10^4]$, the number of manually selected variables is one of $\{5, 10, 15, 20, 25, 50\}$, the number of components for dimension reduction and variable selection techniques is in the set $\{2, 3, 4, 5, 7, 10, 15, 20\}$, the number of basis elements for cubic spline bases is in $\{8, 10, 12, 14, 16\}$, the number of basis elements for Fourier bases is one of $\{3, 5, 7, 9, 11\}$, and the number of neighbors in the KNN classifier is in $\{3, 5, 7, 9, 11\}$. Most algorithms have been taken from the libraries *scikit-learn* (Pedregosa et al., 2011) and *scikit-fda* (Ramos-Carreño et al., 2023), the first oriented to machine learning in general and the second to FDA in particular. However, some methods were not found in these packages and had to be implemented from scratch. This is the case of the FLDA, FPLS and APLS methods, which we coded following the corresponding articles.

C.2 Simulations with non-Gaussian regressors

We performed a set of experiments in linear and logistic regression in which the regressors are not Gaussian processes (GPs). These experiments where run in the same conditions as those reported in Section 4.

Functional linear regression

We use a geometric Brownian motion (GBM) as the regressor variable, defined as $X(t) = e^{BM(t)}$, where $BM(t)$ is a standard Brownian motion. In this case we consider two data sets, one with a RKHS response and one with an L^2 response, both with the same parameters as in the corresponding data sets in Section 4. The comparison results can be seen in Figure 7: in this case we still get better results under the RKHS model, while the results under the L^2-model are slightly worse. However, as with other simulations, the difference is small (except for the *emcee_mean* methods).

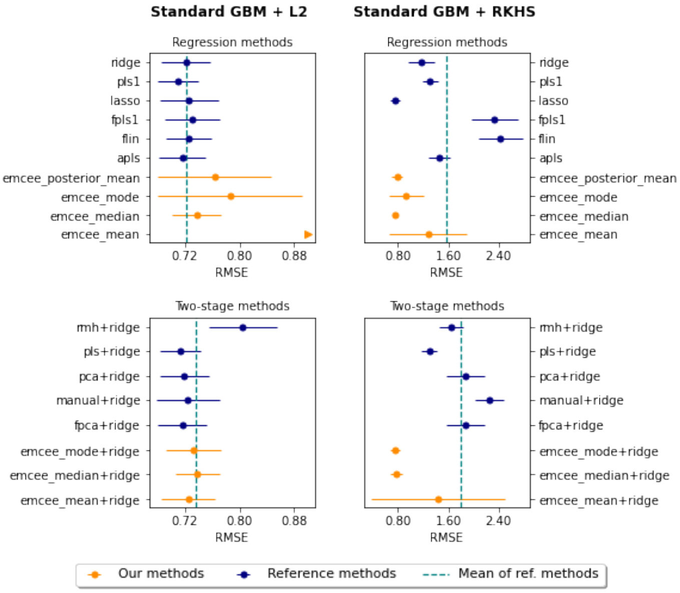

Figure 7: Mean and standard error of RMSE of predictors (lower is better) for 10 runs with GBM regressors. In the first column the response obeys a linear L^2-model, while in the second columns it follows a RKHS linear model. The first row are direct methods and the second are dimensionality reduction methods.

Functional logistic regression

We consider a "mixture" situation in which we combine regressors from two different GPs with equal probability and label them according to their origin. Firstly, we consider a homoscedastic case to distinguish between a standard Brownian motion and a Brownian motion with a mean function that is zero until $t = 0.5$, and then becomes $m(t) = 0.75t$. Secondly, we consider a heteroscedastic case to distinguish between a standard Brownian motion and a Brownian motion with variance 2, that is, with kernel $K(t, s) = 2\min\{t, s\}$. Figure 8 shows that our classifiers perform better than most comparison algorithms when separating two homoscedastic Gaussian processes, but they struggle in the heteroscedastic case. Incidentally, this heteroscedastic case of two zero-mean Brownian motions has a special interest, since it can be shown that the Bayes error is zero in the limit of dense monitoring (i.e. with an arbitrarily fine measurement grid), a manifestation of the "near-perfect" classification phenomenon analyzed for example in Torrecilla et al. (2020). Our results are in line with the empirical studies of this article, where the authors conclude that even though the asymptotic theoretical error is zero, most classification methods are suboptimal in practice (possibly due to the high collinearity of the data), with the notable exception of PCA+QDA.

