AVERTISSEMENT

LIENS

Code la Propriété Intellectuelle – Articles L. 122-4 et L. 335-1 à L. 335-10
Loi n° 92-597 du 1er juillet 1992, publiée au *Journal Officiel* du 2 juillet 1992

http://www.cfcopies.com/V2/leg/leg-droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse Capitole

École doctorale : **Mathématiques, Informatique, Télécommunication de Toulouse**

Présentée et soutenue par

LEJEUNE Clément

le 6 Mai 2021

Multivariate Functional Data : Geometric Features Extraction and Sparse Learning of Dynamics

Discipline : Informatique

Spécialité : Intelligence Artificielle

Unité de recherche : **Institut de Recherche en Informatique de Toulouse (UMR 5505)**

Directeurs de thèse :

- Mme, Josiane, MOTHE, Professeure, Université de Toulouse II
- Mr, Olivier, TESTE, Professeur, Université de Toulouse II

JURY

Rapporteurs	Mr, Benjamin, PIWOWARSKI, Chargé de Recherche, CNRS Mr, Pierre, GANÇARSKI, Professeur, Université de Strasbourg
Suffragants	Mme, Anne, RUIZ-GAZEN, Professeure, Université de Toulouse Capitole Mr, Julien, VELCIN, Professeur, Université de Lyon II
Directeur(s) de thèse	Mme, Josiane, MOTHE, Professeure, Université de Toulouse II Mr, Olivier, TESTE, Professeur, Université de Toulouse II

Remerciements

Tout d'abord, je tiens à remercier mes directeurs de thèse, Olivier Teste et Josiane Mothe, de m'avoir donné l'opportunité de réaliser cette thèse. Je les remercie pour leur accompagnement le long de mes trois premières années de recherche, notamment Josiane pour ses précieux conseils, de présentation de mes travaux, et d'écriture lorsqu'il s'agissait de rédiger le premier article. Je remercie chaleuresement Olivier qui, de façon générale, a très souvent su trouver une solution en ma faveur et s'est montré encourageant du début à la fin.

Je tiens également à remercier mon encadrant insutriel, Adil Soubki, sans qui cette thèse de doctorat n'aurait pas pu exister. Il m'a laissé une grande autonomie, a su me recadrer sur ma problématique générale quand il le fallait et a su me faire prendre du recul sur mes les idées que j'ai proposées. Je remercie aussi les ingénieurs-experts des essais en vol de la société Airbus, Jean-Marc Prangère et Amine Taourchi, de m'avoir guidé lors du développement de mes idées sur les cas d'étude qu'ils ont proposés.

Je remercie les membres du jury, Benjamin Piwowarski et Pierre Gançarski, d'avoir accepté de rapporter mon manuscrit de thèse ainsi que Julien Velcin et Anne Ruiz-Gazen pour l'avoir examiné.

Je remercie tous mes collègues de l'IRIT, qui m'ont supporté et ont largement participé au déroulement de cette thèse dans la bonne humeur: Nabil (avec

qui j'ai échangé sur de nombreux sujets techniques ou non), Nathalie, Zia et Faneva (qui m'ont laissé investir leur bureau pour que je puisse boire mon café en partagant une discussion...), sans oublier Wafa, Oihana, Olivier, Tianyi, Daria, Omar. Je veux aussi remercier Lionel Rigal, mon manager côté Airbus, qui m'a accueilli dans son équipe. Aussi, je remercie l'équipe, notamment mes collègues de bureau: Benjamin, Gaelle, Florence, David, Vincent, Bernard, Guy et Venkat.

Je remercie mes parents et ma soeur de m'avoir continuement soutenu depuis le début de mes études. Enfin, je remercie mes amis (les toulousains et les parisiens !) de m'avoir soutenu (et supporté). Ils m'ont permis de maintenir un certain équilibre et de garder les pieds sur terre durant ces trois années.

Abstract

A multivariate time series is a time-indexed sequence of multidimensional samples. Such a kind of data appears in many fields since they are the observation of dynamic systems (*e.g.* mechanics, biology). Hence, the constituting variables of a multivariate time series are often related to each other. This multidimensionality renders the analysis of the phenomenon underlying the data more complex than with univariate time series.

In this thesis, we deal with datasets made up of multiple multivariate time series. In particular, (i) we are concerned with the detection of abnormal phenomena, which is commonly referred as *outlier* or *anomaly* detection. Furthermore, for a phenomenon being studied, which might be an outlier, the model of the underlying dynamics can provide indepth knowledge on it. Thus, (ii) we also interest in discovering the model of the dynamics in a data-driven manner. To address (i) and (ii), we have made two contributions wherein each one of them, a time series is represented as a function over time.

Our first contribution deals with the detection of outliers in a dataset of multivariate time series. We addressed the problem in a functional data perspective. Functional data analysis is a statistical framework to represent sequences of continuous variables, whose index is the discretization of a continuous variable (*e.g.* time), as functions. Our contribution builds upon this framework. We observed that, due to atypical relationships between the

variables of a time series, the outlyingness can result in its curve shape. To highlight this shape outlyingness, we proposed to aggregate the variables of a multivariate time series in several geometric manners. Then, we used the output representation, provided by these aggregation operators, of the initial time series, as input of existing outlier detection algorithms. We empirically showed that our approach outperforms state-of-the-art methods.

Our second contribution tackles the data-driven discovery of a deterministic model underlying the dynamics between variables of a multivariate time series. We focused on the case where this (unknown) model is a system of ordinary differential equations whose solution is the function representing the time series itself. We proposed a multi-task learning algorithm to discover such a model. Each task aims at learning a single differential equation which can be coupled to the other ones. We addressed the multi-task aspect with a regularizer that enforces both sparsity within tasks and similarity between tasks. In addition, contrary to state-of-the-art multi-task regularizers, ours is nonconvex and thus provides accurate estimate of the model parameters. We empirically showed on a benchmark of systems of ordinary differential equations that learning in a multi-task way with nonconvex sparsity outperforms state-of-the-art approaches in terms of bias and reconstruction error from the model learnt.

