Classification of multivariate functional data on different domains with Partial Least Squares approaches

Issam-Ali Moindjié, Cristian Preda, Sophie Dabo-Niang

Classification of multivariate functional data on different domains with Partial Least Squares approaches

Issam-Ali Moindjié[1,2*], Cristian Preda[1,2] and Sophie Dabo-Niang[1,2]

[1] MODAL team, Inria de l'Université de Lille, France.
[2] Univ. Lille, CNRS, UMR 8524 , Laboratoire Paul Painlevé, F-59000, France.

*Corresponding author(s). E-mail(s): issam-ali.moindjie@inria.fr;
Contributing authors: cristian.preda@univ-lille.fr;
sophie.dabo@univ-lille.fr;

Abstract

Classification of multivariate functional data is explored in this paper, particularly for functional data defined on different domains. Using the partial least squares (PLS) regression, we propose two classification methods. The first one uses the equivalence between linear discriminant analysis and linear regression. The second is a decision tree based on the first technique. Moreover, we prove that multivariate PLS components can be estimated using univariate PLS components. This offers an alternative way to calculate PLS for multivariate functional data. Finite sample studies on simulated data and real data applications show that our algorithms are competitive with linear discriminant on principal components scores and black-boxes models.

Keywords: multivariate functional data analysis, supervised classification, Partial least squares regression (PLS), different domains

1 Introduction

In many areas, high-frequency data are monitored in time and space. For instance, (i) in medicine, a patient's state can be diagnosed by time-related (e.g. electroencephalogram, electrocardiogram) or images (e.g. fMRI) recording(s), (ii) in finance, stocks market are naturally recorded in time and space. Nowadays, in these areas and many others, space-time recordings result in massive high-frequency data. Analyzing such data requires adapted techniques.

For supervised classification purposes, umpteen works used several data reduction techniques such as feature selection to deal with this high dimension of data. Among these methods, the privileged features are driven by experts' knowledge, which could be non-applicable to another problem (see e.g Saikhu, Arifin, and Fatichah (2019), Javed, Rahim, Saba, and Rehman (2020)). Some other works focused on deep learning models, in particular, long short-term memory models have been proposed for time series (Hochreiter and Schmidhuber (1997), Karim, Majumdar, Darabi, and Chen (2017), Karim, Majumdar, Darabi, and Harford (2019)). They have the advantage of being less dependent on prior knowledge but are usually not interpretable. This can be a non-negligible issue when dealing with sensible application areas such as medicine.

Alternatively, functional data analysis, the statistical field of curves, images, shapes or more complex data, has been gaining interest in the field of time or space-time data. Since the pioneer works of Ramsey and Silverman (2005), functional data analysis is nowadays a well-established statistical research domain. Viewed as a sample of a random variable with values in some infinite dimensional space, functional data is mostly associated with a random variable indexed by a continuous parameter such as the time, wavelengths, or percentage of some cycle.

The supervised classification framework of univariate functional data – meaning that the predictor is a random variable with values in some space of real valued functions (e.g a time series) and the response is a categorical random variable – has been the source of diverse contributions. James and Hastie (2001) extended multivariate linear discriminant analysis (LDA) to irregularly sampled curves. As maximizing the between-class variance with respect to the total variance leads to an ill-posed problem, Preda, Saporta, and Lévéder (2007) proposed a partial least square-based classification approach for univariate functional data. Using the concept of depth, López-Pintado and Romo (2006) introduced robust procedures to classify functional data. Non-parametric approaches have also been investigated, using distances and similarities measures, see e.g Ferraty and Vieu (2003), and Galeano, Joseph, and Lillo (2015) for a overview of the use of Mahalanobis distance. Tree-based techniques applied to functional data classification are quite recent: Maturo and Verde (2022) introduced tree models using functional principal component scores as features, and Möller and Gertheiss (2018) presented a tree based on curve distances.

The multivariate functional data classification problem, i.e. the predictor is a multivariate functional variable (e.g. multivariate time series or images), was mainly investigated through the one domain dimension setting. This supposes that all univariate functional data have the same definition domain. Blanquero, Carrizosa, Jiménez-Cordero, and Martín-Barragán (2019b) proposed a methodology which allows optimal selection of the most informative time instants in multivariate functional data. Górecki, Krzyśko, and Wołyński (2015) used regression models to classify multivariate functional data by an orthonormal basis projection. Recently, Gardner-Lubbe (2021) proposed a linear discriminant analysis for multivariate functional data. To maximize the between and intra-class variance ratio, this author used discretization techniques, by pooling at a specific time or ignoring time dependence using time average.

Classification of multivariate functional data observed in different domains is rarely explored, despite its evident interest. Indeed, this framework makes possible the use of different types of data simultaneously (e.g. time series, images) (Happ & Greven, 2018). To the best of our knowledge, only Golovkine, Klutchnikoff, and Patilea (2022) proposed a supervised classification method in this setting. They introduced a tree-based method for unsupervised clustering, and demonstrated the applicability of their method to supervised classification. Nevertheless, as the clustering doesn't take into account the response variable, this can be problematic when clusters are not related to the target's distribution. Their method is based on principal component analysis for multivariate functional defined on different domains (MFPCA), presented in (Happ & Greven, 2018). The use of MFPCA as ordinary principal component analysis (PCA) for supervised learning leads to a non-trivial issue, which is the number of components to retain in the model since PCA is performed only with regard to the predictors.

Therefore, the partial least square (PLS) approach has been an interesting solution, as obtained components are based on the relationship between predictors and the response. Since the introduction of PLS regression on univariate functional data predictors by Preda and Saporta (2002), diverse works have been done, particularly in the univariate functional data setting. As already mentioned above, Preda et al. (2007) demonstrated the ability to use PLS for linear discriminant analysis. Aguilera, Escabias, Preda, and Saporta (2010) showed the relation between the PLS of univariate functional and ordinary PLS on the coefficient obtained by basis expansions. Delaigle and Hall (2012) developed an alternative non-iterative functional partial least square for regression, which helps to demonstrate consistency and establish convergence rates. For interpretability purpose, recently Guan, Lin, Groves, and Cao (2022) introduced a modified partial least square approach, which gives sparse coefficient function. To the best of our knowledge, PLS regression for multivariate functions has been explored only in Beyaztas and Shang (2022), in the one domain setting. They proposed a robust version of PLS for multivariate functional data by extending Aguilera et al. (2010) basis expansion results on multivariate functional data.

In this paper, we are interested in the classification of multivariate data defined in different domains. We go beyond Beyaztas and Shang (2022)'s recent contribution by investigating more exhaustively PLS procedure on multivariate functional data defined on different domains. We derive a clear relation between partial least squares regression on univariate functional data (FPLS) and partial least squares regression on multivariate functional data (MFPLS). In a practical case, this relation furnishes an alternative way to estimate PLS components for multivariate functional data using univariate components. The different dimension framework makes possible to use this PLS approach on data composed of heterogeneous types (e.g. images and time series). As our main goal is the classification problem, we then apply the proposed PLS methodology to the classification of multivariate functional data. In addition, we propose a more flexible classifier, namely a tree classifier based on PLS regression for binary classification of multivariate functional data. Our decision tree is inspired by the Tree Penalized Linear Discriminant Analysis (TPLDA) introduced by Poterie, Dupuy, Monbet, and Rouviere (2019). The main difference is due to the substitution of the penalized linear discriminant analysis with the partial least squares approach for multivariate functional data. Similarly to the TPLDA, our proposed tree allows using groups of dimensions. The groups are defined as a subset of dimensions which are not necessary disjoint. This method can capture more complex structures in functional data than MFPLS. Since multiple MFPLS/FPLSs are performed, it gives interpretable results. This can be an advantage in real data applications, where most algorithms used are black box.

The paper is organized as follows. Section 2 presents the partial least square methods for classification. In fact, after introducing the context, it presents the PLS regression on multivariate functional data defined on different domains. The relation with the Partial least Square for univariate functional data is given, and the based PLS linear discriminant analysis is described. The section finishes with the presentation of TMFPLS, the proposed decision tree, in the case of curves with variations inside classes. Section 3 presents simulation studies from regression to classification to compare the performances of our approaches with existing methods. To evaluate our classification procedure on real datasets, we apply our methods to benchmark data for multivariate time series classification in Section 4. A discussion is given in Section 5. The appendix contains the detailed proofs of all theoretical results. The supplementary material includes additional figures related to the numerical experiments.

2 Methods

2.1 Basic principles and notations

We are dealing with multivariate functional data defined on different domains in a similar framework as Happ and Greven (2018). As a general model for multivariate functional data analysis, let X be a stochastic process represented by

a d-dimensional vector of functional random variables $X = (X^{(1)}, \ldots, X^{(d)})^\top$, defined on the probability space $(\Omega, \mathcal{A}, \mathbb{P})$.

In the classical setting (Ramsey and Silverman (2005), Jacques and Preda (2014), Górecki et al. (2015)), the components $X^{(j)}$, $j = 1, \ldots, d$, are assumed real-valued stochastic processes defined on some finite continuous interval $[0, T]$. In our setting, we consider the general framework where each component $X^{(j)}$ is defined on some specific continuous compact domain $\mathcal{I}_j$ of $\mathbb{R}^{d_j}$, with $d_j \in \mathbb{N} - \{0\}$. Thus, for $d_j = 1$ we deal in general with time or wavelengths domains whereas for $d_j = 2$, the domain $\mathcal{I}_j$ indexes images or more complex shapes. It is also assumed that $X^{(j)}$ is a L_2-continuous process, and it has squared integrable paths, i.e. each trajectory of $X^{(j)}$ belongs to the Hilbert space of the square-integrable functions defined on $\mathcal{I}_j$, $L_2(\mathcal{I}_j)$. These general hypotheses ensure that integrals involving the variables $X^{(j)}$ are well-defined. Let define $\mathcal{H} = L_2(\mathcal{I}_1) \times ... \times L_2(\mathcal{I}_d)$ be the Hilbert space of vector functions

$$\mathcal{H} = \{f = (f^{(1)}, \ldots, f^{(d)}), f^{(j)} \in L_2(\mathcal{I}_j)\ , j = 1, \ldots, d\}$$

endowed with the inner product

$$\langle\langle f, g\rangle\rangle = \sum_{j=1}^{d} \langle f^{(j)}, g^{(j)}\rangle_{L_2(\mathcal{I}_j)} = \sum_{j=1}^{d} \int_{\mathcal{I}_j} f^{(j)}(t) g^{(j)}(t) dt.$$

where dt is the Lebesgue measure on $\mathcal{I}_j$. In the following, if there's no confusion, index $\mathcal{H}$ will be omitted and $|||, |||$ will denote the norm induced by $\langle\langle , \rangle\rangle$.

2.2 The Linear Functional Regression Model

When the aim is the prediction (the supervised context), the stochastic process X is associated to a response variable of interest Y through the conditional expectation $\mathbb{E}(Y|X)$.

Let consider the real-valued response variable Y be defined on the same probability space as X,

$$Y : \Omega \to \mathbb{R}.$$

Without loss of generality, we assume that Y and X are zero-mean,

$$\mathbb{E}(Y) = 0,\ \mathbb{E}(X^{(j)}(t)) = 0\ , \forall j \in \{1, ..., d\}, \forall t \in \mathcal{I}_j \tag{1}$$

and Y has finite variance.

The functional linear regression model assumes that $\mathbb{E}(Y|X)$ exists and is a linear operator as a function of X. Thus, we have:

$$Y = \langle\langle X, \beta\rangle\rangle_{\mathcal{H}} + \epsilon, \tag{2}$$

where

- $\beta \in \mathcal{H}$ denotes the regression parameter (coefficient) function,

$$\beta = \left(\beta^{(1)}, \ldots, \beta^{(d)}\right)^\top,$$

- ϵ denotes the residual term which is assumed to be of finite variance $\mathbb{E}(\epsilon^2) = \sigma^2$ and uncorrelated to X.

In the integral form, the model in (2) is written as :

$$Y = \sum_{j=1}^{d} \int_{\mathcal{I}_j} X^{(j)}(t)\beta^{(j)}(t)dt + \epsilon. \tag{3}$$

Under the least squares criterion, the estimation of the coefficient function β is, in general, an ill-posed inverse problem (Aguilera et al. (2010), Preda and Saporta (2002), Preda et al. (2007)). From a theoretical point of view, this is due to the infinite dimension of the predictor X, which makes that its covariance operator is not invertible (Cardot, Ferraty, & Sarda, 1999). Hence, dimension reduction methods such as principal component analysis (Happ and Greven (2018)) and expansion of X into a basis of functions (Aguilera et al. (2010)) can be used in order to obtain an approximation of linear form in (3).

2.2.1 Expansion (of the predictor) into a basis of functions

For each dimension j of $\mathcal{H}$, $j = 1, \ldots, d$, let consider in $L_2(\mathcal{I}_j)$ the set $\Psi^{(j)} = \{\psi_1^{(j)}, \ldots, \psi_{M_j}^{(j)}\}$ of M_j linearly independent functions. Denote with $M = \sum_{j=1} dM_j$.