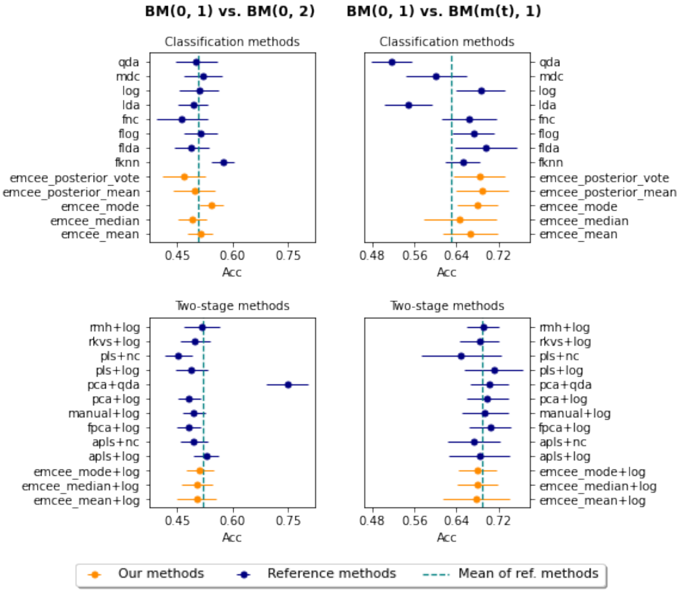

Figure 8: Mean and standard error of accuracy of classifiers (higher is better) for 10 runs with a mix of regressors coming from two different GPs and labeled according to their origin. In the first column we try to separate two Brownian motions with the same mean but different variance, while in the second column we discriminate between two Brownian motions with different mean functions but the same variance. The first row are direct methods and the second are dimensionality reduction methods.

C.3 Model misspecification

One requirement that our model should satisfy is that it ought to be able to recover the true parameter function when the underlying data generation model is a finite-dimensional RKHS model. This is generally the case when the value of p in our model and the true value p_0 coincide, but what happens when we change the value of p in the model? Take for example a RKHS data set with two components generated according to the formula $Y = 5 - 5X(0.1) + 10X(0.8) + \varepsilon$, with $\varepsilon \sim \mathcal{N}(0, 0.5)$. Figure 9 shows the resulting posterior distribution of the parameters $b = (\beta_1, \beta_2, \beta_3)$ and $\tau = (t_1, t_2, t_3)$ for a model with 3 components. As we can see, one of the coefficients has gone to zero to account for the overspecification of the model, while the other two have stabilized very close to the true parameters. The same goes for the time instants, except that in this case there is no default value to represent that a component is unused, so the time corresponding to the null coefficient oscillates back and forth. Note that the estimated function (based for example in the mode of the posterior distributions) will not be perfect, essentially because of the noise in the response. But it should be close to the true parameter function $\alpha(t) = -5K(t, 0.1) + 10K(t, 0.8)$, providing a good predictive performance in most cases.

In contrast, if we consider now a model with $p = 4$ with the same data, we might obtain posterior distributions like the ones in Figure 10. In this situation two coefficients should go to zero, but that is no longer the case. For example, while the green component has a high density around 0, it also has a considerable mass around 10, effectively "competing" with the red component. This is a manifestation of the label switching issue, caused in this case by an excessive number of degrees of freedom in the model. There is still another possible situation, one in which no label switching occurs but the estimated function has four non-negligible components. This can happen because the different components exploit the additional freedom and "work together", so to speak. In this way we might obtain an estimate that does not resemble the true coefficient function, but that has a very low prediction error. However, this could also work to our detriment and cause the estimated function to be worse prediction-wise than simpler alternatives. This phenomenon is expected to strengthen as the difference between the true and assumed value of p grows larger.