Contents

List of Figures

List of Tables

Abbreviations

FDA Functional Data Analysis

SODE System of Ordinay Differential Equations

MTL Multi-Task Learning

SCAD Smoothly Clipped Absolute Deviation

GIST Generative Iterative Sequential Thresholding

DOC Damped Oscillator with Cubic dynamic

LV Lotka-Volterra

LAT Lorenz Attractor

Mathematical notations

- $\mathbb{R}^d$: the set of real valued d-dimensional column vectors
- $\mathbb{R}^{n\times d}$: the set of real valued $n \times d$ matrices
- a: an element of $\mathbb{R}$
- $\mathbf{a} = \Big(a_1, \ldots, a_d\Big)^\top$: an element of $\mathbb{R}^d$
- $\boldsymbol{A} = \Big(\mathbf{a}_1, \ldots, \mathbf{a}_n\Big)$: an element of $\mathbb{R}^{n\times d}$
- $\mathbf{a}_{k,\bullet}$: the k-th row of $\boldsymbol{A}$
- $\mathbf{a}_{\bullet,l}$: the l-th column of $\boldsymbol{A}$
- $\langle \mathbf{a}, \mathbf{b}\rangle$: the standard (Euclidean) inner product between $\mathbf{a}$ and $\mathbf{b}$
- $\|\mathbf{a}\|_p = (\sum_k |a_k|^p)^{1/p}$: the ℓ_p norm of $\mathbf{a} \in \mathbb{R}^d$ with $p \geq 1$
- $\|\boldsymbol{A}\|_{p,q} = \Big(\sum_k \|\mathbf{a}_{k\bullet}\|_p^q\Big)^{1/q}$: the $\ell_{p,q}$ norm of $\boldsymbol{A}$ with $p, q \geq 1$
- $\mathcal{T}$: a closed interval of $\mathbb{R}$
- $\tilde{\mathcal{T}} = \Big\{t_1, \ldots, t_m\Big\}$: an arbitrary discretization of $\mathcal{T}$ with m values
- $\mathbf{x}(t)$: the value of the function $\mathbf{x}$ at t. If $\mathbf{x}(t) \in \mathbb{R}^{d=1}$, then $\mathbf{x}(t) = x(t)$
- $\tilde{\mathbf{x}}$: an approximation of the function $\mathbf{x}$
- $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$: the Gaussian distribution with mean $\boldsymbol{\mu}$ and (positive-

definite) covariance $\mathbf{\Sigma}$

- $\mathcal{U}(\mathbf{a}, \mathbf{b})$: the uniform distribution in the $\mathbf{a} \times \mathbf{b}$ rectangle with $a_i > b_i$

Chapter I

General introduction

Abstract.

In this chapter, we first present the industrial context that has motivated our research exposed in this thesis. For that, we define what a flight test is, and introduce the issues involved by the analysis of time series generated by flight tests, *i.e.* the detection and understanding of abnormal dynamic behaviors. Secondly, we both define the notions of time series and outliers, which we encompass in the functional data framework. Thirdly, we sum up our two contributions: the first one addresses the detection of outliers in functional data, and the second one tackles the discovery of systems of differential equations from a time series.

I.1 Industrial context

I.1.1 What is a flight test ?

In civil aviation, to be allowed to operate, an aircraft must be certified by a government certifying agency (*e.g.* European Aviation Safety Agency) independent from the manufacturer (*e.g.* Airbus). To achieve the certification level, the manufacturer needs to perform multiple flight tests to ensure that all the safety and performance requirements are met: this is referred as the *flight test campaign* as illustrated in Figure I.1. The certification can vary from a whole aircraft to a single system. During a flight test, a large number of embedded functions in the aircraft is used and most of them require the monitoring and/or the analysis by flight test experts in-flight (online) as well as on ground (offline). These functions involve physical flight parameters that are sampled by specific sensors along time: such kind of measurements gives rise to *time series* data. These time series depict events like engine start, compression pumping engine or an automatic approach, to name a few. For a specific system and a set of flight parameters given by flight test experts, the time series relative to the flight parameters are often correlated with each other. Indeed, their behavior are induced by a control logic and constrained by a set of physical laws (*e.g.* from flight mechanics, combustion laws) which, depending on the system of interest, are fully or partially understood.

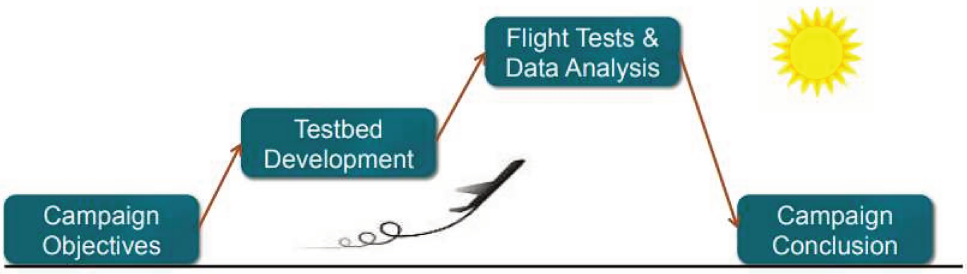

Figure I.1: High-level flow-chart description of a flight test campaign (from [Bunge *et al.* 2015])

I.1.2 Why analyzing flight test data ?