Assuming that the functional predictor X and the regression coefficient function β admit the expansions

$$X^{(j)}(t) = \sum_{k=1}^{M_j} a_k^{(j)}\psi_k^{(j)}(t), \qquad \beta^{(j)}(t) = \sum_{k=1}^{M_j} b_k^{(j)}\psi_k^{(j)}(t), \tag{4}$$

$\forall t \in \mathcal{I}_j,\ j = 1, \ldots, d$, the functional regression model in (3) is equivalent to the multiple linear regression model:

$$Y = (Fa)^\top b + \epsilon \tag{5}$$

where

- a is the vector of size M obtained by concatenation of vectors $a^{(j)} = (a_1^{(j)}, a_2^{(j)}, \ldots, a_{M_j}^{(j)})$, $j = 1, \ldots, d$,
- b is the coefficient vector of size M obtained by concatenation of vectors $b^{(j)} = (b_1^{(j)}, b_2^{(j)}, \ldots, b_{M_j}^{(j)})$, $j = 1, \ldots, d$, and
- F is the block matrix of size $M \times M$ with diagonal blocks $F^{(j)}$, $j = 1, \ldots, d$,

$$F = \begin{pmatrix} F^{(1)} & 0 & ... & 0 \\ 0 & F^{(2)} & .... & 0 \\ 0 & 0 & ... & F^{(d)} \end{pmatrix}.$$

For each $j = 1, \ldots, d$, $F^{(j)}$ is the matrix of inner products between the basis functions, $F^{(j)}_{k,l} = \langle \psi^{(j)}_k, \psi^{(j)}_l \rangle_{L_2(\mathcal{I}_j)}$, $1 \leq k, l \leq M_j$.

Hence, under the assumption of basis expansion hypothesis (4), the estimation of the coefficient function, β, is equivalent to the estimation of the coefficient vector b in a classical multiple linear regression model with a design matrix involving the basis expansion coefficients of the predictor (the vector a) and the metric provided by the choice of the bases functions (the matrix F).

The least square criterion for the estimation of b yields in some settings (e.g. large number of basis functions) to multicollinearity and high dimension issues, similar to the univariate setting (see (Aguilera et al., 2010) for more details). Two well established methods of estimation, principal component regression (PCR) and partial least squares regression (PLS), are reputed for the efficiency of their estimation algorithm and interpretability of the results. As mentioned in Jong (1993) in the finite dimensional setting and in Aguilera et al. (2010) for the functional one, for a fixed number of components, the PLS regression fits closer than the PCR. Thus, the PLS regression provides a more efficient solution (sum of square errors'criterion). Numerical experiments confirm these results for the regression with univariate functional data (see for mode details Delaigle and Hall (2012), Guan et al. (2022)).
In the next section, we present the proposed PLS regression of multivariate functional data.

2.3 PLS regression with multivariate functional data: MFPLS

PLS regression penalizes the least squares criterion by maximizing the covariance between linear combinations of the predictor variables X (the PLS components) and the response Y. It is based on an iterative algorithm building at each step PLS components as predictors for the final regression model. In the multivariate setting, analogously to the univariate case (Preda and Saporta (2002)), the weights for the linear combinations are obtained as solution to the Tucker criterion:

$$\max_{w \in \mathcal{H}} \mathrm{Cov}^2(\langle \langle w, X \rangle \rangle, Y), \tag{6}$$

with $w = (w^{(1)}, \ldots, w^{(d)})^\top$ such that $|||w|||_{\mathcal{H}} = 1$.

The following proposition establishes the solution of (6).

Proposition 1 *The solution* $(w(t))_{t \in \mathcal{I}}$ *of* (6) *is given by*

$$w^{(j)}(t) = \frac{\mathbb{E}(X^{(j)}(t)Y)}{\sqrt{\sum_{k=1}^{d} \int_{\mathcal{I}_k} \mathbb{E}^2(X^{(k)}(s)Y)ds}}, \ \forall j,\ 1 \leq j \leq d, \ and \ \forall t \in \mathcal{I}_j. \tag{7}$$

□

Let denote by ξ the PLS component defined as the linear combination of variables X given by the weights w, i.e.,

$$\xi = \langle\langle X, w\rangle\rangle = \sum_{k=1}^{d} \int_{\mathcal{I}_k} X^{(k)}(t)w^{(k)}(t)dt.$$

The iterative PLS algorithm works as follows:

- Step 0: Let $X_0 = X$ and $Y_0 = Y$.
- Step h, $h \geq 1$: Define w_h as in Proposition 1 with $X = X_{h-1}$ and $Y = Y_{h-1}$. Then, define the h-th PLS component as

$$\xi_h = \langle\langle X_{h-1}, w_h\rangle\rangle,$$

Compute the residuals X_h and Y_h of the linear regression of X_{h-1} and Y_{h-1} on ξ_h,

$$X_h = X_{h-1} - \rho_h \xi_h,$$
$$Y_h = Y_{h-1} - c_h \xi_h,$$

where $\rho_h = \dfrac{\mathbb{E}(X_{h-1}\xi_h)}{\mathbb{E}(\xi^2)} \in \mathcal{H}$ and $c_h = \dfrac{\mathbb{E}(Y_{h-1}\xi_h)}{\mathbb{E}(\xi^2)} \in \mathbb{R}$.

- Go to the next step ($h = h + 1$).

Moreover, the following proprieties, stated in the univariate setting (Proposition 3 in Preda and Saporta (2002)), are still valid in the multivariate case. Let $L(X)$ denotes the linear space spanned by X.

Proposition 2 *Under the assumptions of Proposition 1, for any* $h \geq 1$

- $\{\xi_h\}_h$ *forms an orthogonal system of* $L(X)$
- $Y = c_1\xi_1 + c_2\xi_2 + ... + c_h\xi_h + Y_h$
- $X(t) = \rho_1(t)\xi_1 + \rho_2(t)\xi_2 + ... + \rho_h(t)\xi_h + X_h(t)$
- $\mathbb{E}(Y_h\xi_k) = 0,\ k = 1, \ldots, h$
- $\mathbb{E}(X_h^{(j)}(t)\xi_k) = 0,\ t \in \mathcal{I}_j\ j = 1, \ldots, d\ k = 1, \ldots, h.$

□

Nevertheless, the above properties don't furnish the direct relation between Y and X as in (2). The second and third points can be used to derive such relationship. The key idea is to determine the set of functions which allow calculating components ξ_h using only X. The following lemma gives the general form of these functions, here denoted by $\{v_h\}_h$.

Lemma 1 Let $\{v_h\}_h$, $v_h \in \text{span}\{w_1, \ldots, w_h\}$ be defined by :

$$v_h(t) = w_h(t) - \sum_{j=1}^{h-1} \langle\langle \rho_j, w_h \rangle\rangle v_j(t) \ , t \in \mathcal{I}. \tag{8}$$

We have, each v_h verifies

$$\xi_h = \langle\langle v_h, X \rangle\rangle.$$

□

This lemma shows that PLS regression defines a new set of functions such as X can be decomposed by

$$\forall t \in \mathcal{I} \ , X(t) = \sum_{i=1}^{h} \langle\langle v_i, X \rangle\rangle v_i(t) + X_{[h]}(t),$$

where $\xi_i = \langle\langle v_i, X \rangle\rangle$ is the score of X on $\{v_h\}_h$. The functions v_j are then suitable bases for regression, as they capture the maximum amount of information between X and Y according to Tucker's criterion. Moreover, this helps to derive a direct relation between X and Y.

$$Y = \langle\langle X, \beta^{[h]} \rangle\rangle + Y_{[h]},$$

with $\beta^{[h]}(t) = \sum_{i=1}^{h} c_i v_i(t)$ is order h approximation of the coefficient-function.

Using Lemma 1, scores of the functions $\{v_j\}$ on $\{w_j\}$ are determined using the following operations.

Let $\mathcal{V}_h$, and $\mathcal{W}_h$ be the concatenation of functions $\{v_k\}_k$ and $\{w_k\}_k$, such as

$$\mathcal{V}_h(t) = \begin{pmatrix} v_1^\top(t) \\ v_2^\top(t) \\ \ldots \\ v_h^\top(t) \end{pmatrix} \in \mathbb{R}^{h \times d}, \ \mathcal{W}_h(t) = \begin{pmatrix} w_1^\top(t) \\ w_2^\top(t) \\ \ldots \\ w_h^\top(t) \end{pmatrix} \in \mathbb{R}^{h \times d}. \tag{9}$$

Then equation (8) is equivalent to the following expression

$$\mathcal{V}_h(t) = \mathcal{W}_h(t) - P_{[h]}^{\mathcal{W}} \mathcal{V}_h(t), \tag{10}$$

where

$$P_{[h]}^{\mathcal{W}} = \begin{pmatrix} 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ \langle\langle \rho_1, w_1 \rangle\rangle & 0 & 0 & 0 & \ldots & 0 & 0 \\ \langle\langle \rho_1, w_2 \rangle\rangle & \langle\langle \rho_2, w_2 \rangle\rangle & 0 & 0 & \ldots & 0 & 0 \\ \langle\langle \rho_1, w_3 \rangle\rangle & \langle\langle \rho_2, w_3 \rangle\rangle & \langle\langle \rho_3, w_3 \rangle\rangle & 0 & \ldots & 0 & 0 \\ \ldots & & \ldots & & \ldots & & \ldots \\ \langle\langle \rho_1, w_h \rangle\rangle & \langle\langle \rho_2, w_h \rangle\rangle & \langle\langle \rho_3, w_h \rangle\rangle & \langle\langle \rho_4, w_h \rangle\rangle & \ldots & \langle\langle \rho_{h-1}, w_h \rangle\rangle & 0 \end{pmatrix} \in \mathbb{R}^{h \times h}.$$

Let $\mathbb{I}_{h\times h}$ be the identity matrix of size $h\times h$. Since $\mathbb{I}_{h\times h}+P^{\mathcal{W}}_{[h]}$ is non-singular[1], then equation(10) yields to

$$\mathcal{V}_h(t) = (\mathbb{I}_{h\times h} + P^{\mathcal{W}}_{[h]})^{-1}\mathcal{W}_h(t). \tag{11}$$

This finally gives the score of function $\{v_k\}_k$ on $\{w_k\}_k$.

Remark 1 It is worth noting that contrary to eigen-functions, $\{v_j\}_j$ is not orthogonal by the inner product $\langle\langle,\rangle\rangle$. Nonetheless, as they provide orthogonal components, they verify $\mathbb{E}(\xi_k\xi_l) = \mathbb{E}(\xi_k^2)\delta_{k,l}$, here $\delta_{k,l}$ is the Kronecker symbol. This implies that $\{v_j\}_j$ is an orthogonal basis function with respect to $\langle\langle,\rangle\rangle_*$, where

$$\langle\langle f,g\rangle\rangle_* = \sum_{i=1}^{d}\sum_{j=1}^{d}\int_{\mathcal{I}_i\times\mathcal{I}_j} f^{(i)}(s)\mathbb{E}(X^{(i)}(s)X^{(j)}(t))g^{(j)}(t)dsdt \quad f\in\mathcal{H}, g\in\mathcal{H}.$$

The next part focuses on the relationship between the partial least squares regression of univariate functional data (FPLS) and the proposed multivariate version (MFPLS) using Proposition 1.

2.3.1 Relationship between MFPLS and FPLS

Let $\tilde{w}_1^{(j)}$ be the first weight function obtained by FPLS on dimension j (Proposition 2 in Preda and Saporta (2002)):

$$\tilde{w}_1^{(j)}(t) = \frac{\mathbb{E}(X^{(j)}(t)Y)}{\sqrt{\int_{\mathcal{I}_j}\mathbb{E}^2(X^{(j)}(s)Y)ds}},\ t\in\mathcal{I}_j,\ 1\le j\le d.$$

Notice that functions $\tilde{w}_1^{(j)}$ and $w_1^{(j)}$ (as defined in Proposition 1) hold

$$w_1^{(j)}(t) = u_j\tilde{w}_1^{(j)}(t),\ u_j = \frac{\sqrt{\int_{\mathcal{I}_j}\mathbb{E}^2(X^{(j)}(s)Y)ds}}{\sqrt{\sum_{j=1}^{d}\int_{\mathcal{I}_j}\mathbb{E}^2(X^{(j)}(s)Y)ds}}, \text{and } t\in\mathcal{I}_j. \tag{12}$$

Moreover, $u = \left(u_1,...,u_d\right)^\top$ verifies $||u||_{\mathbb{R}^d} = 1$, where $||,||_{\mathbb{R}^d}$ denotes the Euclidean norm on $\mathbb{R}^d$. Thus, let consider $\xi_{j,1}$, the j-th associated component given by $\xi_{j,1} = \langle\tilde{w}_1^{(j)}, X^{(j)}\rangle$, $1\le j\le d$, since w_1 is solution of (6) then $\mathrm{Cov}^2(\sum_{i=1}^{d}u_i\xi_{i,1}, Y)$ is maximal. In other words, the two problems are equivalent: obtaining the first component ξ_1 is same as doing a classical PLS on *relative* components $\{\xi_{j,1}\}_j$.