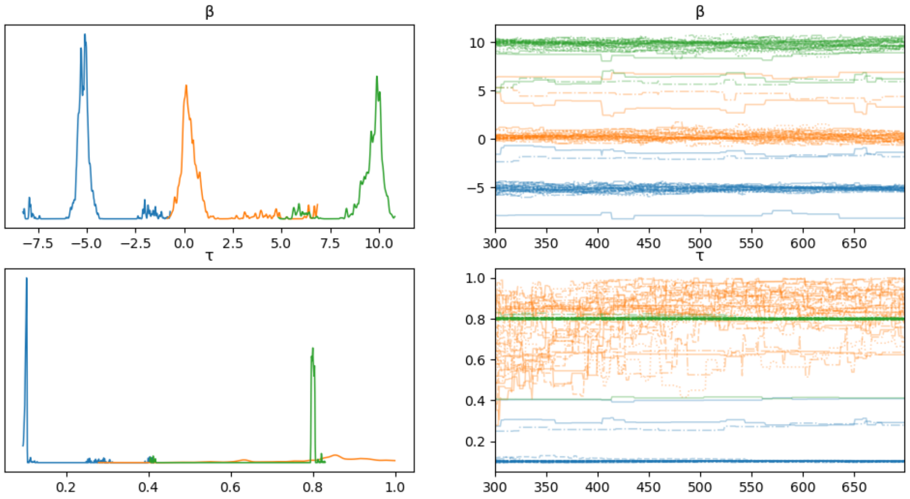

Figure 9: Left: estimated posterior distribution for the parameters b and τ using our RKHS linear regression model with $p = 3$, in a linear dataset with $p_0 = 2$ components. Right: the corresponding trace evolution for 400 iterations in the MCMC sampler.

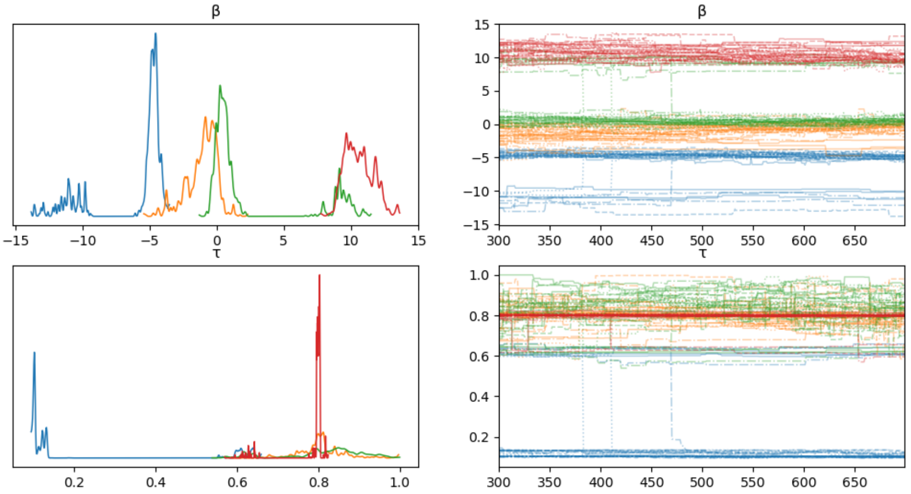

Figure 10: Left: estimated posterior distribution for the parameters b and τ using our RKHS linear regression model with $p = 4$, in a linear dataset with $p_0 = 2$ components. Right: the corresponding trace evolution for 400 iterations in the MCMC sampler.

C.4 Dependence on the number of components

Another thing we wanted to look at was the dependence of the final prediction result on the chosen value of p, especially when there is no concept of "components" in the underlying model. We can take for example the homoscedastic mixture data set described in Appendix C.2 for the logistic regression problem, and fix the parameter $\eta = 0.01$. The corresponding mean accuracies (in 10 repetitions) for RKHS models with $p = 1, 2, \ldots 10$ components are shown in Figure 11, along with their standard errors (which are arguably not very informative). It would appear that the methods that use the whole posterior distribution (*emcee_posterior_mean* and *emcee_posterior_vote*) are more stable and somewhat independent of the value of $p > 1$ in terms of accuracy. On the other hand, the rest of the algorithms show a slight downward trend as p increases (although not so much in the variable selection methods), and in general their best results are obtained at $p = 2$. We expect that this effect or some small variation of it will remain valid in other situations, and it would be in line with our view that the RKHS models work best with fewer components. However, a profound study of this would be the subject of a different experiment altogether.

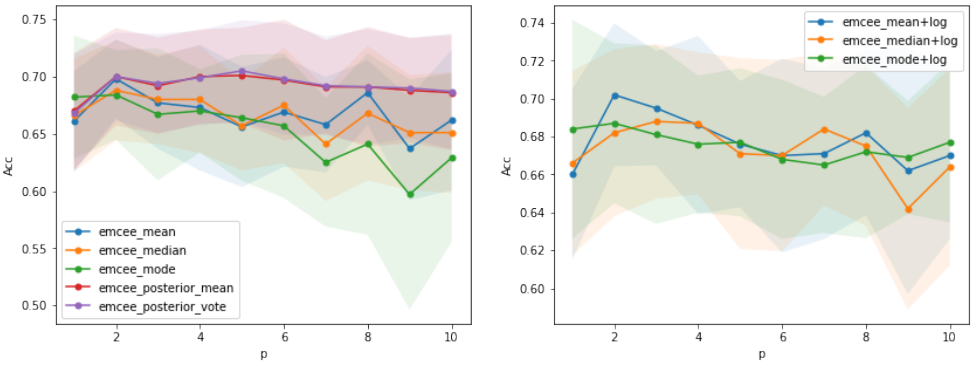

Figure 11: Mean accuracy in 10 independent repetitions for our logistic RKHS methods as a function of p, using $\eta = 0.01$ and the homoscedastic mixture data set. The corresponding standard errors are shown in faded colors.