Flight test data analysis aims both at fixing design problems and validating the aircraft performances. Once these goals have been achieved, the manufacturer has to write documentation for government certification and customer acceptance. Such a documentation entails, for instance, outlines, tables and flow-charts of aircraft performances under nominal operating conditions and emergency procedures as well. As operating conditions can vary with respect to external conditions (*e.g.* weather), every flight is different. Therefore the documentation cannot cover every nominal operating condition but a reference domain of known ones from which the pilot can easily infer unknown ones. "Interpolating" means adapting and inferring the aircraft behavior to a situation that does not exactly correspond to one described in the documentation but is *a priori* close to it. For that reason, it is necessary to understand, under various operating conditions, the behavior of the aircraft through flight test data. Furthermore, such an understanding helps flight test engineers to classify the nominal operating conditions and thereby to detect abnormal/unexpected aircraft behaviours.

I.1.3 Multiple-flight data analysis

When the behavior of the flight parameters is known by flight test engineers, multiple-flight data analysis is in principle fast. However, on the first hand, to understand both the similarities and specificities of multiple flights, the data analyses must be done in a short time (*e.g.* propulsive system engineers can be asked to analyze 200 flight test data in four days). On the other hand, when there is a lack of knowledge about the behavior of some flight parameters involved in a specific system, their analysis result harder and longer. As an example, propulsive systems are designed by engine manufacturers who cannot share information on the engine operating logic, making the analysis of engine parameters harder. Taking benefit from the data variability of multiple flights can help to understand the operating modes of such a system. Hence, there is a need of interpretable data-driven based methods to

perform multiple flight analyses. Such analyses can reveal, for instance, an abnormal behavior of a given system which thus requires some correction on ground. Since the flight test data take the form of time series (see example in Figure I.2), in this thesis we propose several time series based methods and algorithms which aim to extract knowledge *e.g.* to assist flight test analysts.

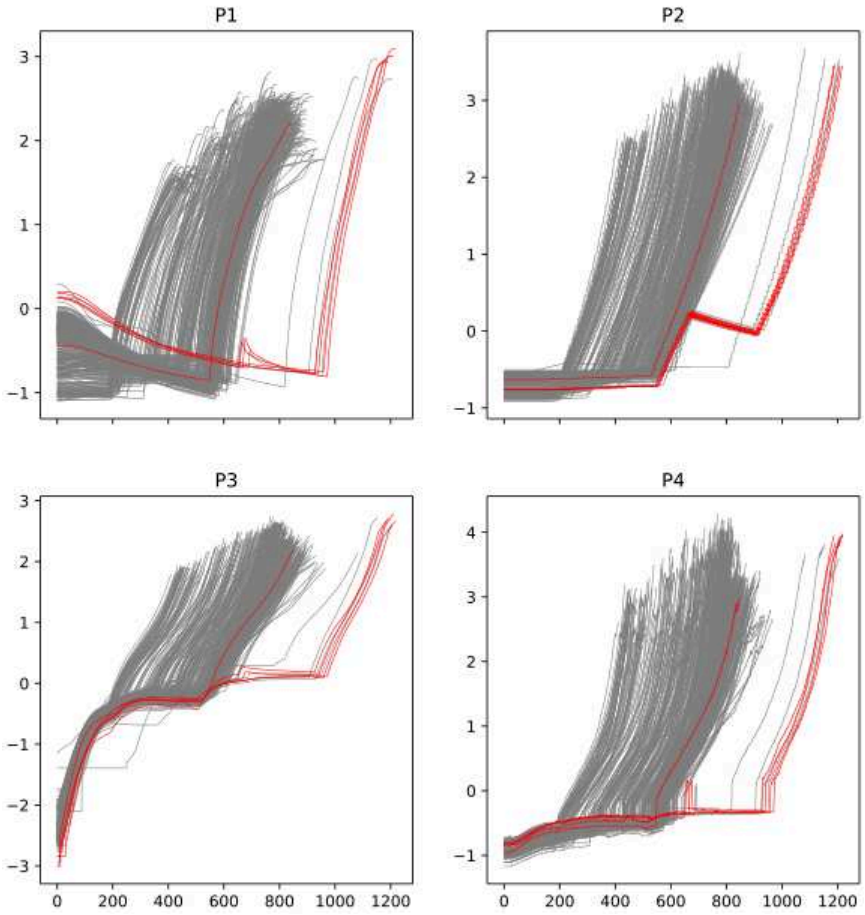

Figure I.2: Example of an Airbus dataset of multivariate time series relating the behavior, along time (vertical axis), of a system involving four parameters $(P_1, \ldots, P_4)$. Grey curves correspond to normal behaviors whereas red ones correspond to abnormal behaviors. For data privacy, the vertical axes have been scaled.

I.2 Scientific motivations

I.2.1 General definition of time series

In many fields (*e.g.* engineering, chemometrics, finance), measurement systems generate data under the form of time series [Dos Santos *et al.* 2016, Boulfani *et al.* 2020, Ramsay & Silverman 2006]. A time series is a sequence of random objects x indexed by some time steps $t \in \tilde{\mathcal{T}} = \left\{t_1, \ldots, t_m\right\}$, either naturally discrete (*e.g.* month, year), either resulting from a discretization of a time-continuous variable, *i.e.* a sampling process. When the indexed objects are univariate, and lie in $\mathbb{R}$, the dataset is a collection of real-valued univariate time series (see example in Figure I.3). By extension, when the indexed objects are multivariate, and lie in $\mathbb{R}^p$ (*i.e.* x is vector-valued of dimension p), the dataset is a collection of multivariate time series (see examples in Figure I.4 with $p = 2$), [Lafabregue *et al.* 2019]. When substituting $\mathbb{R}$ with $\mathbb{N}$, one can have time series of categorical data. Hence, the notion of time series can be easily extended to non-numerical types of data like text [Baril *et al.* 2020].