In the next subsection, we use this relation to introduce a new methodology for the estimation of PLS on multivariate functional data using basis expansion.

[1] Since, it is a triangular matrix with 1 as diagonal terms

2.3.2 Partial least square estimation of multivariate functional data

Let the centered samples $X_1, ..., X_n$ of X, and $Y_1, ..., Y_n$ of Y. For each dimension $j \in \{1, \dots, p\}$, we consider a univariate basis function $\{\psi_k^{(j)}\}_{k\in\{1,...,M_j\}}$ of M_j functions, and the matrix A_j denotes the sample scores associated :

$$A_j = \begin{pmatrix} a_{1,1}^{(j)} & a_{1,2}^{(j)} & ... & a_{1,M_j}^{(j)} \\ a_{2,1}^{(j)} & a_{2,2}^{(j)} & ... & a_{2,M_j}^{(j)} \\ ... & & ... & \\ a_{n,1}^{(j)} & a_{n,2}^{(j)} & ... & a_{n,M_j}^{(j)} \end{pmatrix} \in \mathbb{R}^{n\times M_j},\ 1 \leq j \leq d, \tag{13}$$

where $X_i^{(j)}(t) = \sum_{k=1}^{M_j} a_{i,k}^{(j)} \psi_k^{(j)}(t),\ t \in \mathcal{I}_j$.
Define $\tilde{Y} = (Y_1, Y_2, ..., Y_n)^\top$ the Y observations, and F_j the matrix of the two-by-two inner products of basis $\{\psi_i^{(j)}\}_{i=1,...,M_j}$.
Let $F_j^{1/2}$ denotes the root matrix of F_j such as $F_j = (F_j^{1/2})^\top F_j^{1/2}$, then Proposition 2 in (Aguilera et al., 2010) makes partial least squares regression of multivariate functional data equivalent to the following steps:

1. For each dimension j, PLS of $\tilde{Y}$ on $A_j(F_j^{1/2})^T$ gives $\hat{\xi}_{j,1} \in \mathbb{R}^n$, the d *relative* first components.
   The relative weight functions are denoted by $\tilde{w}_1^{(j)}(t_j)$: $\tilde{w}_1^{(j)}(t) = (F_j^{-1/2}\tilde{v}^j)^\top \psi^{(j)}(t),\ t \in \mathcal{I}_j$, where $\tilde{v}^j$ denotes the projection vector obtained by PLS on dimension j.
2. Define the matrix, $\hat{\Xi}_1 = \left(\hat{\xi}_{1,1}\ \hat{\xi}_{2,1}\ ...\ \hat{\xi}_{d,1}\right) \in \mathbb{R}^{n\times d}$ which is the concatenation of components $\{\hat{\xi}_{j,1}\}_{1\leq j\leq d}$. Perform PLS of $\tilde{Y}$ on $\hat{\Xi}_1$ and define $u \in \mathbb{R}^d$, the first component weight vector.
   The final weight function is then given by $1 \leq j \leq d,\ t \in \mathcal{I}_j,\ w_1^{(j)}(t) = u[j]\tilde{w}_1^{(j)}(t)$, where $u[j]$ is the j-th element of the vector u.
3. Then the first multivariate component is given by $\hat{\xi}_1 = \hat{\Xi}_1 u,\ \hat{\xi}_1 \in \mathbb{R}^n$. Define[2] $\hat{\sigma}_1^2 = \hat{\xi}_1^\top \hat{\xi}_1$ its estimated variance, and ρ_1 the PLS coefficient function of X:

$$\rho_1^{(j)}(t) = (\zeta_1^j)^\top \psi^{(j)}(t),\ t \in \mathcal{I}_j, 1 \leq j \leq d,$$

   where $\zeta_1^j = \frac{1}{\sigma_1^2} A_j^\top \hat{\xi}_1$, and $\psi^{(j)}(t) = \left(\psi_1^{(j)}(t), ..., \psi_{M_j}^{(j)}(t)\right)^\top$. The PLS coefficient $c_1 \in \mathbb{R}$ of Y is given by $c_1 = \frac{1}{\sigma_1^2}\tilde{Y}^\top \hat{\xi}_1$
4. Residuals are given by $(A_j)_{[1]} = A_j - \xi_1(\zeta_1^j)^\top\ \forall j$, and $\tilde{Y}_{[1]} = \tilde{Y} - c_1\xi_1$ denotes the residual of Y.

[2]The $(n-1)^{-1}$ term is omitted since it is also used in the calculation of ρ_1, which leads to dividing $(n-1)$ by $(n-1)$.

5. Replacing A_j by $(A_j)_{[1]}$ and $\tilde{Y}$ by $\tilde{Y}_{[1]}$ in steps 1 to 4 gives the second component and associated residuals. This procedure is repeated to have the h components, and associated functions $(\{w_k\}_k, \{v_k\}_k, \{\rho_k\}_k)$.

6. The final step is the reconstruction of the set of functions $\{v_k\}_{1\leq k\leq h}$, this is done using scores of $\{w_k\}_{1\leq k\leq h}$ and $\{\rho_k\}_{1\leq k\leq h}$.
Recall the matrix $P^{\mathcal{W}}_{[h]}$ and $\mathcal{V}_h$ as defined in equations (9, 10), and let Ω_k and ζ_l the vectors such as

$$w_k(t) = \Psi(t)\Omega_k, \text{ and } \rho_l(t) = \Psi(t)\zeta_l,$$

where Ψ is the concatenation of the basis functions by dimensions

$$\Psi(t) = \begin{pmatrix} \psi_1^{(1)}(t_1) & \dots & \psi_{M_1}^{(1)}(t_1) & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & \psi_1^{(2)}(t_2) & \dots & \psi_{M_2}^{(2)}(t_2) & 0 \dots & 0 \\ & \dots & & & \dots & & & \\ 0 & \dots & 0 & & \dots & 0 & \psi_1^{(d)}(t_d) \dots & \psi_{M_d}^{(d)}(t_d) \end{pmatrix} \in \mathbb{R}^{d\times M},$$

such as $t_j \in \mathcal{I}_j$, $1 \leq j \leq d$.
We define the matrix ζ and Ω as $\zeta = \begin{pmatrix}\zeta_1 & \zeta_2 & \dots & \zeta_{h-1} & 0_{\mathbb{R}^M}\end{pmatrix}$, $\Omega = \begin{pmatrix}\Omega_1 & \Omega_2 & \dots & \Omega_{h-1} & \Omega_h\end{pmatrix}$.
Since $\{\langle\langle \rho_l, w_k\rangle\rangle\}_{k,l} = \{\zeta_l^\top F \Omega_k\}_{k,l}$, then

$$P^{\mathcal{W}}_{[h]} = \mathbf{Lower}(\Omega^\top F P),$$

where **Lower** is the operator matrix which sets terms on the upper triangular matrix and diagonal to 0.
Finally, we have

$$\mathcal{V}_h(t) = (\mathbb{I}_{h\times h} + \mathcal{P}^{\mathcal{W}}_{[h]})^{-1} \begin{pmatrix} \Omega_1^\top \\ \Omega_2^\top \\ \dots \\ \Omega_h^\top \end{pmatrix} \mathbf{Diag}(\Psi^\top(t), \dots, \Psi^\top(t)), \tag{14}$$

where **Diag** is the block diagonal matrix operator.

Remark 2 The approach proposed by Beyaztas and Shang (2022) is an extension of the basis expansion result from Aguilera et al. (2010). It was proposed for one domain definition. Note that our approach is more flexible since it allows different intervals. The case of one domain is then a special case of the proposed methodology (see Section 3.1.1 for numerical comparison).

Although the proposed methodology is for regression problems, it can be extended to classification by using the relation between linear discriminant analysis and linear regression (Aguilera et al. (2010), Preda et al. (2007). The next section addresses a classification application based on PLS regression.

2.3.3 From PLS regression to PLS binary-classification

Using previous notations, $(X(t))_{t\in\mathcal{I}}$ is the predictor function (not necessarily zero mean) and Y is the target, in this part we assume binary: $Y \in \{0,1\}$. The PLS regression can be extended to binary classification with the help of the following trick.
Firstly, we define the variable Y^*

$$Y^* = \begin{cases} \sqrt{\frac{\pi_1}{\pi_0}} & Y = 0 \\ -\sqrt{\frac{\pi_0}{\pi_1}} & Y = 1, \end{cases} \tag{15}$$

with $\pi_1 = \mathbb{P}(Y = 1)$ and $\pi_0 = \mathbb{P}(Y = 0)$.
A direct application of the law of total expectation[3] shows that $\mathbb{E}(Y^*) = 0$ and $\mathbb{V}(Y^*) = 1$, $\mathbb{V}$ denotes the variance.
Then, we fit a PLS model regression on $(X(t), Y^*)$ denoted by $\Gamma(X)$:

$$\Gamma(X) = \hat{\alpha} + \langle\langle X, \hat{\beta}^{[h]}\rangle\rangle,$$

with $\hat{\alpha}$= -$\langle\langle \mu, \hat{\beta}^{[h]}\rangle\rangle$, and $\mu(t) = \mathbb{E}(X(t))$.
Finally, the predicted class $\hat{Y}_0$ of a new curve $X_0(t)$ is given by

$$\hat{Y}_0 = \begin{cases} 0 \text{ if } \Gamma(X_0) > 0 \\ 1 \text{ otherwise.} \end{cases}$$

In practical case, π_1 is estimated by $\hat{\pi}_1 = 1/n \sum_{i=1}^{n} \mathbb{I}(Y_i = 1)$, where $\mathbb{I}$ is the indicator function, and n the sample size. The number of components h is chosen by cross-validation to maximize the AUC criterion.

2.4 Tree based on PLS regression of multivariate functional data

The linear relationship hypothesis between predictors and targets assumed in the previous sections is not always fulfilled. In this case, the partial least square model can be ineffective for regression and classification purposes. Linear models don't take into account the possibility of having several subclasses and so different coefficient functions. In this case, more flexible methods are usually preferred, which have more or less interpretability, such as non-linear SVM (see e.g Rossi and Villa (2006) Blanquero, Carrizosa, Jiménez-Cordero, and Martín-Barragán (2019a)), or clusterwise regression (see e.g Preda and Saporta (2005), Yao, Fu, and Lee (2011), Li, Song, Zhang, Zhu, and Zhu (2021)). In this part, we propose to use the tree structure to extend the linear discriminant analysis PLS to the case of semi-linear relationship. By doing so, the resulting discriminant analysis gains in adaptability and, for a reasonable number

[3] Indeed

- $\mathbb{E}(Y^*) = \pi_1\mathbb{E}(Y^*|Y = 1) + \pi_0\mathbb{E}(Y^*|Y = 0) = 0$
- $\mathbb{V}(Y^*) = \mathbb{E}((Y^*)^2) = \pi_1\mathbb{E}((Y^*)^2|Y = 1) + \pi_0\mathbb{E}((Y^*)^2|Y = 0) = 1$

of ramifications, gives easy interpretable results. The decision tree (TMFPLS) presented in this part is, by construction, less dependent to the linear relationship than the linear discriminant analysis presented in the previous part. TMFPLS or Tree Multivariate Functional Partial Least Square is designed for multivariate functional data predictors, but can be used for univariate functional data as a special case. It is, in a way, a generalization of Poterie et al. (2019)'s work to the case of multivariate functional data: the construction of a decision tree based on classification rules on some dimensions of predictors (groups). The procedure consists in decomposing a node by successively selecting a group of discriminating dimensions, and then applying MFPLS/FPLS to this group. This is repeated until the minimum purity criterion is reached. In the presented methodology, we limit our attention to the case of binary classification ($Y \in \{0, 1\}$).

2.4.1 Method

Let $((X_i(t))_{t\in\mathcal{I}}, Y_i)_{1\leq i\leq n}$ be the n observations of (X, Y), meaning that $(\{X_i(t)\}_{t\in\mathcal{I}}, Y_i)$ represents the i-th individual. Moreover, we suppose that X is composed of G groups of variables (not necessarily disjoint). This group structure is not restrictive, if this assumption does not hold, one can consider each dimension as a group, or consider one group of all dimensions.

For convenience, we introduce the following notations.