C.5 Tables of experimental results

We present below the tables corresponding to the empirical comparison studies in Appendix C.2 and in Section 4, which show the numerical values that were depicted there graphically. In each case the best and second-best results are shown in **bold** and *italics*, respectively.

Functional linear regression

Prediction method	BM	fBM	O-U	Gaussian
emcee_mean	0.913 (0.310)	0.759 (0.068)	0.806 (0.098)	1.408 (1.359)
emcee_median	*0.729 (0.048)*	*0.729 (0.045)*	*0.740 (0.052)*	0.743 (0.041)
emcee_mode	0.735 (0.039)	0.748 (0.068)	0.769 (0.102)	0.803 (0.147)
emcee_posterior_mean	0.743 (0.047)	**0.726 (0.036)**	0.863 (0.416)	0.766 (0.061)
apls	1.003 (0.045)	0.792 (0.030)	1.167 (0.068)	*0.728 (0.035)*
flin	1.219 (0.056)	0.800 (0.022)	1.630 (0.051)	0.738 (0.030)
fpls1	1.235 (0.069)	0.800 (0.024)	1.631 (0.053)	0.738 (0.035)
lasso	**0.727 (0.034)**	0.738 (0.027)	**0.731 (0.039)**	**0.726 (0.032)**
pls1	1.032 (0.116)	0.782 (0.034)	0.974 (0.063)	0.729 (0.041)
ridge	0.920 (0.043)	0.778 (0.021)	0.965 (0.059)	*0.728 (0.035)*
emcee_mean+ridge	0.816 (0.154)	0.749 (0.044)	*0.734 (0.039)*	0.799 (0.175)
emcee_median+ridge	*0.759 (0.063)*	*0.741 (0.041)*	0.751 (0.065)	0.755 (0.058)
emcee_mode+ridge	**0.746 (0.058)**	**0.735 (0.036)**	**0.726 (0.038)**	0.735 (0.036)
fpca+ridge	1.149 (0.041)	0.784 (0.020)	1.420 (0.063)	*0.728 (0.033)*
manual+ridge	1.221 (0.050)	0.784 (0.021)	1.548 (0.072)	**0.727 (0.032)**
pca+ridge	1.153 (0.041)	0.784 (0.022)	1.422 (0.050)	0.730 (0.033)
pls+ridge	0.955 (0.053)	0.783 (0.031)	0.962 (0.059)	0.729 (0.035)
rmh+ridge	1.423 (0.117)	0.847 (0.043)	1.375 (0.266)	1.226 (0.117)

Table 1: Mean RMSE of predictors (lower is better) for 10 runs with GP regressors, one in each column, that obey an underlying linear RKHS model. The corresponding standard errors are shown between brackets.

Prediction method	**BM**	**fBM**	**O-U**	**Gaussian**
emcee_mean	0.769 (0.037)	0.744 (0.061)	0.756 (0.054)	0.794 (0.129)
emcee_median	0.730 (0.049)	0.751 (0.071)	0.718 (0.031)	**0.722 (0.030)**
emcee_mode	0.732 (0.032)	0.739 (0.075)	0.730 (0.038)	0.730 (0.029)
emcee_posterior_mean	0.723 (0.040)	0.720 (0.032)	0.755 (0.079)	*0.726 (0.026)*
apls	*0.715 (0.030)*	**0.710 (0.030)**	**0.710 (0.029)**	*0.726 (0.031)*
flin	0.733 (0.035)	0.727 (0.033)	0.733 (0.035)	0.735 (0.032)
fpls1	0.718 (0.039)	0.726 (0.035)	0.731 (0.034)	*0.726 (0.033)*
lasso	**0.712 (0.027)**	*0.712 (0.028)*	0.717 (0.029)	**0.722 (0.029)**
pls1	0.717 (0.041)	0.720 (0.036)	0.722 (0.029)	0.729 (0.031)
ridge	0.716 (0.029)	0.717 (0.032)	*0.716 (0.032)*	0.727 (0.033)
emcee_mean+ridge	**0.717 (0.029)**	*0.718 (0.030)*	0.719 (0.027)	0.743 (0.052)
emcee_median+ridge	0.722 (0.038)	0.723 (0.038)	*0.717 (0.035)*	0.730 (0.025)
emcee_mode+ridge	0.726 (0.036)	0.735 (0.048)	0.736 (0.030)	0.743 (0.050)
fpca+ridge	**0.717 (0.032)**	*0.718 (0.030)*	0.718 (0.033)	**0.727 (0.031)**
manual+ridge	**0.717 (0.030)**	0.719 (0.030)	**0.716 (0.032)**	*0.728 (0.031)*
pca+ridge	**0.717 (0.032)**	0.720 (0.031)	**0.716 (0.032)**	**0.727 (0.031)**
pls+ridge	*0.719 (0.033)*	0.728 (0.046)	0.720 (0.033)	0.730 (0.031)
rmh+ridge	0.753 (0.029)	**0.713 (0.030)**	0.791 (0.037)	0.812 (0.027)