I.2.2 The variability of univariate time series

In this thesis, we deal with datasets of time series whose indexes are the discretization of a continuous-variable. More specifically, we consider the time series as random realizations of functions depending on a continuous variable of $\mathcal{T} \subset \mathbb{R}$. Representing time series as functions is the building block of Functional Data Analysis (FDA). Such a representation aims to extract features containing information on the functional variability of the time series and can serve for dimensionality reduction, smoothing, feature extraction, prediction as well as visualization [Ramsay & Silverman 2006, Ferraty & Vieu 2006]. In Figure I.3, we give an example of univariate functional data generated by an arbitrary function.

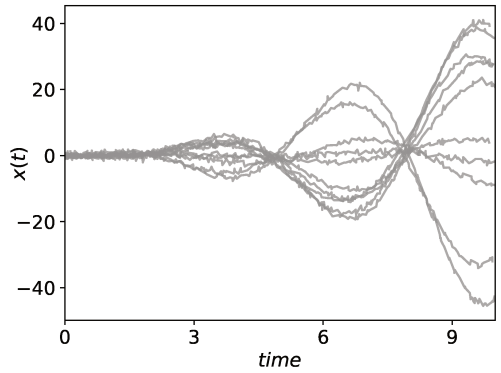

Figure I.3: Synthetic dataset of $n = 10$ univariate functional time series, with time steps in $\tilde{\mathcal{T}} = \left\{t_1 = 0, \ldots, t_{1000} = 10\right\}$, generated by $x(t) = t\sin(t) + t^2\cos(t) + \epsilon$ for $t \in \mathcal{T} = [0, 10]$, where ϵ is a random variable with distribution $\mathcal{N}(0, \sigma = 0.25)$.

I.2.3 The second variability of multivariate time series

Although, the only difference between univariate and multivariate time series is their dimension at each time step, in multivariate time series the p variables can be correlated between them along time. This correlation can result from a mechanism that is deterministic (*e.g.* a system of differential equations) or stochastic (*e.g.* a vector auto-regressive model, stochastic differential equations). Therefore, in addition to the temporal variability of each variable individually, multivariate time series have a variability across its p variables.

Example I.1. In Figure I.4, we illustrate the correlation between the variables with a synthetic dataset of bivariate time series generated as follows:

- For each sample $i \in \{1, \ldots, n = 10\}$, the time steps $\tilde{\mathcal{T}}_i = \{t_{1i}, \ldots, t_{m_i}\}$ are randomly chosen from $\mathcal{U}(0, 10)$ with $m_i = 250$.
- The first variable for sample i is generated as $x_{1i}(t) = t\sin(t) + t^2\cos(t) + \epsilon$ where ϵ is a random under the distribution $\mathcal{N}(0, \sigma = 0.25)$ and for $t \in \tilde{\mathcal{T}}_i$.
- The second variable is generated according to $x_{2i} = A_i \sin(2\pi f_0 t) + \epsilon$

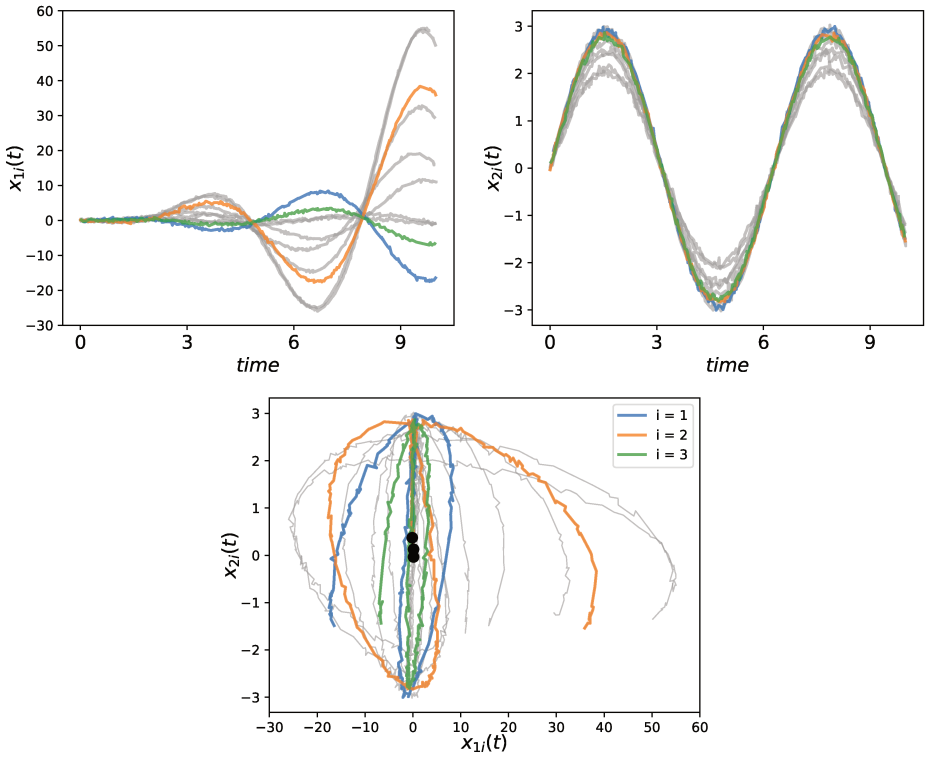

Figure I.4: Synthetic dataset of $n = 10$ bivariate time series dataset, generated according to Example I.1, among which three were randomly chosen and colored not to overload the bottom plot. Top-left: the samples of the variable x_{1i}. Top-right: the samples of the variable x_{2i}. Bottom: the three $(i = 1, 2, 3)$ colored samples plotted in the space of variables to visualize the correlation between x_{1i} and x_{2i} along t.

where ϵ is independently generated as in x_{2i}, and A_i is drawn from $\mathcal{U}(2, 3)$.