- $\mathcal{A}_p$ is a decision tree of depth $p \in \mathbb{N}$, see examples in Figure 1.
- $\mathcal{C}_j^p$ is the j-th node in depth p.
  $\mathcal{C}_1^0 := (X_i)_{1\leq i\leq n}$ is the root node, and each node $\mathcal{C}_j^p := X_{\mathcal{C}_j^p}$ can be seen as a partition of observations of X. Note that for depth $p : 1 \leq j \leq 2^p$.
- $\mathbf{card}\, \mathcal{C}_j^p$ denotes the cardinal of $\mathcal{C}_j^p$: $\mathbf{card}\, \mathcal{C}_1^0 = n$.
- $\pi_{1,\mathcal{C}_j^p}$, $\pi_{0,\mathcal{C}_j^p}$ are resp. the distribution of class 1 and 0 in $\mathcal{C}_j^p$: $\pi_{k,\mathcal{C}_j^p} = \mathbb{P}(Y = k|X \in \mathcal{C}_j^p)$, $k \in \{0, 1\}$.
  Hence, $\pi_1 = \mathbb{P}(Y = 1)$ is equivalent to $\pi_1 = \pi_{1,\mathcal{C}_1^0}$.
  These proportions can be estimated by

$$\hat{\pi}_{1,\mathcal{C}_j^p} = \frac{1}{\mathbf{card}\, \mathcal{C}_j^p} \sum_{i\in\{1,\ldots,n\}|X_i\in\mathcal{C}_j^p} \mathbb{I}(y_i = 1),$$

  where $\mathbb{I}$ is the indicator function. Moreover $\pi_{0,\mathcal{C}_j^p} = 1 - \pi_{1,\mathcal{C}_j^p}$.
- $\mathcal{G}$ is the set of group variables, and $G = \mathbf{card}\, \mathcal{G} \in \mathbb{N}^*$ is the number of groups.

Inspired by Poterie et al. (2019)'s methodology, our algorithm is composed of two main steps. In a nutshell, with the help of partial least squares, the first step gives potential splitting according to groups, and the second one selects the best splitting candidate using the Gini criterion. They are applied on nodes until the minimum purity threshold is reached, details are given below.

Consider node j at depth p: $\mathcal{C}_j^p$.

- **Step 1: PLS of functional data**.
  G FPLS/MFPLS are performed on $\mathcal{C}_j^p$, Γ^g denotes the estimated PLS model for group $g \in \mathcal{G}$.

$$\Gamma^g(X) = \hat{\alpha}^g + \langle\langle X^g, \hat{\beta}^g \rangle\rangle, \, 1 \leq g \leq G$$

  where $X^g = (X^{(d_1)}, ..., X^{(d_{n_g})})^\top$ and $g = \{d_1, ..., d_{n_g}\}$, meaning that n_g dimensions $\{\mathcal{I}_i\}_{1 \leq i \leq n_g}$ belong to group g.
  Thus, for each g, two sub-nodes of $\mathcal{C}_j^p$ are proposed.

$$\mathcal{C}_{2j}^{p+1}[g] = \{x \in \mathcal{C}_j^p, x | \Gamma^g(x) > 0\}$$
$$\mathcal{C}_{2j-1}^{p+1}[g] = \{x \in \mathcal{C}_j^p, x | \Gamma^g(x) \leq 0\}$$

- **Step 2: Splitting**.
  Select the optimal splitting according to group g^* which minimizes the *empirical* Gini criterion.

$$\mathcal{Q}(\mathcal{C}_j^p) = \pi_{1,\mathcal{C}_j^p}(1 - \pi_{1,\mathcal{C}_j^p})$$

  Taking into account nodes cardinal, group g^* should maximize the decrease of *impurity* function $\Delta\mathcal{Q}$.

$$\Delta\mathcal{Q}^g(\mathcal{C}_j^p) = \mathcal{Q}(\mathcal{C}_j^p)\,\mathbf{card}\,\mathcal{C}_j^d - \mathcal{Q}(\mathcal{C}_{2j-1}^{p+1}[g])\,\mathbf{card}\,\mathcal{C}_{2j-1}^{p+1}[g] - \mathcal{Q}(\mathcal{C}_{2j}^{p+1}[g])\,\mathbf{card}\,\mathcal{C}_{2j}^{p+1}[g]$$

$$g^* = \underset{g \in \mathcal{G}}{\arg\max}\, \Delta\mathcal{Q}^g(\mathcal{C}_j^p)$$

  Hence, the optimal splitting is the one obtained by Γ^{g^*}.

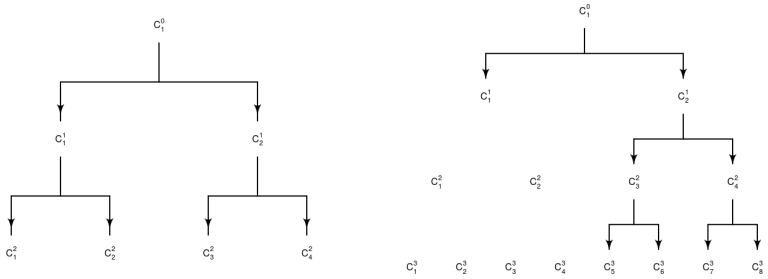
A node $\tilde{\mathcal{C}}$ is terminal if its impurity index is lower than a defined purity threshold $\eta \geq 0$: $\mathcal{Q}(\tilde{\mathcal{C}}) \leq \eta$.
Let $\mathcal{T}$ be the set of terminal nodes, if $\mathcal{C}_j^p$ is a terminal node, (i.e $\mathcal{C}_j^p \in \mathcal{T}$) then $\mathcal{C}_{2j}^{p+1} = \mathcal{C}_{2j-1}^{p+1} = \emptyset$, and if $\mathcal{C}_{j'}^{p'} = \emptyset$ then $\mathcal{C}_{2j'}^{p'+1} = \mathcal{C}_{2j'-1}^{p'+1} = \emptyset$.

2.4.2 Classify new curves

Consider $(X_0(t))_{t \in \mathcal{I}}$ a new function to classify, and define Y_0 its class we need to predict. Recall $\mathcal{T}$ the set of terminal nodes and $\mathcal{A}$ our decision tree. Let suppose that $\mathcal{A}$ has n_t terminal nodes: $\mathcal{T} = \{\tilde{\mathcal{C}}_1, ..., \tilde{\mathcal{C}}_{n_t}\}$. The law of total probability yields the equation

$$\mathbb{P}(Y_0 = y) = \sum_{j=1}^{n_t} \pi_{1,\tilde{\mathcal{C}}_j}^{y} \pi_{0,\tilde{\mathcal{C}}_j}^{1-y} \mathbb{I}\left(X_0 \in \tilde{\mathcal{C}}_j\right) \;\forall y \in \{0, 1\}.$$

The predicted class is then given by



$\mathcal{T}$ (a): $\{\mathcal{C}_1^2; \mathcal{C}_2^2; \mathcal{C}_3^2; \mathcal{C}_4^2\}$ (b): $\{\mathcal{C}_1^1; \mathcal{C}_5^3; \mathcal{C}_6^3; \mathcal{C}_7^3; \mathcal{C}_8^3\}$

Fig. 1: Tree examples, and their terminal nodes.

$$\hat{y}_0 = \underset{y \in \{0,1\}}{\arg\max}\, \mathbb{P}(Y_0 = y).$$

In other terms, our decision tree $\mathcal{A}$ gives the class of X_0 as the most present label in the terminal node $\tilde{\mathcal{C}}$ it belongs. The predicted class $\hat{Y}_0$ of $(X_0(t))_{t\in\mathcal{I}}$ is given by $\hat{Y}_0 = \arg\max_{y\in\{0,1\}} \pi_{y,\tilde{\mathcal{C}}}$.

2.4.3 Prevent over-fitting: Pruning

The strategy used to prevent over-fitting is the same as in (Poterie et al., 2019). The motivation is to find the optimal depth of our model. Considering that the tree $\mathcal{A}_p$ can be decomposed into several nested trees $\mathcal{A}_m$, each with a different set of terminal nodes $\mathcal{T}_m$ $(0 \leq m \leq p)$. The pruning goal is to find the nested tree $\mathcal{A}_{m*}$ such as

$$m^* = \underset{m\in\{1,\ldots,p\}}{\arg\max}\; r(Y_v, \hat{Y}_v[m]),$$

where r is a performance metric (ex: AUC, accuracy, ...), and $((X_v(t))_{t\in\mathcal{I}}, Y_v)$ a validation set, $\hat{Y}_v[m]$ denotes the prediction of Y_v obtained by the tree $\mathcal{A}_m$. For instance, the Table 1 gives the nested trees and terminal nodes of Figure 1 (b) tree.

m	$\mathcal{A}_m$	$\mathcal{T}_m$
0	$\{\mathcal{C}_1^0\}$	$\{\mathcal{C}_1^0\}$
1	$\{\mathcal{C}_1^0, \mathcal{C}_1^1, \mathcal{C}_2^1\}$	$\{\mathcal{C}_1^1, \mathcal{C}_2^1\}$
2	$\{\mathcal{C}_1^0, \mathcal{C}_1^1, \mathcal{C}_2^1, \mathcal{C}_3^2, \mathcal{C}_4^2, \emptyset\}$	$\{\mathcal{C}_1^1, \mathcal{C}_3^2, \mathcal{C}_4^2\}$
3	$\{\mathcal{C}_1^0, \mathcal{C}_1^1, \mathcal{C}_2^1, \mathcal{C}_3^2, \mathcal{C}_4^2, \mathcal{C}_5^3, \mathcal{C}_6^3, \mathcal{C}_7^3, \mathcal{C}_8^3, \emptyset\}$	$\{\mathcal{C}_1^1, \mathcal{C}_5^3, \mathcal{C}_6^3, \mathcal{C}_7^3, \mathcal{C}_8^3\}$

Table 1: **Fig** 1 (b): Nested trees

3 Simulations

This section focuses, with the help of synthetic data, on the comparison of our methods (MFPLS and TMFPLS) with each other and competitor method(s). Two main parts are presented, corresponding to the definition of domain settings: one domain and different domains. For ease of notation and seek of readability, we use X as predictor functions and Y as the variable of interest (scalar or binary response). In each simulation, their definitions are different, the same for ϵ which denotes the residual term (either function or scalar).

3.1 One domain

3.1.1 Sim 1.1: Scalar response

To show that our approach is competitive with the Beyaztas and Shang (2022) direct approach, we use their simulation framework for methods comparison.

Framework

Consider the domain $\mathcal{I} = [0, 1]$, and 3-dimensional functional predictors $X = (X^{(1)}, X^{(2)}, X^{(3)})^\top$ defined on $\mathcal{I}$.

$$X^{(j)}(t) = \sum_{j=1}^{5} \gamma_k \upsilon_k(t) \ \forall j, 1 \leq j \leq 3,$$

with $\gamma_k \sim \mathcal{N}(0, 4k^{-3/2})$ and $\upsilon_k(t) = \sin k\pi t - \cos k\pi t$.
The coefficient function β is defined by

$$\beta(t) = \big(\sin(2\pi t), \sin(3\pi t), \cos(2\pi t)\big)^\top,$$

then

$$Y = \langle\langle X, \beta\rangle\rangle_{\text{SIM}} + \epsilon.$$

$\langle\langle,\rangle\rangle_{SIM}$ denotes the inner product, here integrals are approximated by Simpson's rule, and $\epsilon \sim \mathcal{N}(0, \sigma^2)$.
This is the first difference, as our approach is computed with the help of Happ (2017) package, which approximates integral by the trapezoidal method. The second variation, which is the main difference, is methodological. Indeed, Beyaztas and Shang's approach uses an equivalence of PLS of multivariate functional data and ordinary PLS of the scores obtained in the projection of an arbitrary multivariate basis functions. They proposed a multivariate extension of Proposition 2 of Aguilera et al. (2010). Our approach is different, as we compute multivariate components using two-stage PLS. The first one is performed on univariate functional data, and the second one is done on partial components obtained (see section 2.3.2).

The Beyaztas and Shang's methodology and simulation are computed thanks to R scripts [4] shared by the authors: 200 times points are observed, and 400 replication of X are simulated. Among them, 200 replications are used for

[4] Avalaible on https://github.com/UfukBeyaztas/RFPLS

training and the left 200 are used for validation. For comparison, the principal component regression is also computed (by Beyaztas and Shang's scripts). Furthermore, we test values $\sigma^2 = 1$ and $\sigma^2 = 0$, meaning that Y is observed with and without noise. For each value of σ^2, we did 200 simulations.
The number of components in all approaches is chosen by 10-Fold cross-validation procedures, and data are projected in 20 basis spline functions.
Performances of approaches are measured by metrics[5]:

$$\text{MSPE} = \frac{1}{200}\sum_{i \in V_{\text{set}}} (Y_i - \hat{Y}_i)^2 \text{ , } \text{RISEE}_j = \frac{||\beta^{(j)} - \hat{\beta}^{(j)}||^2}{||\beta^{(j)}||^2},\ 1 \leq j \leq 3 \quad (16)$$

with $\hat{Y}_i$ is the predicted response for the $i-th$ individual in the validation set (V_{set}) and Y_i the actual value. RISEE is the relative integrated squared estimation error, for each method, it assesses the estimation error of the functional coefficient. $||.||$ denotes the $L^2([0,1], \mathbb{R})$ norm, it is approximated by the Riemann sum as in Beyaztas and Shang (2022).

Figure 2 depicts the MSPE boxplots of our approach (MFPLS), the direct approach (MFPLS_D) proposed by Beyaztas and Shang (2022), and the principal component regression (FPCA). Despite the differences in methods, MSPEs obtained are closed. Our method gives a relatively low error, particularly in the non-noise case. Since the Table 2 shows that coefficient function approximations are close, differences along the MSPE may then be due to the different integral approximations. This means that our approach and direct approach give equivalent results. Moreover, results show that our approach is competitive with principal component analysis regression.