Table 2: Mean RMSE of predictors (lower is better) for 10 runs with GP regressors, one in each column, that obey an underlying linear L^2-model. The corresponding standard errors are shown between brackets.

Prediction method	**GBM + $\mathbf{L^2}$**	**GBM + RKHS**
emcee_mean	0.948 (0.354)	1.278 (0.622)
emcee_median	0.737 (0.036)	**0.747 (0.031)**
emcee_mode	0.786 (0.106)	0.928 (0.275)
emcee_posterior_mean	0.763 (0.083)	0.786 (0.084)
apls	*0.716 (0.034)*	1.456 (0.170)
flin	0.726 (0.033)	2.427 (0.352)
fpls1	0.731 (0.040)	2.336 (0.365)
lasso	0.726 (0.042)	*0.759 (0.073)*
pls1	**0.710 (0.029)**	1.309 (0.122)
ridge	0.721 (0.035)	1.175 (0.205)
emcee_mean+ridge	0.725 (0.040)	1.432 (1.059)
emcee_median+ridge	0.738 (0.033)	*0.780 (0.093)*
emcee_mode+ridge	0.733 (0.040)	**0.760 (0.073)**
fpca+ridge	*0.716 (0.036)*	1.873 (0.302)
manual+ridge	0.724 (0.046)	2.253 (0.226)
pca+ridge	0.719 (0.036)	1.879 (0.304)
pls+ridge	**0.713 (0.030)**	1.299 (0.125)
rmh+ridge	0.805 (0.051)	1.640 (0.189)

Table 3: Mean RMSE of predictors (lower is better) for 10 runs with GBM regressors. In the first column the response obeys a linear L^2-model, while in the second column it follows a RKHS linear model. The corresponding standard errors are shown between brackets.

Prediction method	Moisture	Sugar	Tecator
emcee_mean	1.268 (1.096)	9.207 (9.248)	9.811 (7.446)
emcee_median	0.296 (0.051)	3.130 (2.584)	3.714 (0.922)
emcee_mode	0.301 (0.049)	2.628 (0.700)	3.531 (1.494)
emcee_posterior_mean	0.255 (0.039)	2.813 (0.897)	2.918 (0.222)
flin	0.257 (0.026)	1.978 (0.210)	*2.604 (0.344)*
fpls1	0.236 (0.038)	1.993 (0.223)	*2.604 (0.294)*
lasso	0.242 (0.028)	*1.975 (0.199)*	2.892 (0.270)
pls1	*0.228 (0.023)*	2.045 (0.190)	2.704 (0.467)
ridge	**0.221 (0.026)**	**1.952 (0.235)**	3.387 (0.218)
apls	0.234 (0.031)	2.050 (0.238)	**2.349 (0.470)**
emcee_mean+ridge	0.262 (0.043)	2.020 (0.198)	6.673 (1.037)
emcee_median+ridge	0.260 (0.034)	1.995 (0.219)	5.393 (1.210)
emcee_mode+ridge	0.302 (0.092)	2.037 (0.200)	5.442 (0.563)
fpca+ridge	0.289 (0.035)	*1.976 (0.227)*	9.521 (0.603)
manual+ridge	*0.228 (0.026)*	1.987 (0.227)	4.126 (0.305)
pca+ridge	**0.226 (0.027)**	**1.963 (0.234)**	*3.388 (0.218)*
pls+ridge	**0.226 (0.025)**	2.012 (0.218)	**2.415 (0.501)**
rmh+ridge	0.327 (0.086)	2.031 (0.216)	5.580 (0.513)

Table 4: Mean RMSE of predictors (lower is better) for 10 runs with real data sets, one in each column. The corresponding standard errors are shown between brackets.