Simple algebraic manipulations show that x_1 and x_2 are related by $x_1(t) = tx_2(t) + t^2 x_2(t + \frac{\pi}{2})$. Thus there is a nonlinear correlation between x_1 and x_2. Furthermore, one can show, by computing analytically $\dot{x}_1$ and $\dot{x}_2$, that the non-noisy version of the functions x_1 and x_2 (without ϵ) is the solution of the following system of ordinary differential equations:

$$\begin{cases} \dot{x}_1(t) = A(1 - t^2)x_1(t) + 3Atx_1(t + \frac{\pi}{2}) \\ \dot{x}_2(t) = Ax_1(t + \frac{\pi}{2}) \end{cases} \tag{I.1}$$

with $A \neq 0$ and the initial condition $x_1(0) = x_2(0) = 0$.

This example serves as an illustration of a (functional data based) generating process of multivariate time series. Of course, in practice the generating process is unknown but it shows how the correlation between the p variables along time can be underlied. This example also motivates to retrieve a functional mechanism that underlies a time series dataset, this is the first motivation of FDA. Example I.1 illustrates how a deterministic model can characterize the correlation between the variables of a multivariate time series. Accessing to the system of differential equations underlying a real dataset can be of huge interest for engineers in real life applications. For instance it can serve as a simulation model or as a representation of the phenomenon dynamics. For real datasets, such kind of model is often unknown by the practitioner. Recently the machine learning community has proposed data driven methods to retrieve the model underlying the dynamics of a noisy dataset of time series [Brunton *et al.* 2016, Schaeffer & McCalla 2017, Schaeffer 2017, Mangan *et al.* 2017].

In Chapter III, we interest in the data-driven discovery, in closed form, of a system of differential equations that models the dynamics of a multivariate time series. Such an approach is built upon the recent framework of *sparse identification on nonlinear dynamics* [Brunton *et al.* 2016].

I.2.4 General definition of outliers

The concept of outlier has been defined in [Hawkins 1980] as "an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism". This definition is not specific to time series and is now well accepted. According to the definition in [Aggarwal & Yu 2001], an outlier is defined as a data point that is highly different from the others, based on some *measure*. Therefore, outliers can be seen as observations of a dataset that do not follow an expected

behavior. Depending on the context, outliers are also referred as *anomalies* in [Erfani *et al.* 2016, Chandola *et al.* 2009, Rousseeuw & Hubert 2018, Liu *et al.* 2008], *novelty* in [Markou & Singh 2003, Pimentel *et al.* 2014] or more rarely *singularities*. In certain cases, outliers are of analysis interest since they can reveal a design fault in the system. For instance, detecting outliers can help engineers working on a system requiring a high safety level (*e.g.* aircraft) to detect unexpected system operating modes during the system test phase.

I.2.5 Outliers in time series

Outlier in time series has different meanings in the literature. In many papers, it is assumed that the dataset is made up of a *single* multivariate or univariate time series. Hence, detecting outliers in such a dataset amounts to find points of the time series that deviates from a regular pattern in a given sense. Oppositely, a dataset can entail *multiple* time series. Detecting outliers in such dataset amounts to find the time series that deviate from the other ones. In this thesis, we deal with datasets containing multiple multivariate time series.

I.2.5.1 Outliers in a univariate time series dataset

In a dataset of univariate time series, a sample can result as an outlier in several ways. Assuming that a function u of a temporal continuous variable, $t \in \mathcal{T}$ (see example Figure I.3) underlies the dataset, according to the taxonomy of [Hubert *et al.* 2015], the deviation of an outlier can be:

- horizontal-shift: the time series was generated by the same process as inliers up to a time translation, *i.e.* $u_{horizontal} = u(t - \tau)$ where τ is random with nonzero mean.
- in the magnitude: the time series was generated by the same process as inliers up to a magnitude shift, *i.e.* $u_{magnitude} = u(t) + a$ where a is random with nonzero mean.

- in shape: the time series was generated by a different process as inliers and does not stand out at any time point.

In Figure I.5, we give an illustration for each of these types of outlier.

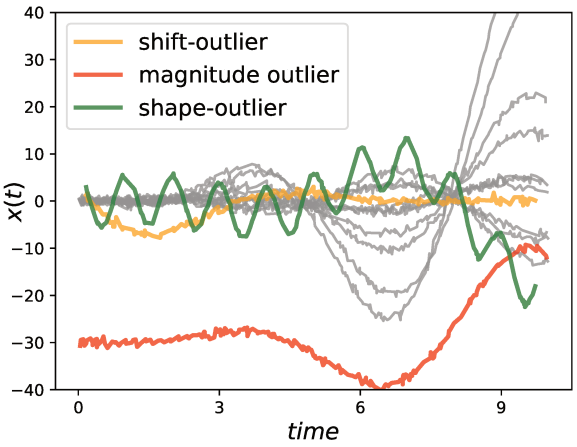

Figure I.5: Same synthetic dataset than in Figure I.3 (grey) with outliers (not grey) of three types. The shift outlier, in yellow, was generated by $u_{hoz-shift}$ with τ drawn from $\mathcal{N}(6, 0.1)$. The magnitude outlier, in red, was generated by $u_{magnitude}$ with a drawn from $\mathcal{N}(-30, 0.1)$. The shape outlier, in green, was generated according to $u(t) + \sin(2\pi f_0 t)$ with $f_0 = 7$.