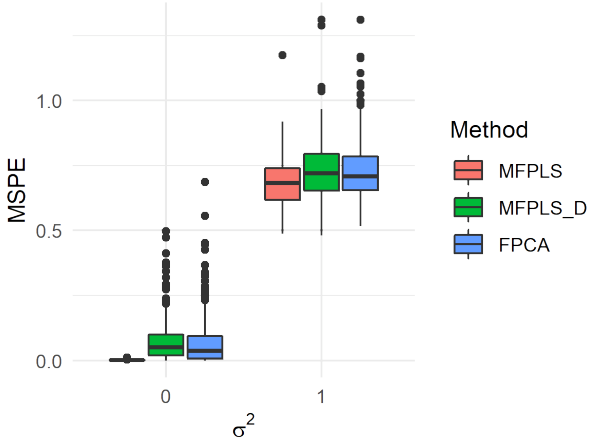


Fig. 2: Boxplots of MSPE obtained by each approach in Sim 1.1.

[5]MSPE stands for mean squared prediction error.

	σ^2	(1)	(2)	(3)
MFPLS		0.084	0.084	0.082
MFPLS_D	0	0.095	0.097	0.099
FPCA		0.084	0.085	0.083
MFPLS		0.158	0.177	0.159
MFPLS_D	1	0.162	0.165	0.157
FPCA		0.164	0.171	0.160

Table 2: Table of RISEE medians obtained (Sim 1.1). For each value of σ^2, the lowest value by dimensions: (1), (2), and (3) is underlined.

3.1.2 Sim 1.2: Binary response

In the following experiment, for ease of interpretation, we limit our simulation to a one-dimensional domain: $t \in \mathcal{I} := [0, 50]$.
In a practical case, for example in pattern recognition, the same class curves can have more or fewer variations and the rule to belong to one class be more *visual* than linear. Here, we allow intra-class curves to have positive and negative amplitudes and test the flexibility of both models (MFPLS, TMFPLS). For comparison purposes, as our approach gives similar results to the (non-robust) approach of (Beyaztas & Shang, 2022) in one domain (see Sim 1.1), we only compute linear discriminant analysis on principal component scores.

Simulation model:

The residual function is generated using Happ and Greven (2018) simulation framework. We consider N observed curves, the first step consists in simulating a bi-variate function $(d = 2)$ $\{e_i(t)\}_{1\leq i\leq N}$ given by $e_i(t) = \sum_{m=1}^{M} \theta_{i,m}\phi_m(t)$ with $\{\phi_m\}_{1\leq i\leq M}$ are the M first eigenfunctions associated to the covariance operator of $\{e_i(t)\}_{1\leq i\leq N}$. The scores on base ϕ_m of $e_i : \theta_m$ are independent, and follow a gaussian law $\mathcal{N}(0, \sigma^2_{\theta,m})$ where $\sigma^2_{\theta,m} = \frac{M+1-m}{M}$. Finally, ϵ_i which denotes the residual function for an individual i is defined using e_i by the following expression:

$$t \in \mathcal{I},\ \epsilon_i(t) = k_\epsilon e_i(t) \text{ with } k_\epsilon = \sqrt{\frac{2}{M+1}},$$

in order to have $\sum_m \langle\langle \epsilon_i, \phi_m \rangle\rangle \sim \mathcal{N}(0, 1)$.
Define the family of triangle functions $\{h_s(t)\}_{s\in\{1,2,3,4\}}$:

$$h_s(t) = \left(1 - \frac{|t - t_s|}{4}\right)_+ t \in \mathcal{I},$$

where $(.)_+$ denotes the positive part function.
Curves $X(t)$ are given by

$$X^{(1)}(t) = \sum_{s=1}^{4} a_s h_s(t) + \epsilon^{(1)}(t)\ , X^{(2)}(t) = \sum_{s=1}^{4} (1 - a_s) h_s(t) + \epsilon^{(2)}(t)$$

with $\{a_s\}_s$ are discrete variables in $\{-1, 0, 1\}$, and $Z \in \{1, 2\}$ a binary variable. The distribution of $\{a_s\}_s$ is generated to be dependent on the latent variable Z. Indeed, we have

$$\mathbb{P}(a_1 = 1|Z = z) = p_1^+[z], \text{ and } \mathbb{P}(a_1 = -1|Z = z) = p_1^-[z].$$

In addition, variables $\{a_j\}_j$ are dependent to each other in the following way

$$\mathbb{P}(a_{j+1} = k|a_j = k, Z = z) = p_{j+1}[z],\ 1 \leq j \leq 3$$
$$\text{and } \mathbb{P}(a_{j+1} = k_{j+1}|a_j \neq k_{j+1}, Z = z) = \frac{1}{2}(1 - p_{j+1}[z]),\ 1 \leq j \leq 3$$

with $k_{j+1} \in \{-1, 0, 1\}$.
And for the sake of simplicity, we make $\{a_j\}_j$ holds the Markov chain hypothesis

$$\mathbb{P}(a_{j+1} = k_{j+1}|a_1 = k_1, ..., a_j = k_j) = \mathbb{P}(a_{j+1} = k_{j+1}|a_j = k_j). \tag{17}$$

Finally, the distribution of Z is controlled by the parameter p

$$p = \mathbb{P}(Z = 2).$$

For given X, its class Y equals 1 if there are strictly two consecutive peaks at the beginning of its first dimension ($X^{(1)}$), where beginning means that peaks of interest can only be at $t = 10, 20$ or $t = 20, 30$. S_1 and S_2 denote those cases respectively. Moreover, for testing our model's flexibility, we allow peaks to have positive and negative amplitudes and add the constraint that consecutive peaks of interest should have the same amplitude sign.
This is summarized in the following equation

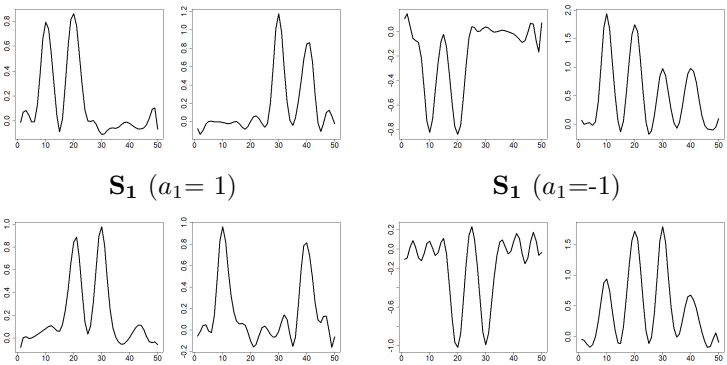
$$Y = \begin{cases} 1 \text{ if } \begin{array}{l} S_1 : |a_1| = 1, a_2 = a_1, a_3 = a_4 = 0\} \\ S_2 : \{a_1 = 0, |a_2| = 1, a_3 = a_2, a_4 = 0\} \end{array} \\ 0 \quad \text{otherwise.} \end{cases}$$

Under the above conditions and definitions, $\mathbb{P}(Y = 1)$ can be calculated (see appendix for details), and it is given by

$$\begin{aligned} \mathbb{P}(Y = 1) = (1 - p) \Big[& \frac{1}{2} p_4[1] p_2[1] (1 - p_3[1]) p_1[1] \\ & + \frac{1}{2} (1 - p_4[1]) p_3[1] (1 - p_2[1]) (1 - p_1[1]) \Big] \\ & + p \Big[\frac{1}{2} p_4[2] p_2[2] (1 - p_3[2]) p_1[2] \\ & + \frac{1}{2} (1 - p_4[2]) p_3[2] (1 - p_2[2]) (1 - p_1[2]) \Big] \end{aligned} \tag{18}$$

with $p_1[z] = p_1^+[z] + p_1^-[z]$.

One can notice that S_1 and S_2 are composed of two sub-events. For the first one, they correspond respectively to the case $a_1 = 1$, and $a_1 = -1$. In other words, the cases of positive consecutive peaks and negative consecutive peaks are at the first position $(10, 20)$. Analogously, if S_2 is true (consecutive peaks are at $t = 20, 30$), two cases can be distinguished $a_2 = 1$, or $a_2 = -1$. Examples are presented in Figure 3.



S₁ (a_1= 1) **S₁** (a_1=-1)

S₂ ($a_2 = 1$) **S₂** ($a_2 = -1$)

Fig. 3: Example of S_1, S_2 (Sim 1.2)

.

Numerical experiments

Two scenarios are studied, and parameters used for numerical simulation are presented in Table 3. The other parameters are fixed, the number of curves $N = 500$ and the number of eigenfunctions $M = 100$.

Parameters	**Scenario 1**		**Scenario 2**	
p	0.01		0.5	
Z	**1**	**2**	**1**	**2**
p_1^+	0.890	0.450	0.450	0.005
p_1^-	0.010	0.450	0.450	0.005
p_2	0.900	0.010	0.900	0.010
p_3	0.010	0.900	0.010	0.900
p_4	0.900	0.010	0.900	0.010

Table 3: Settings

Parameters used to simulate scenarios 1 and 2 (Sim 1.2).

Remark 3 Since $p = 0.01$ for the first scenario, curves are mainly in the $Z = 1$ regime. Moreover, as $p^+[1]$ equals 0.89 and $p_2[1]$ is set to 0.90 in most cases consecutive peaks

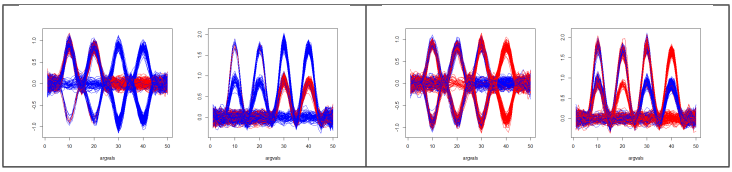


Fig. 4: Example of $X(t)$ in Sim 1.2 , under both scenarios($\kappa = 1$)

Class 1 curves are in red and class 0 curves are in blue.
The second scenario (right) shows more heterogeneity in class 1 compared to the first scenario (left).

are S_1 with the majority of positive peaks. This is the simplest scenario. MFPLS is expected to give good results. Indeed, it can estimate the coefficient function to have maximal weight on $t = 10, 20$. Since only two consecutive can belong to the first case, this will probably lead to miss-classify some curves, particularly those which have more than two consecutive peaks.
For the second scenario, we design a more difficult classification task. It highlights the limits of MFPLS and shows how TMFPLS can, in this case, be an interesting alternative. We set both Z regimes with the same probability to occur. The regime $Z = 1$, contrary to the previous scenario, allows consecutive peaks to have either positive or negative amplitude at $t = 10, 20$ (since $p_1^+[1] = p_1^-[1] = 0.45$). Regime $Z = 2$ is set as consecutive peaks occur at $t = 20, 30$. They also can be positive or negative.

The functional form of $X(t)$ is reconstructed using a cubic B-spline basis smoothing with 20 basis functions (see Figure 4). For the estimation, we set the minimum criterion η at 1% and the maximum tree depth at 10. For a given scenario, we did 200 experiments. At each, 75 % of the data are used for learning and 25 % for validation. By defining the set of groups as $\mathcal{G} = \{1, 2, \{1, 2\}\}$, we allow more flexibility in TMFPLS. Indeed, the decision tree tests whether the FPLS gives the best splitting or MFPLS does. The number of components for the MFPLS (in both models) or FPLS is chosen in the training set by 10-Fold cross-validation. Moreover, MFPCA-LDA is computed for comparison. It consists, firstly in the estimation of principal components (using Happ (2017) package), and then applying linear discriminant analysis to them. As in the previous models, the number of components is chosen by 10-Fold cross-validation. In addition, to have an estimation of the optimal depth m^*, we randomly take 75% of learning data to train an intermediate TMFPLS, and 25% for pruning (by AUC metric). This procedure is repeated 10 times. $\hat{m}^*$ is the most occurred number from the 10 obtained. The final tree is then trained on the whole learning data, with the maximum tree fixed to $\hat{m}^*$.

Results

In the first scenario (Scen 1), Table 4 shows that AUC differences along the scenario are about 10% between MFPLS and TMFPLS. Furthermore,

		AUC	Sensibility	Specificity
Scen 1	MFPLS	83.39	96.4	65.62
	TMFPLS	<u>93.90</u>	97.14	88.78
	MFPCA-LDA	92.60	100	80.34
Scen 2	MFPLS	50.79	60.61	41.13
	TMFPLS	<u>81.99</u>	87.67	76.27
	MFPCA-LDA	50.00	60.17	40.00

Table 4: Metrics in % obtained on the two scenarios (<u>Sim 1.2</u>).

At each step, 500 curves are simulated, 375 are used for learning and 125 for testing.
Metrics are the medians of the 200 obtained on each validation set.
The highest AUC by scenario is underlined.