Functional logistic regression

Classification method	BM	fBM	O-U	Gaussian
emcee_mean	0.743 (0.052)	0.739 (0.040)	0.734 (0.029)	0.751 (0.039)
emcee_median	0.771 (0.048)	0.716 (0.048)	0.714 (0.049)	0.746 (0.055)
emcee_mode	*0.777 (0.037)*	0.752 (0.043)	0.724 (0.029)	0.760 (0.048)
emcee_posterior_mean	0.764 (0.044)	0.756 (0.046)	0.734 (0.029)	0.753 (0.040)
emcee_posterior_vote	0.765 (0.043)	0.753 (0.040)	*0.747 (0.027)*	0.753 (0.043)
fknn	0.765 (0.027)	*0.768 (0.033)*	0.743 (0.032)	0.738 (0.022)
flda	0.767 (0.055)	0.755 (0.048)	0.735 (0.036)	*0.761 (0.050)*
flog	0.771 (0.033)	0.761 (0.042)	0.745 (0.025)	**0.777 (0.040)**
fnc	0.743 (0.029)	**0.775 (0.042)**	0.661 (0.063)	0.755 (0.035)
lda	0.514 (0.054)	0.601 (0.030)	0.578 (0.032)	0.702 (0.059)
log	**0.778 (0.031)**	0.750 (0.042)	**0.761 (0.039)**	*0.761 (0.031)*
mdc	0.724 (0.033)	0.762 (0.037)	0.648 (0.052)	0.732 (0.023)
qda	0.499 (0.038)	0.488 (0.041)	0.472 (0.055)	0.483 (0.027)
emcee_mean+logistic	0.781 (0.036)	0.746 (0.038)	0.725 (0.051)	0.750 (0.034)
emcee_median+logistic	0.766 (0.041)	0.749 (0.045)	0.717 (0.024)	0.732 (0.066)
emcee_mode+logistic	0.776 (0.042)	0.746 (0.047)	0.726 (0.025)	0.761 (0.036)
apls+log	*0.783 (0.025)*	0.756 (0.036)	0.739 (0.020)	0.761 (0.028)
apls+nc	0.771 (0.048)	0.745 (0.045)	0.740 (0.022)	0.751 (0.034)
fpca+log	0.773 (0.028)	0.755 (0.038)	**0.758 (0.032)**	0.765 (0.039)
manual+log	0.753 (0.033)	0.758 (0.040)	0.742 (0.031)	0.754 (0.032)
pca+log	0.780 (0.032)	0.758 (0.036)	*0.756 (0.032)*	0.756 (0.033)
pca+qda	0.751 (0.037)	0.750 (0.049)	0.736 (0.019)	0.741 (0.030)
pls+log	**0.786 (0.040)**	**0.768 (0.037)**	0.740 (0.035)	0.766 (0.033)
pls+nc	0.744 (0.032)	*0.766 (0.039)*	0.745 (0.055)	0.767 (0.035)
rkvs+log	0.770 (0.037)	0.757 (0.040)	0.738 (0.026)	*0.772 (0.039)*
rmh+log	0.768 (0.047)	0.760 (0.043)	0.745 (0.019)	**0.781 (0.036)**

Table 5: Mean accuracy of classifiers (higher is better) for 10 runs with GP regressors, one in each column, that obey an underlying logistic RKHS model. The corresponding standard errors are shown between brackets.