I.2.5.2 Outliers in a multivariate time series dataset

In order to reduce the analysis of a dataset of p-dimensional multivariate time series, one can split it into p datasets of univariate time series. This boils down to omit the correlation between the p variables of each time series. However, such a correlation can be of great matter to detect outliers. Indeed, when considering each $x_1, \ldots, x_p$ independently, the deviation of an outlier in, say x_k, can be irrelevant compared to its deviation in the joint representation of, say x_k and $x_{k'}$. This is mentioned in [Hubert *et al.* 2015, section 2.2] as "None of the outliers [...] will necessarily be outlying in one of the marginals, hence it is not sufficient to analyze the components separately" and illustrated in Figure I.4. For instance, in Figure I.4 the samples $i = 1$ (blue) and $i = 3$ (green) are quite similar in their individual variable

representation (top) but their distinction is much more obvious in their joint variable representation (bottom).

Then, comes the question of how to consider the p variables jointly across the time variable. To answer such a question, in Chapter II, we propose to represent multivariate time series as multidimensional path from which we extract geometric features that highlight their outlyingness.

I.3 Contributions

To tackle the detection of outliers in multivariate time series, we made two contributions. The common idea of our two contributions is to represent time series as functions over a time-continuous variable. In the first one, a multivariate time series is represented as a multivariate path (or a trajectory) from which shape-based features are extracted, under the form of a univariate time series, and then input into an outlier detection algorithm. In this manner, multiple kinds of outliers can be detected. In our second contribution, a multivariate time series is seen as the solution of an unknown system of differential equations that is recovered by a multi-task learning algorithm. Our algorithm returns the analytic form of the differential equation which is useful to model and understand the underlying mechanism of a multivariate time series.

I.3.1 Overview

I.3.1.1 Outlier detection with shape-based features

Our first contribution is to propose a shape-based feature extraction from multivariate time series and is based on the multivariate functional data framework [Lejeune *et al.* 2020a, Lejeune *et al.* 2020b]. Multivariate functional data refer to a population of multivariate functions generated by a system involving dynamic variables depending on a continuous variable (*e.g.* time). Outlier detection in such a context is a challenging problem because, as we mentioned along Chapter I, both the individual behavior of the variables and the dynamic correlation between them, are important. To identify the outliers, recent work has focused on multivariate functional depth [Claeskens *et al.* 2014, Dai & Genton 2019, Kuhnt & Rehage 2016] which assigns an outlyingness score to each variable independently and then sums over the scores. However, such an approach fails when the outlyingness manifests in the curve shape rather than in the curve magnitude. To remedy it, the curve geometry has to be considered across its variables, jointly, rather in each

variable separately. In Chapter II, we propose to detect outliers in multivariate functional data by aggregating the p dimensions with mapping functions from differential geometry. Our proposal can be seen as a functional-shape-based features extraction enabling to highlight the outlyingness of a curve with interpretability. Then, we used the output representation of these mappings as input of existing outlier detection algorithms. We conducted an experimental study on real and synthetic datasets and compared the proposed method with functional-depth-based ones. One of the real datasets is from Airbus flight test department, thus not publicly available. The results demonstrate that our proposal can outperform the functional-depth-based methods. Moreover, in contrast with the baseline methods, our method is efficient for a large range of outlier proportions.

I.3.1.2 Learning a system of differential equations from a multivariate time series

Our second contribution is to learn the governing equation, in closed form, that models the dynamic relationship between variables of a multivariate time series [Lejeune *et al.* 2021]. Since a multivariate time series can be seen as a vector-valued function over time, the governing equation takes the form of System of Ordinary Differential Equations (SODE). A SODE brings an accurate understanding on the corresponding dynamic phenomenon. It consists in multiple equations, as in Example I.1, where each one relates the time derivative of a single variable to several variables. A variable can appear in multiple equations, making the equations potentially depending on one to the other. While in some cases the SODE can be written thanks to expert or phenomenon knowledge, in most of the cases, the SODE is unknown. Data-driven approaches to automatically discover the underlying SODE have been made possible thanks to sensor technology which can collect large amount of data [Brunton *et al.* 2016, Long *et al.* 2018, Schaeffer *et al.* 2018, Bhat & Rawat 2019]. Nevertheless, state of the art approaches are based on single-task learning that means each component of the SODE is learned independently [Argyriou *et al.* 2008] with sparse linear regression, where the

sparsity is reflected in the convex $\ell_{1,1}$ penalty. This leads to SODEs that weakly identify the underlying phenomenons since in reality, equations are related to each others. Moreover, it turns out that the convexity of the penalty involved in the learning criterion results in a SODE which is biased with respect to the true one [Fan & Li 2001, Zhang 2010]. In Chapter III, we develop a Multi-Task Learning (MTL) approach, where a task refers to the discovery of a single equation of the SODE, to learn the behavior of the dynamic system more accurately. Discovering a SODE is a hard problem since real data are usually noisy and state-variables are often underlied by a nonlinear dynamic. The nonconvex MTL approach we have proposed improves the SODE identification by leveraging from a nonconvex sparse matrix penalty that both considers the coupling within the SODE and solves the bias issue. Results from several numerical experiments on a reference benchmark of SODEs confirm that, compared with single-task learning, MTL is better to recover the underlying form of the SODE and that the nonconvexity enables more accurate estimate of it.

Chapter II

Shape-based outlier detection in multivariate functional data

Abstract.

In this chapter, we expose our first contribution which deals with the detection of outliers in multivariate functional data analysis (*i.e.* multivariate time series). State of the art approaches are recent and consider that the outlyingness resides in each variable of the time series. However, this is limiting since the outlyingness can reside in the relationship between the variables. We address this limitation by considering the correlation between the variables, implicitly, through the curve shape. Specifically, we propose to aggregate the variables in several geometric manners thus representing the samples to univariate functional data. The outliers are then detected from such new representation. Numerical experiments on real (public and Airbus ones) and synthetic datasets, show that our approach outperforms state of the art in most situations.