MFPCA-LDA is very competitive with TMFPLS.
The second scenario shows more differences between the two methods, MFPLS and MFPCA-LDA are non-effective. In the following paragraphs, we analyze examples of trees obtained from the two scenarios to understand why there is such a discrepancy.

<u>Scenario 1</u>: Figure 5 depicts an estimated TMFPLS tree trained in the first scenario (after the pruning procedure). The associated discriminant coefficient functions are given in the supplementary materials. The first split is performed along the second dimension (see Figure 1 (a) in supplementary materials). MFPLS are then used on greater depths to obtain homogeneous classes. One can note that splitting is mainly performed to recognize class 0 curves, as it's the most heterogeneous class. The fact that its depth is greater than one explains why TMFPLS outperforms classical MFPLS.

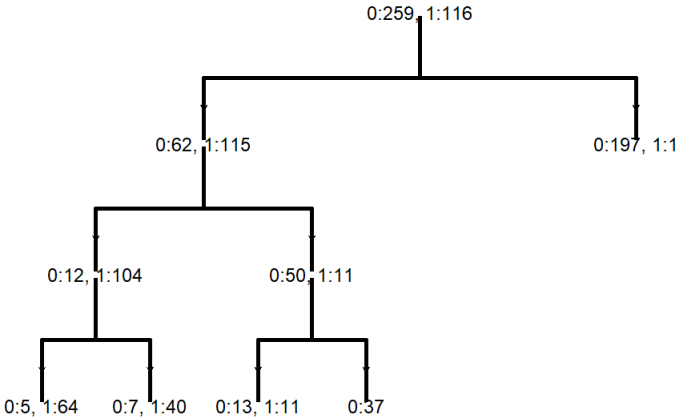


Fig. 5: Example of post pruned tree in scenario 1(Sim 1.2).

375 curves are used for training (75% of simulated data in each step).
For flexibility, groups considered are $\mathcal{G}$={1, 2, {1,2}}.

Scenario 2 In this scenario, the goal is to estimate a more complex rule of classification, as the amplitudes can be either positive or negative. Compared to the precedent scenario, the tree[6] in Figure 6 has more ramifications. This highlights how our method can be flexible, but also warns us of how it can easily over-fit our data. However, the results (in Table 6) demonstrate how this modelling can lead to very interesting improvements of simple MFPLS (or MFPCA-LDA) for the complex classification task.

[6]It's a tree obtained in one experiment among the 200 experiments we did.

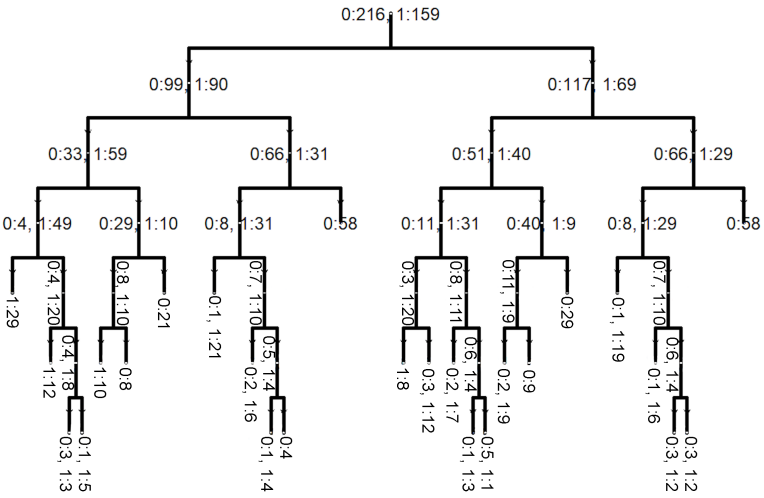


Fig. 6: Example of post pruned tree in scenario 2 of Sim 1.2.

375 curves are used for training (75% of simulated data at each step).
For flexibility, groups considered are $\mathcal{G}$={1, 2, {1,2}}.

For each estimated tree, several coefficient functions are computed, and as they are used for the split rule they can be useful for insight and interpretation. Associated coefficient functions for the presented trees are given in supplementary materials.

3.2 Different domains

3.2.1 Sim 2.1: Image and Time series classification

Our approach allows the use of images and time series simultaneously. In this part, we show the advantage, if they are available, to use both information instead of focusing on only one dimension domain. This simulation aims to be easily understandable, so we don't allow a lot of heterogeneity inside classes, then the use of TMFPLS is not necessary. In addition, since it is multi-domain data only MFPLS and MFPCA-LDA[7] (the principal component analysis is computed thanks to Happ (2017) package) are presented in this part.

Framework

Define $X(t) = \left(X^{(1)}(t_1), X^{(2)}(t_2)\right)^\top$, where the first dimension is a time series (TS) and the second represents an image (IM) of 50 × 50. The dimensions contain random noise ϵ and may contain a deterministic pattern h and q respectively for the first and second dimensions.

[7]Linear Discriminant Analysis computed on principal component scores

$$X^{(1)}(t) = h_{Z_1}(t) + \epsilon^{(1)}(t),\ t \in [0, 20]$$
$$X^{(2)}(t) = q_{Z_2}(t) + \epsilon^{(2)}(t),\ t \in [0, 1] \times [0, 1]$$

The noise term ϵ has two dimensions, the first one $\epsilon^{(1)}$ is a white noise of variances σ^2, and the second one $\epsilon^{(2)}$ is a gaussian random field. $\epsilon^{(2)}$ is simulated as a Matern covariance model, with parameters pairs: 0.25, 0.75, and nuggets parameter equal to σ, see Ribeiro Jr, Diggle, et al. (2001) for details. The deterministic patterns h and q (signals) depend on unobserved Bernoulli variables Z_1, $Z_2 \in \{1, 2\}$, such as:

$$h_{Z_1}(t) = \begin{cases} 3.14\left(1 - \frac{|t-10|}{4}\right)_+ & \text{if } Z_1 = 1 \\ 0 & \text{if } Z_1 = 2 \end{cases} \quad t \in [0, 20],$$

$$q_{Z_2}(t) = \begin{cases} -2\log\left(\sqrt{(t^{(1)} - 0.5)^2 + (t^{(2)} - 0.5)^2}\right) & \text{if } Z_2 = 1 \\ 0 & \text{if } Z_2 = 2 \end{cases} \quad t \in [0, 1] \times [0, 1].$$

with $(.)_+$ denotes the positive part, and (Z_1, Z_2) are independent.
The variable of interest Y is constructed with the help of (Z_1, Z_2), indeed Y=1, if both $\{Z_k\}_k$ are simultaneously 1. This means that X belongs to class 1 if on both dimensions we observe a pattern.

$$Y = \begin{cases} 1 \text{ if } Z_1 = 1 \text{ and } Z_2 = 1 \\ 0 \quad \text{otherwise.} \end{cases}$$

To get the functional form of X, the first dimension and second dimension

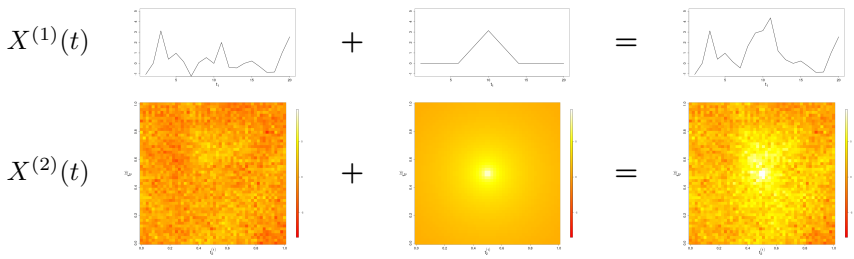


Fig. 7: Construction of class 1 ($Y = 1$), curve $X(t)$ in Sim 2.1, under SNR=0.5

If Y=0, X is random noise ϵ (left figures).

are projected respectively in 20 univariate quadratic spline functions, and 4 two-dimensional splines (one order).
The signal-to-noise ratios are given by :

$$\text{SNR}_{\text{TS}} = \frac{\text{V}_{h_1}}{\sigma^2}, \text{ and } \text{SNR}_{\text{IM}} = \frac{\text{V}_{q_1}}{\sigma^2},$$

where V_{h_1} and V_{q_1} are the variance along t_1 and t_2 of h_1 (which denotes h_{Z_1} if $Z_1 = 1$) and q_1 (which denotes q_{Z_2} if $Z_2 = 1$).

The sampling rates, for the first dimension and second dimension, are respectively 1 and 0.02, the estimated variance is $\text{V}_{h_1} \simeq 1$ and $\text{V}_{q_1} \simeq 1$, which means that $\text{SNR}_{\text{TS}} \simeq \text{SNR}_{\text{IM}}$, then the SNR on both dimension is practically the same, and depends only on σ.
Several values of σ are tested from 0.45 to $\sqrt{2}$ which leads to SNR varying from 0.5 to 4.9. The number of components in both approaches is chosen by 10-Fold cross-validation using AUC.

We set $\mathbb{P}(Z_1 = 1) = \mathbb{P}(Z_2 = 1) = 3/4$, then $\mathbb{P}(Y = 1) = 9/16 \simeq 0.56$. In addition, 500 curves are simulated: 375 are used for learning and 175 are used as a validation set. We did 200 simulations, and each approach is assessed by AUC on the validation set.

For each value of SNR, we compute the MFPCA-LDA and MFPLS on both dimensions (IM_TS), the first dimension(TS), and the second dimension(IM). Figure 8 (a) shows the boxplot for the two approaches computed on both dimensions, and Figure 8 (b) depicts the detailed performance of PLS along the three cases: IM, TS, and IM-TS. See the supplementary materials for the figure of MFPCA-LDA performances for the three cases.

Figure 8 shows that MFPLS gives better results than MFPCA-LDA for the lowest value of SNR, and the difference between methods disappears with the increase of SNR. Since information on classes is partially observed on dimensions, one domain classifications are not ineffective (AUCs are greater than 0.56). Nonetheless, Figure 8 **(b)** shows the clear advantage of using both dimensions.
This simulation demonstrates the ability of our method to classify different domain data. In addition, as it's specially designed for supervised learning, it can be more effective than principal component analysis-based techniques such as MFPCA-LDA in a noisy context.

4 Real data application: Time series classification

In this part, we compare our methods with black box models on benchmark data. The application is in the case of one domain definition, mainly because finding free-of-use datasets on different domain definitions is not easy. They are usually from medicine (neuroimaging, etc...), and so are often protected by medical secret. And as the majority of work on the classification of multivariate functional data is on one domain, it is convenient for the comparison of our methods with existing approaches.

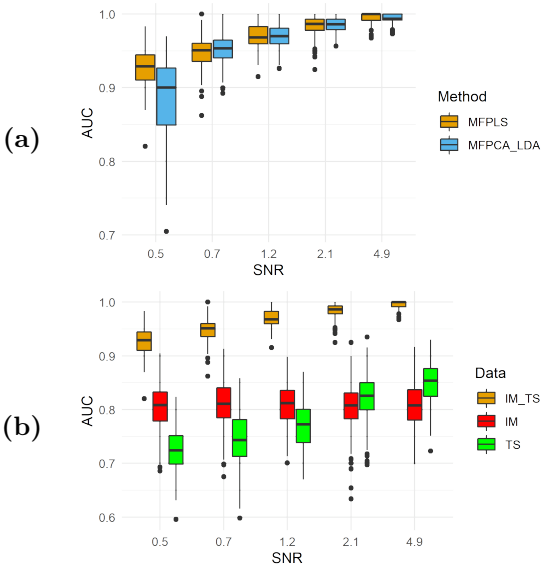


Fig. 8: Boxplots of Sim 2.1 results for different values of signal-to-noise ratio (SNR). **(a)** Comparison of MFPLS and MFPCA-LDA (computed on both dimension, image and time series), **(b)** Comparison of MFPLS on data of different domains (IM_TS: Image and Time series) and one domain MFPLS (IM: Image, TS: Time series).

4.1 Data presentation

Similarly to Karim et al. (2019), we use benchmark data (for time series classification) to obtain a more exhaustive performance of our methods (MFPLS and TMFPLS) on real data applications. The benchmark data are ranging from online character recognition to activity recognition. Some of them have already been used in Pei, Dibeklioğlu, Tax, and van der Maaten (2017), and Schäfer and Leser (2017). Since our methods can take only two-class problems, for now, we consider exclusively binary classification, which leads to taking into account 11 out of 35 available datasets. The following table summarizes the data used, it is part of Table 1 in Karim et al. (2019).