Classification method	BM	fBM	O-U	Gaussian
emcee_mean	0.571 (0.053)	*0.557 (0.037)*	**0.594 (0.021)**	0.565 (0.082)
emcee_median	0.557 (0.033)	0.539 (0.045)	0.559 (0.059)	0.556 (0.049)
emcee_mode	0.553 (0.063)	0.513 (0.067)	0.575 (0.038)	0.550 (0.042)
emcee_posterior_mean	0.575 (0.049)	**0.560 (0.036)**	*0.593 (0.022)*	**0.578 (0.039)**
emcee_posterior_vote	0.576 (0.034)	0.554 (0.039)	0.579 (0.023)	*0.576 (0.041)*
fknn	0.557 (0.022)	0.505 (0.047)	0.576 (0.042)	0.557 (0.026)
flda	0.554 (0.032)	0.516 (0.053)	0.543 (0.057)	0.526 (0.049)
flog	0.587 (0.034)	0.542 (0.036)	0.576 (0.038)	**0.578 (0.043)**
fnc	*0.601 (0.036)*	0.545 (0.045)	0.587 (0.037)	0.546 (0.056)
lda	0.507 (0.041)	0.482 (0.056)	0.524 (0.042)	0.525 (0.052)
log	0.576 (0.034)	0.546 (0.033)	0.560 (0.053)	0.554 (0.037)
mdc	**0.605 (0.039)**	0.544 (0.055)	0.584 (0.039)	0.533 (0.042)
qda	0.476 (0.050)	0.518 (0.048)	0.470 (0.071)	0.485 (0.039)
emcee_mean+logistic	**0.583 (0.038)**	**0.575 (0.043)**	0.569 (0.021)	0.559 (0.030)
emcee_median+logistic	0.565 (0.044)	0.552 (0.037)	*0.589 (0.029)*	**0.585 (0.041)**
emcee_mode+logistic	0.568 (0.047)	0.544 (0.036)	0.581 (0.041)	0.557 (0.033)
apls+log	0.556 (0.066)	0.535 (0.040)	0.548 (0.070)	0.560 (0.055)
apls+nc	0.549 (0.056)	0.523 (0.040)	0.545 (0.054)	0.545 (0.047)
fpca+log	0.559 (0.037)	0.548 (0.033)	0.579 (0.034)	0.564 (0.032)
manual+log	0.573 (0.037)	0.542 (0.029)	0.575 (0.043)	0.568 (0.035)
pca+log	0.570 (0.036)	0.541 (0.033)	0.579 (0.028)	0.567 (0.033)
pca+qda	0.567 (0.030)	0.532 (0.045)	0.577 (0.037)	0.574 (0.059)
pls+log	0.564 (0.042)	0.556 (0.028)	0.559 (0.041)	0.564 (0.036)
pls+nc	*0.581 (0.038)*	0.535 (0.048)	*0.589 (0.043)*	0.558 (0.057)
rkvs+log	0.572 (0.058)	0.550 (0.023)	**0.592 (0.018)**	0.567 (0.024)
rmh+log	0.570 (0.036)	*0.557 (0.033)*	0.584 (0.024)	*0.581 (0.025)*

Table 6: Mean accuracy of classifiers (higher is better) for 10 runs with GP regressors, one in each column, that obey an underlying logistic L^2-model. The corresponding standard errors are shown between brackets.

Classification method	**Heteroscedastic**	**Homoscedastic**
emcee_mean	0.513 (0.035)	0.667 (0.053)
emcee_median	0.492 (0.039)	0.647 (0.070)
emcee_mode	*0.543 (0.033)*	0.680 (0.038)
emcee_posterior_mean	0.497 (0.056)	*0.690 (0.050)*
emcee_posterior_vote	0.469 (0.058)	0.684 (0.048)
fknn	**0.574 (0.031)**	0.652 (0.034)
flda	0.489 (0.047)	**0.696 (0.059)**
flog	0.515 (0.045)	0.673 (0.040)
fnc	0.463 (0.069)	0.664 (0.053)
lda	0.493 (0.040)	0.548 (0.046)
log	0.509 (0.055)	0.686 (0.046)
mdc	0.521 (0.052)	0.601 (0.058)
qda	0.502 (0.056)	0.517 (0.039)
emcee_mean+logistic	0.503 (0.054)	0.678 (0.063)
emcee_median+logistic	0.504 (0.041)	0.680 (0.038)
emcee_mode+logistic	0.512 (0.036)	0.681 (0.036)
apls+log	*0.529 (0.034)*	0.684 (0.058)
apls+nc	0.496 (0.039)	0.674 (0.050)
fpca+log	0.481 (0.032)	*0.704 (0.041)*
manual+log	0.496 (0.029)	0.694 (0.044)
pca+log	0.483 (0.030)	0.699 (0.040)
pca+qda	**0.748 (0.055)**	0.703 (0.037)
pls+log	0.489 (0.043)	**0.711 (0.055)**
pls+nc	0.454 (0.037)	0.649 (0.076)
rkvs+log	0.499 (0.041)	0.684 (0.037)
rmh+log	0.516 (0.049)	0.691 (0.030)

Table 7: Mean accuracy of classifiers (higher is better) for 10 runs with a mix of regressors coming from two different GPs and labeled according to their origin. In the first column we try to separate two heteroscedastic Brownian motions, while in the second column we discriminate between two homoscedastic Brownian motions. The corresponding standard errors are shown between brackets.