II.1 Introduction

Functional data analysis (FDA) is a branch of modern statistics, the principle of which is the representation of high-dimensional measurement vectors through functions (see [Ramsay & Silverman 2006, Ferraty & Vieu 2006] for a practical and theoretical introduction to FDA). They appear in various fields, such as biology, engineering, or medicine, where different sources of measurements are recorded. As a real example of such data, we can consider a longitudinal study for analyzing the height of a human population, such as the Berkeley growth study [Tuddenham & Snyder 1954], in which a physiological parameter or variable is measured for all subjects at various time instants. Such data can be seen as realizations of a univariate function depending on time. Although a continuous function depending on a single continuous variable (*e.g.* time, wavelength, or frequency) underlies the data, it is finely discretized, resulting in high-dimensional vectors. Such data are referred to as functional data. Regarding data as functions enables recovering the true nature of the process underlying the function that generated the data. It also provides a smooth representation of the initial curves, which can be affected by measurement noise. Moreover, the FDA framework enables the handling of curves that are irregularly sampled or sampled on grids of different sizes, where a grid refers to the discretization of a closed interval in which the continuous variable lies. This is achieved by evaluating the resulting functions on a common and arbitrary grid.

Specifically, when a single variable is recorded at each observation point (as in the previous example), that is, the underlying function $x(t) \in \mathbb{R}$, where $t \in \mathcal{T} \subset \mathbb{R}$, the resulting data are called univariate functional data. More generally, when p (possibly correlated) variables are simultaneously recorded at each observation point, that is, $\mathbf{x}(t) = \left[x_1(t), \ldots, x_p(t)\right] \in \mathbb{R}^p$, these data are called multivariate functional data. In the example, if weight were measured in addition to height, these data would result as realizations of a multivariate function, in this case bivariate.

A typical task in FDA is outlier detection, which has several applications, for instance, in biology (to determine abnormal gene expression levels in time-course micro-array data [Arribas-Gil & Romo 2014, Hubert *et al.* 2015]), in chemometrics (to determine the nature of an active substance produced by a chemical process based on near-infrared spectra data [Hubert *et al.* 2015]), or in air pollution studies (to detect highly contaminated locations in urban areas [Torres *et al.* 2011]). In these fields, the data are typically functional and exhibit outlying behavior. Moreover, several parameters should be simultaneously recorded to accurately understand the studied process. Hence, outlier-detection methods should be specifically designed for multivariate functional data. When the variables are cautiously selected by a domain expert, the outlying behavior can be detected through the potential correlation between them.

The correlation between the p variables is important in multivariate functional data because it can reveal an outlying behavior of the underlying process, as discussed in [Hubert *et al.* 2015]. In Figure II.1, we show a multivariate functional dataset contaminated by one outlier whose variables are non-linearly correlated where: in (a) and (b), the variables x_1 and x_2 are plotted independently, whereas in (c) and (d), they are plotted one versus the other, thus highlighting their correlation along t. Thus, independently analyzing each variable implies that the potential correlation between the variables is not considered,

According to the definition by [Aggarwal & Yu 2001], an outlier is defined as a data point that is highly different from the others, based on some *measure*. Such a point often contains useful information regarding the abnormal behavior of the system described by the data. Moreover, if the data dimension is high, the data are likely to be more scattered in the space (*i.e.* curse of dimensionality), and therefore, the probability that the outliers are scattered is higher. Hence, outlier-detection is inclined to the curse of dimensionality as other classification tasks that assume well-balanced classes. However, regarding some typical algorithms for classification (*e.g.* logistic regression) and clustering (*e.g.* K-means and mean-shift), the rarity and scattering of

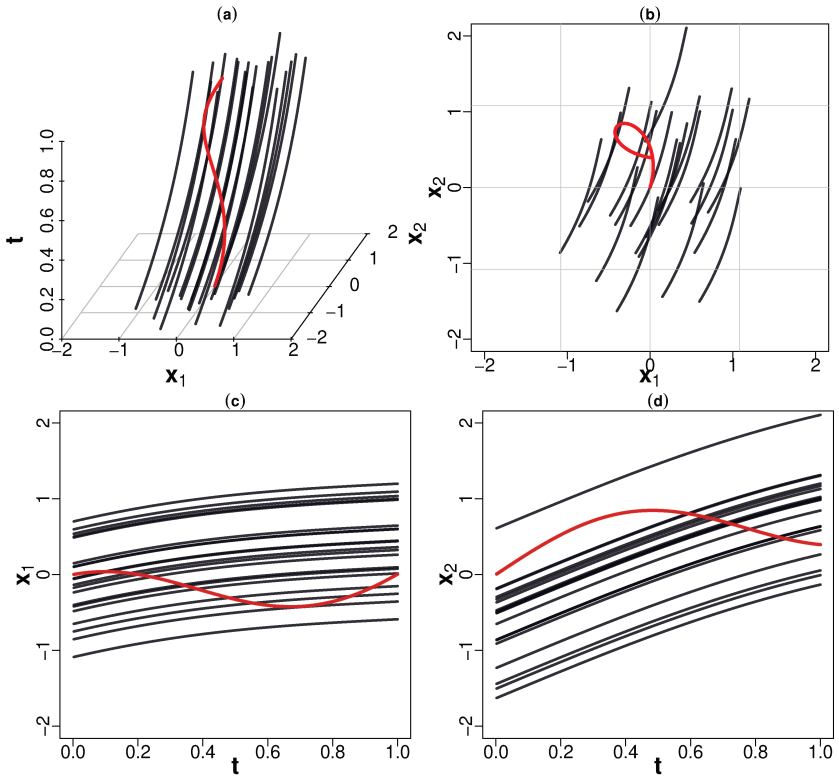

Figure II.1: (see the color version for clarity). (a) A dataset of 21 bivariate curves, with variables $x_{i1}(t), x_{i2}(t)$, $i = 1...21$, is plotted along the variable and the $t \in [0, 1]$ axes. There are 20 inliers (black) and one shape outlier (red). (b) The dataset is projected along the t axis; the red curve clearly shows an outlying relationship between its variables, resulting in a different shape. This is the "view" adopted in this study. In (c) and (d), the variables x_{i1} and x_{i2} are plotted as two univariate functions with respect to t. Determining the degree of difference of the red curve without computing derived functions (*e.g.* derivative(s)) is not simple. Moreover, if the dataset is very large, visual detection is difficult.

outliers may render these algorithms inefficient for outlier detection, owing to the well-known *class imbalance problem* [Japkowicz & Stephen 2002].