Dataset	*d*	T	Task	Ratio	Sources
CMUsubject16	62	534	Action Recognition	50-50 split	Carnegie (0)
ECG	2	147	ECG Classification	50-50 split	Oleszewski (2012)
EEG	13	117	EEG Classification	50-50 split	Lichman (2013)
EEG2	64	256	EEG Classification	20-80 split	(Lichman, 2013)
KickvsPunch	62	761	Action Recognition	62-38 split	Carnegie (0)
Movement AAL	4	119	Movement Classification	50-50 split	Lichman (2013)
NetFlow	4	994	Action Recognition	60-40 split	Sübakan, Kurt, Cemgil, and Sankur (2014)
Occupancy	5	3758	Occupancy Classification	35-65 split	Lichman (2013)
Ozone	72	291	Weather Classification	50-50 split	Lichman (2013)
Wafer	6	198	Manufacturing Classification	25-75 split	Oleszewski (2012)
WalkVsRun	62	1918	Action Recognition	64-36 split	Carnegie (0)

Table 5: Datasets summary. **T** denotes the number of sampling time points, **d**: the data dimensions, and **Ratio** of the train-test split. All datasets are available in https://github.com/titu1994/MLSTM-FCN/releases/tag/v1.0

We compare our models with the Long Short-Term Memory Fully Convolutional Network (LSTM-FCN) and Attention LSTM-FCN (ALSTM-FCN) proposed by Karim et al. (2019), and the state-of-the-art models. The objective is to show that our models can be competitive with black-box approaches. Since they are based on regression models, they have the advantage to be more interpretable. Data are split into training and test datasets (see Table 5). Models are compared by the accuracy metric, which is the rate of well-predicted classes obtained on the test datasets.

4.2 Results

The number of components in MFPLS is chosen by 20-fold cross-validation on training data. For comparison, we also compute linear discriminant analysis (LDA) on scores obtained by Multivariate functional principal component analysis (Happ, 2017). The number of principal components retained for LDA is also chosen by 20-fold cross-validation.

The maximum tree depth m^* is an important hyperparameter. It may significantly affect the performance of our model, as it helps to prevent the overfitting of TMFPLS. We estimate m^* by cross-validation alike procedure. More precisely, we randomly split 10 times the training data. At each split, 75% of the learning data have been used to train an intermediate TMFPLS and the remaining 25% is used for pruning by AUC. This operation yields 10 m^*. $\hat{m}^*$ is the most frequently occurring element in those. Then the final TMFPLS is trained on the whole training data with the constraint that the maximum depth tree cannot exceed $\hat{m}^*$. As in the previous section, group are defined as $G = \{1, 2, ..., d, \{1, ..., d\}\}$. We test whether FPLS gives better splitting than MFPLS. This choice is motivated by the fact that testing more combinations of dimensions is time-consuming. The ideal choice of groups should be driven by some prior knowledge of the data structure.

Two strategies are used for the number of components in the decision tree: **TMFPLS H-1** denotes the decision tree where only one component in FPLS/MFPLS is used, and **TMFPLS H-CV** is the decision tree where the H components are estimated by 20-fold cross-validation as in MFPLS. The

first tree is faster to train than the second one, and it's less likely to overfit the data. However, the second is often a more efficient tree, since it estimates more complex coefficient functions β than the second one.

For all functional data methods, we use 30 B-Splines basis functions by dimension to have a curve representation of each dataset. This number is arbitrary, the use of more or fewer functions can yield to different results. The first justification for this number of basis functions is for most cases it seems enough for having a good curve representation (see supplementary materials for figures of the smoothed functions). In addition, as it is relatively a few basis functions, this shows how our methods can reach acceptable performances with less complexity than black box models.

Our models are compared with non-FDA classification methods. The table 6 shows the state-of-the-art method (SOTA) and the best results obtained by Karim et al. (2019) (among LSTM-FCN and ALSTM-FCN). The SOTA column presents as well obtained baselines by Karim et al. (2019). Those baselines are acquired on these datasets by using techniques like Dynamic time warping (DTW), Random Forest (RF), SVM with a linear kernel, and SVM with a 3rd degree polynomial kernel (SVM-Poly) and choosing the highest score as the baseline.

Datasets	$\mathbf{N_{Obs}}$	$\mathbf{N_{Test}}$	**MFPLS**	**TMFPLS** H-1	**TMFPLS** H-CV	**MFPCA** LDA	Karim et al.	**SOTA**	**Methods**
CMUsubject16	29	29	86.21	89.66	**100**	89.66	**100**	**100**	[1]
ECG	100	100	85	83	87	88	86	**93**	[2]
EEG	64	64	48.44	54.69	53.12	46.88	**65.63**	62.5	[3]
EEG2	600	600	81.83	68.67	82.67	72.17	**91.33**	77.5	[3]
KickVsPunch	16	10	90	90	60	80	**100**	**100**	[2]
MovementAAL	157	157	67.52	56.69	53.50	61.78	**79.63**	65.61	[4]
NetFlow	803	534	84.64	86.52	85.77	80.90	95	**98**	[2]
Occupancy	41	76	71.05	61.84	59.21	**80.26**	76	67.11	[4]
Ozone	173	173	73.99	73.41	73.41	79.19	**81.5**	75.14	[5]
Wafer	298	896	85.04	87.39	97.99	97.32	**99**	**99**	[2]
WalkVsRun	28	16	**100**	**100**	**100**	**100**	**100**	**100**	[2]

Table 6: Comparison of MFPLS, TMFPLS, and other non-FDA classification methods by their accuracies (in %).
xx : highest accuracy, xx : second-highest accuracy.
[**1**]: Tuncel and Baydogan (2018), [**2**]:Schäfer and Leser (2017), [**3**]:RF , [**4**]: SVM-Poly, [**5**]: DTW

Table 6 shows that, in general, our models are less efficient than Karim et al. (2019) algorithms, but in most cases, they are in the same range of values. Linear discriminant analysis on MFPCA's scores is also competitive. The main difference between the two approaches (MFPCA and MFPLS) as mentioned before is the fact that components are searched with no regard to Y. In 5 out of 11 cases, our proposed models (TMFPLS and MFPLS) reach the highest or the second-highest accuracy. TMFPLS has generally greater accuracy than MFPLS.

TMFPLS H-1 obtains good performances than TMFPLS H-CV for KickVsPunch. This is because TMFPLS H-CV seems to overfit the data when N_{Obs} is small (< 20). This is one of the well-known drawbacks of the decision tree. Tuning hyperparameters is then crucial and can have a huge impact on performances. Moreover, preprocessing and groups' choices driven by prior information on classes should increase our models' performance.

Additional figures and details on estimated models are presented in the supplementary materials.

5 Conclusion and discussion

In this paper, after extending the partial least squares regression to multiple functional predictors, we proposed two classification models of multivariate functional data. The first one relies on the partial least square regression (MFPLS). The second one (TMFPLS) combines the first method with a decision tree. Our models give interpretable results, although a lot of ramifications in the decision tree make insight development difficult. Real applications and simulation studies have been showing our models' competitiveness with existing methods. Especially in multivariate time series classification, where our methods (MFPLS and TMFPLS) have been demonstrating close results on benchmark data with black-box models, in particular with the long short-term memory (LSTM). In addition, as our models are developed for multivariate functional data defined on different domains, it is possible to deal with heterogeneous types of data (e.g. images, time series, etc.). Simulation study has been showing that using images and time series simultaneously for classification increase model performance. It also appeared that in a noisy context, MFPLS outperforms the principal component scores linear discrimination method (MFPCA-LDA). It is worth emphasizing the fact that functional data analysis (FDA) models here use basis expansion techniques. We chose mainly the B-splines basis functions, but another basis can also lead to good results.

This work highlighted one way to deal with non-linearity in classification problems, using a partial least square tree (TMFPLS). In the same idea, clusterwise regression techniques can also be extended for the classification of multivariate functional data, using (Preda & Saporta, 2005) framework and the relationship between linear discriminant analysis with regression.

The relationship between the partial least square of multivariate functional data with its univariate counterparts could offer interesting perspectives.

Indeed, it can be used to introduce sparsity conditions in the partial least square regression of multivariate functional data. The high dimension in functional data analyses is mainly considered through the fact that the predictor (and so the coefficient function) is continuous, hence of infinite dimension. Then, if the purpose is to have a multivariate functional coefficient with sparse univariate function. It may suffice to replace the first PLS by Guan et al. (2022) procedure. This seems feasible as the SFPLS (Sparse FPLS) of Guan et al. (2022) is iterative, and has similar steps with FPLS. The other way to introduce sparsity is by the number of dimensions. In the context where the predictor functions have a high number of dimensions, we can replace the second PLS by ordinary sparse PLS (see e.g Lê Cao, Rossouw, Robert-Granié, and Besse (2008)). This would lead to consider only a few dimensions in the component contribution construction. However, special attention should be on forcing PLS components to be orthogonal, as sparse PLS doesn't always fulfil it. Future work should focus on sparse methods, for classification and regression.

Supplementary information. The supplementary material includes additional figures related to the numerical experiments.

Appendix A Technical arguments

Proof of Proposition 1 Here C-S (1) and C-S (2) stand respectively for Cauchy-Schwartz inequality on integrals and sums.

$$\begin{aligned}
\mathrm{Cov}^2(\langle\langle X, w\rangle\rangle, Y) &= \mathbb{E}^2\left(\langle\langle X, w\rangle\rangle Y\right) \\
&= \left[\sum_{j=1}^{d}\left[\int_{\mathcal{I}_j}\mathbb{E}\left(X^{(j)}(t)Y\right)w^{(j)}(t)dt\right]\right]^2 \\
\text{C-S(1)} \implies \mathrm{Cov}^2(\langle\langle X, w\rangle\rangle, Y) &\leq \left[\sum_{j=1}^{d}\left(\int_{\mathcal{I}_j}\mathbb{E}^2(X^{(j)}(t)Y)dt\right)^{1/2}\right. \\
&\qquad \left.\left(\int_{\mathcal{I}_j}[w^{(j)}(t)]^2dt\right)^{1/2}\right]^2 \\
\text{C-S (2)} \implies \mathrm{Cov}^2(\langle\langle X, w\rangle\rangle, Y) &\leq \left[\sum_{j=1}^{d}\int_{\mathcal{I}_j}\mathbb{E}^2(X^{(j)}(t)Y)dt\right]\underbrace{\left[\sum_{j=1}^{d}\int_{\mathcal{I}_j}[w^{(j)}(t)]^2dt\right]}_{|||w|||^2=1} \\
\mathrm{Cov}^2(\langle\langle X, w\rangle\rangle, Y) &\leq \sum_{j=1}^{d}\int_{\mathcal{I}_j}\mathbb{E}^2(X^{(j)}(t)Y)dt
\end{aligned}$$

Inequalities are equalities if for each i there exists real constants a and a' such as $w^{(j)}(t) = a\mathbb{E}(X^{(i)}(t)Y), t \in \mathcal{I}_j$ and $\left(\int_{\mathcal{I}_j}[w^{(j)}(t)]^2dt\right)^{1/2} = a'\left(\int_{\mathcal{I}_j}\mathbb{E}^2(X^{(j)}(t)Y)dt\right)^{1/2}$.

Notice that the first condition implies the second one. If $w^{(j)}(t) = a\mathbb{E}(X^{(j)}(t)Y)$ then $\left(\int_{\mathcal{I}_j}[w^{(j)}(t)]^2dt\right)^{1/2} = |a|\left(\int_{\mathcal{I}_j}\mathbb{E}^2(X^{(j)}(t)Y)dt\right)^{1/2}$, hence a' equals $|a|$.
For having $|||w||| = 1$, we take $a = \frac{1}{k}$, where $k = \sqrt{\sum_{j=1}^{p}\int_{\mathcal{I}_j}\mathbb{E}^2(X^{(j)}(t)Y)dt}$.
Thus, the solution of (6) is

$$w^{(j)}(t) = \frac{\mathbb{E}(X^{(i)}(t)Y)}{k},\ t \in \mathcal{I}_j. \tag{A1}$$

□

Proof of Proposition 2 $X(t)$ first order residual definition is $X(t) = \xi_1\rho_1(t) + X_{[1]}(t)$, where $X_{[1]}$ satisfy the following proprieties

$$\mathbb{E}(X_{[1]}(t)\xi_1) = 0_{\mathbb{R}^d} \iff \mathbb{E}(X^{(j)}_{[1]}(t)\xi_1) = 0\ t \in \mathcal{I}_j,\ 1 \le j \le d. \tag{A2}$$

Analogously higher-order residuals verify

$$\mathbb{E}(X_{[h]}(t)\xi_h) = 0_{\mathbb{R}^d}\ \forall h \in \mathbb{N}. \tag{A3}$$

Proof of **1.** is by induction, similarly to Tenenhaus, Gauchi, and Ménardo (1995).