Classification method	**Growth**	**Medflies**	**Phoneme**
emcee_mean	0.858 (0.147)	0.533 (0.041)	0.763 (0.041)
emcee_median	0.894 (0.112)	0.573 (0.032)	0.776 (0.044)
emcee_mode	0.932 (0.034)	0.582 (0.034)	0.770 (0.056)
emcee_posterior_mean	0.926 (0.032)	*0.596 (0.044)*	0.797 (0.035)
emcee_posterior_vote	0.919 (0.046)	0.575 (0.052)	*0.801 (0.031)*
fknn	0.942 (0.040)	0.534 (0.031)	0.760 (0.046)
flda	*0.945 (0.032)*	0.561 (0.020)	0.781 (0.037)
flog	0.935 (0.050)	**0.601 (0.029)**	0.766 (0.041)
fnc	0.735 (0.112)	0.546 (0.038)	0.703 (0.036)
lda	0.894 (0.052)	0.572 (0.019)	0.618 (0.040)
log	**0.965 (0.030)**	0.575 (0.028)	**0.822 (0.026)**
mdc	0.700 (0.087)	0.524 (0.026)	0.663 (0.031)
qda	0.581 (0.000)	0.569 (0.023)	0.457 (0.043)
emcee_mean+logistic	0.906 (0.118)	0.528 (0.029)	0.779 (0.033)
emcee_median+logistic	0.932 (0.049)	0.580 (0.031)	0.796 (0.036)
emcee_mode+logistic	0.935 (0.043)	0.585 (0.032)	0.791 (0.038)
apls+log	0.952 (0.026)	0.572 (0.016)	**0.816 (0.028)**
apls+nc	0.952 (0.041)	0.554 (0.020)	*0.807 (0.032)*
fpca+log	**0.965 (0.030)**	0.551 (0.032)	0.797 (0.021)
manual+log	*0.961 (0.032)*	0.584 (0.018)	0.778 (0.039)
pca+log	0.958 (0.029)	0.576 (0.030)	0.794 (0.030)
pca+qda	0.958 (0.032)	0.567 (0.036)	0.784 (0.034)
pls+log	0.952 (0.036)	0.578 (0.018)	0.804 (0.031)
pls+nc	0.829 (0.094)	0.570 (0.040)	0.754 (0.041)
rkvs+log	0.923 (0.052)	*0.596 (0.032)*	0.796 (0.031)
rmh+log	0.955 (0.030)	**0.606 (0.025)**	0.778 (0.048)

Table 8: Mean accuracy of classifiers (higher is better) for 10 runs with real data sets, one in each column. The corresponding standard errors are shown between brackets.

D Source code overview

The Python code developed for this work is available under a GPLv3 license at the GitHub repository https://github.com/antcc/rk-bfr. The code is adequately documented and is structured in several directories as follows:

- In the `rkbfr` folder we find the files responsible for the implementation of our Bayesian models, separated according to the functionality they provide.
- The `reference_methods` folder contains our implementation of the functional comparison algorithms that were not available through a standard Python library.
- The `utils` folder has some utility files for simulation, experimentation and visualization.
- The `experiments` folder contains plain text files with the numerical experimental results shown in Appendix C.5, as well as `.csv` and `.npz` files that facilitate working with them directly in Python.
- At the root folder we have files for executing our experiments, which accept several user-specified parameters (such as the number of iterations or the type of data set). In particular, the script `results_cv.py` contains the code for our comparison experiments, while the script `results_all.py` executes our Bayesian methods without a cross-validation loop.

When possible, the code was implemented in a generic way that would allow for easy extensions or derivations. It was also developed with efficiency in mind, so many functions and methods exploit the vectorization capabilities of the *numpy* and *scipy* libraries. Moreover, since we followed closely the style of the *scikit-learn* and *scikit-fda* libraries, our methods are compatible and could be integrated (after some minor tweaking) with both of them. The code for the experiments was executed with a random seed set to the value 2022 for reproducibility. We provide a script file `launch.sh` that illustrates a typical execution. Lastly, there are *Jupyter* notebooks that demonstrate the use of our methods in a more visual way. Inside these notebooks there is a step-by-step guide on how one might execute our algorithms, accompanied by many graphical representations, and offering the possibility of changing multiple parameters to experiment with the code. In addition, there is also a notebook that can be used to generate all the tables and figures of this document pertaining to the experimental results.