The base case verifies, as (A2) is true for each j implies that $\sum_{j=1}^{d}\int_{\mathcal{I}_j}\mathbb{E}\left(\xi_1X^{(j)}_{[1]}(t)\right)w^{(j)}_2(t)dt = 0$, which is equivalent to $\mathbb{E}(\xi_1\xi_2) = 0$.
We show that for any natural integer h, if $\{\xi_1, \xi_2, ..., \xi_h\}$ are orthogonal implies $\{\xi_1, \xi_2, ..., \xi_{h+1}\}$ are also orthogonal.
Assume the induction hypothesis $\mathcal{H}_0$, $\mathcal{H}_0$: $\{\xi_1, \xi_2, ..., \xi_h\}$ *forms an orthogonal system.*

$$\mathbb{E}(\xi_h\xi_{h+1}) = \sum_{j=1}^{d}\int_{\mathcal{I}_j}\mathbb{E}\left(\xi_hX^{(j)}_h(t)\right)w^{(j)}_{h+1}(t)dt$$

$$(A3) \implies \mathbb{E}(\xi_h\xi_{h+1}) = 0$$

$$\mathbb{E}(\xi_{h-1}\xi_{h+1}) = \sum_{j=1}^{d}\int_{\mathcal{I}_j}\mathbb{E}\left(\xi_{h-1}X^{(j)}_h(t)\right)w^{(j)}_{h+1}(t)dt$$

Since $X_{[h-1]}(t) = \rho_h(t)\xi_h + X_h$

$$\implies \mathbb{E}(\xi_{h-1}\xi_{h+1}) = \sum_{j=1}^{d}\int_{\mathcal{I}_j}\underbrace{\mathbb{E}\left(\xi_{h-1}X^{(j)}_{[h-1]}(t)\right)}_{=0 \text{ by } (A3)}dt$$

$$-\rho^{(j)}_h(t)\int_{\mathcal{I}_j}\underbrace{\mathbb{E}\left(\xi_{h-1}\xi_h\right)}_{=0 \text{ by } \mathcal{H}_0}\sum_{j=1}^{d}w^{(j)}_{h+1}(t)dt,$$

$$\text{then } \mathbb{E}(\xi_{h-1}\xi_{h+1}) = 0$$

The same procedure can be used to show that $\mathbb{E}(\xi_j\xi_{h+1}) = 0\ \forall j \le h-2$. Thus $\{\xi_j\}_j$ forms an orthogonal system.
Remain points (from **2** to **5**) are implications of **1.** □

Proof of Lemma 1 For h =1, the lemma gives $v_1(t) = w_1(t)$, as $\xi_1 = \langle\langle X, w_1\rangle\rangle$, base case verifies.
Assume that $\langle\langle X, v_j\rangle\rangle = \xi_j$ is true up to order h ($\forall j \leq h$). We show that it holds for $h+1$.
Recall that,

$$\xi_{h+1} = \langle\langle X_{[h]}, w_{h+1}\rangle\rangle. \tag{A4}$$

Using the third propriety of Proposition 1, we obtain that $X_{[h]}(t) = X(t) - \sum_{i=1}^{h} \rho_i(t)\langle\langle v_i, X\rangle\rangle$.
Then

$$\xi_{h+1} = \langle\langle X, w_{h+1}\rangle\rangle - \sum_{i=1}^{h}\langle\langle v_i, X\rangle\rangle\langle\langle \rho_i, w_{h+1}\rangle\rangle$$

$$\xi_{h+1} = \langle\langle X, w_{h+1}\rangle\rangle - \sum_{i=1}^{h}\langle\langle \rho_i, w_{h+1}\rangle\rangle\langle\langle X, v_i\rangle\rangle$$

$$= \langle\langle X, w_{h+1} - \sum_{i=1}^{h}\langle\langle \rho_i, w_{h+1}\rangle\rangle v_i\rangle\rangle$$

$$\text{Since } v_{h+1}(t) = w_{h+1}(t) - \sum_{i=1}^{h}\langle\langle \rho_i, w_{h+1}\rangle\rangle v_i(t) \implies \xi_{h+1} = \langle\langle X, v_{h+1}\rangle\rangle$$

□

(18) in details From the definition of Y $\mathbb{P}(Y=1) = \mathbb{P}(S_1) + \mathbb{P}(S_2)$, since $\mathbb{P}(S_1 \cap S_2) = 0$.
Define,

$$S_1^+ = \{a_1 = 1, a_2 = 1, a_3 = 0, a_4 = 0\} \quad S_1^- = \{a_1 = -1, a_2 = -1, a_3 = 0, a_4 = 0\}$$
$$S_2^+ = \{a_1 = 0, a_2 = 1, a_3 = 1, a_4 = 0\} \quad S_2^- = \{a_1 = 0, a_2 = -1, a_3 = -1, a_4 = 0\},$$

then $S_1 = S_1^+ \cup S_1^-$, and $S_2 = S_2^+ \cup S_2^-$.

$$\mathbb{P}(S_1^+) = \mathbb{P}(a_4 = 0 | a_3 = 0, a_2 = 1, a_1 = 1)$$
$$\mathbb{P}(a_3 = 0 | a_2 = 1, a_1 = 1)\mathbb{P}(a_2 = 1 | a_1 = 1)\mathbb{P}(a_1 = 1)$$
$$\mathbb{P}(S_1^+) =^{\text{by(17)}} \mathbb{P}(a_4 = 0 | a_3 = 0)$$
$$\mathbb{P}(a_3 = 0 | a_2 = 1)\mathbb{P}(a_2 = 1 | a_1 = 1)\mathbb{P}(a_1 = 1).$$

The same procedure gives $\mathbb{P}(S_1^-)$, $\mathbb{P}(S_2^+)$, and $\mathbb{P}(S_2^-)$. Using conditional probability to Z, we obtain

$$\mathbb{P}(S_1|Z=z) = \mathbb{P}(S_1^+|Z=z) + \mathbb{P}(S_1^-|Z=z) = \frac{1}{2}p_4[z]p_2[z](1-p_3[z])p_1[z]$$

$$\mathbb{P}(S_2|Z=z) = \mathbb{P}(S_2^+|Z=z) + \mathbb{P}(S_2^-|Z=z) = \frac{1}{2}(1-p_4[z])p_3[z](1-p_2[z])(1-\rho_1[z]).$$

Finally, the law of total probability yields to

$$\mathbb{P}(Y=1) = (1-p)\left[\frac{1}{2}p_4[1]p_2[1](1-p_3[1])p_1[1] + \frac{1}{2}(1-p_4[1])p_3[1](1-p_2[1])(1-p_1[1])\right]$$
$$+p\left[\frac{1}{2}p_4[2]p_2[2](1-p_3[2])p_1[2] + \frac{1}{2}(1-p_4[2])p_3[2](1-p_2[2])(1-p_1[2])\right]$$

□

References

Aguilera, A.M., Escabias, M., Preda, C., Saporta, G. (2010). Using basis expansions for estimating functional pls regression: applications with chemometric data. *Chemometrics and Intelligent Laboratory Systems*, *104*(2), 289–305.

Beyaztas, U., & Shang, H.L. (2022). A robust functional partial least squares for scalar-on-multiple-function regression. *Journal of Chemometrics*, e3394.

Blanquero, R., Carrizosa, E., Jiménez-Cordero, A., Martín-Barragán, B. (2019a). Functional-bandwidth kernel for support vector machine with functional data: an alternating optimization algorithm. *European Journal of Operational Research*, *275*(1), 195–207.

Blanquero, R., Carrizosa, E., Jiménez-Cordero, A., Martín-Barragán, B. (2019b). Variable selection in classification for multivariate functional data. *Information Sciences*, *481*, 445–462.

Cardot, H., Ferraty, F., Sarda, P. (1999). Functional linear model. *Statistics & Probability Letters*, *45*(1), 11–22.

Carnegie (0). *Carnegie Mellon University- cmu graphics lab - motion capture library.* http://mocap.cs.cmu.edu/. (Accessed: 2022-05)

Delaigle, A., & Hall, P. (2012). Methodology and theory for partial least squares applied to functional data. *The Annals of Statistics*, *40*(1), 322–352.

Ferraty, F., & Vieu, P. (2003). Curves discrimination: a nonparametric functional approach. *Computational Statistics & Data Analysis*, *44*(1), 161-173. (Special Issue in Honour of Stan Azen: a Birthday Celebration)

Galeano, P., Joseph, E., Lillo, R.E. (2015). The mahalanobis distance for functional data with applications to classification. *Technometrics*, *57*(2), 281–291.

Gardner-Lubbe, S. (2021). Linear discriminant analysis for multiple functional data analysis. *Journal of Applied Statistics*, *48*(11), 1917-1933.

Golovkine, S., Klutchnikoff, N., Patilea, V. (2022). Clustering multivariate functional data using unsupervised binary trees. *Computational Statistics & Data Analysis*, *168*, 107376.

Górecki, T., Krzyśko, M., Wołyński, W. (2015). Classification problems based on regression models for multi-dimensional functional data. *Statistics in Transition new series*, *16*(1).

Guan, T., Lin, Z., Groves, K., Cao, J. (2022). Sparse functional partial least squares regression with a locally sparse slope function. *Statistics and Computing*, *32*(2), 1–11.

Happ, C. (2017). Object-oriented software for functional data. *arXiv preprint arXiv:1707.02129*.

Happ, C., & Greven, S. (2018). Multivariate functional principal component analysis for data observed on different (dimensional) domains. *Journal of the American Statistical Association*, *113*(522), 649–659.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735-1780.

Jacques, J., & Preda, C. (2014). Model-based clustering for multivariate functional data. *Computational Statistics & Data Analysis*, *71*, 92–106.

James, G.M., & Hastie, T.J. (2001). Functional linear discriminant analysis for irregularly sampled curves. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *63*(3), 533–550.

Javed, R., Rahim, M.S.M., Saba, T., Rehman, A. (2020). A comparative study of features selection for skin lesion detection from dermoscopic images. *Network Modeling Analysis in Health Informatics and Bioinformatics*, *9*(1), 1–13.

Jong, S.D. (1993). Pls fits closer than pcr. *Journal of chemometrics*, *7*(6), 551–557.

Karim, F., Majumdar, S., Darabi, H., Chen, S. (2017). Lstm fully convolutional networks for time series classification. *IEEE access*, *6*, 1662–1669.

Karim, F., Majumdar, S., Darabi, H., Harford, S. (2019). Multivariate lstm-fcns for time series classification. *Neural Networks*, *116*, 237–245.

Lê Cao, K.-A., Rossouw, D., Robert-Granié, C., Besse, P. (2008). A sparse pls for variable selection when integrating omics data. *Statistical applications in genetics and molecular biology*, *7*(1).

Li, T., Song, X., Zhang, Y., Zhu, H., Zhu, Z. (2021). Clusterwise functional linear regression models. *Computational Statistics & Data Analysis*, *158*, 107192.

Lichman, M. (2013). *Uci machine learning repository.* http://archive.ics.uci.edu/ml//.

López-Pintado, S., & Romo, J. (2006). Depth-based classification for functional data. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, *72*, 103.

Maturo, F., & Verde, R. (2022). Supervised classification of curves via a combined use of functional data analysis and tree-based methods. *Computational Statistics*, 1–41.

Möller, A., & Gertheiss, J. (2018). A classification tree for functional data. *International workshop on statistical modeling.*

Oleszewski, R. (2012). http://www.cs.cmu.edu/~bobski//.

Pei, W., Dibeklioğlu, H., Tax, D.M., van der Maaten, L. (2017). Multivariate time-series classification using the hidden-unit logistic model. *IEEE transactions on neural networks and learning systems*, *29*(4), 920–931.

Poterie, A., Dupuy, J.-F., Monbet, V., Rouviere, L. (2019). Classification tree algorithm for grouped variables. *Computational Statistics*, *34*(4),

1613–1648.

Preda, C., & Saporta, G. (2002). Régression pls sur un processus stochastique. *Revue de statistique appliquée*, *50*(2), 27–45.

Preda, C., & Saporta, G. (2005). Clusterwise pls regression on a stochastic process. *Computational Statistics & Data Analysis*, *49*(1), 99–108.

Preda, C., Saporta, G., Lévéder, C. (2007). Pls classification of functional data. *Computational Statistics*, *22*(2), 223–235.

Ramsey, J.O., & Silverman, B.W. (2005). *Functional data analysis* (2nd ed.). Springer-Verlag.

Ribeiro Jr, P.J., Diggle, P.J., et al. (2001). geor: a package for geostatistical analysis. *R news*, *1*(2), 14–18.

Rossi, F., & Villa, N. (2006). Support vector machine for functional data classification. *Neurocomputing*, *69*(7-9), 730–742.

Saikhu, A., Arifin, A.Z., Fatichah, C. (2019). Correlation and symmetrical uncertainty-based feature selection for multivariate time series classification. *International Journal of Intelligent Engineering and System*, *12*(3), 129–137.

Schäfer, P., & Leser, U. (2017). Multivariate time series classification with weasel+ muse. *arXiv preprint arXiv:1711.11343*.

Sübakan, Y.C., Kurt, B., Cemgil, A.T., Sankur, B. (2014). Probabilistic sequence clustering with spectral learning. *Digital Signal Processing*, *29*, 1–19.

Tenenhaus, M., Gauchi, J.-P., Ménardo, C. (1995). Régression pls et applications. *Revue de statistique appliquée*, *43*(1), 7–63.

Tuncel, K.S., & Baydogan, M.G. (2018). Autoregressive forests for multivariate time series modeling. *Pattern recognition*, *73*, 202–215.

Yao, F., Fu, Y., Lee, T.C. (2011). Functional mixture regression. *Biostatistics*, *12*(2), 341–353.