Previous work on outlier detection in functional data primarily focused on the univariate case [Fraiman & Muniz 2001, Cuevas *et al.* 2006, López-pintado & Romo 2009], whereas the multivariate case is more recent [Claeskens *et al.* 2014, Ieva & Paganoni 2013, López-pintado *et al.* 2014, Hubert *et al.* 2015, Kuhnt & Rehage 2016, Dai & Genton 2019]. Multivariate functional outliers can be characterized by deviations in the correlation between the variables $x_1(t), ..., x_k(t), ..., x_p(t)$ and, potentially, in their correlation w.r.t t. There can be scattering among functional outliers depending on how outlyingness is expressed. According to the functional-outlier taxonomy in [Hubert *et al.* 2015], there are two general classes: isolated and persistent outliers. An isolated outlier exhibits extreme behavior in a small part of the domain $\mathcal{T}$, resulting in a narrow peak in at least one of the variables. By contrast, a persistent outlier is defined as a sample in which outlyingness manifests itself in a large part of the domain. Among persistent outliers, different classes were distinguished by [Hubert *et al.* 2015] as follows, see their description and illustration in Section I.2.5.1.

The detection of shape outliers is quite recent and is attracting increasing attention in FDA [López-pintado *et al.* 2014, Arribas-Gil & Romo 2014, Kuhnt & Rehage 2016, Dai & Genton 2019]. Shape outliers are difficult to detect in a curve population because the shapes are often non-linearly discriminant (see Figure II.1(b) wherein the outlier curve is mixed with inliers but exhibits a different shape) and exhibit larger variability than isolated outliers. Considering curve discrimination in terms of shape, one can augment the curve variables by using differential analysis [Claeskens *et al.* 2014]. This refers to adding derivatives or integrals (computed with respect to t) for each initial variable. Hence, curve shape provides information regarding "hidden outlying features" of the curve variables and the outlying relationship between them. However, as mentioned previously, the joint analysis of the p variables becomes complex as p increases. In the present study, we address

this problem by using differential geometry. Specifically, we use aggregation functions (here termed as *mapping functions*) of the variables. Thereby, we implicitly consider the correlation of the variables through geometrical characterizations of curve shape. In contrast with current functional-outlier detection methods, which consider curve shape differently and only base the final detection on the resulting *depth values* (Section II.2), we use both functional curve-shape features and state-of-the-art outlier-detection algorithms *e.g.* Isolation Forest [Liu *et al.* 2008]. Thus, the originality of the proposed approach lies in the shape characterization of the initial curves through the proposed mapping functions whose output representation is input to state-of-the-art outlier-detection algorithms.

The mapping functions that we use refer to aggregation functions that enable capturing curve-shape features, such as curvature, length (*i.e.* perimeter of a shape), or tangential velocity, and consider all the variables, as a curve is viewed as a path. More precisely, a mapping function aggregates the variables through different interpretable combinations of the derivatives of the variables. Mapping functions have been used in shape analysis [Srivastava & Klassen 2016], that is, for curves lying in a two- or three-dimensional space (*e.g.* extracted from images), but not in the detection of multivariate functional outliers.

To capture the potential outlyingness of the curves through their shape, we propose mapping functions among those used in differential geometric-method in shape analysis [Srivastava & Klassen 2016]. These functions map multivariate to univariate curves; however, for accurate computation, they require the curves to be smooth. Although this is the case for multivariate functional data, raw data are often noisy when sampled, and we use the functional-data representation to recover the smooth version of the curves. Then, using the proposed mapping functions, we map the functional representation (in the form of a curve) so that some of its shape features capture curve outlyingness. Finally, based on this new representation, we use outlier-detection algorithms to assess the outlyingness of each sample and determine a threshold for flagging outliers.

Our contributions are summarized as follows:

(i) We propose an end-to-end method for detecting outliers through their curve shape, which is characterized by geometrical transformations. The method is based on the functional representation of the data.

(ii) We propose different mapping functions to capture different types of outlyingness based on curve shape.

(iii) We demonstrate that the proposed method is superior to previous outlier-detection algorithms and, in contrast to baseline methods, performs well regardless of the proportion of outliers.

This chapter is organized as follows. In Section II.2, we review related work on outlier detection in both univariate and multivariate functional data. In Section II.3, we discuss curve representations in the functional-data framework. In Section II.4, we present the mapping functions that can capture shape outlyingness from the obtained functional representation. The experimental results are presented and discussed in Section II.5. Finally, Section II.6 concludes this part.

II.2 Related work

II.2.1 Depth-based univariate functional outlier detection

The detection of outliers in functional data is a recent topic and has primarily been addressed by extending *statistical depth*[1] to *functional depth*. Statistical depth measures the centrality of a sample relative to a dataset by providing an outward-center ordering of the samples through a score lying in $[0, 1]$. A value close to zero implies that the sample is more likely to be an outlier [Tukey 1975]. Statistical depth has several theoretical properties (see [Zuo & Serfling 2000] for details):

(i) It attains its maximum value for the most centered (*i.e.* most representative) sample.

(ii) It decreases monotonically and vanishes as the sample moves away from the center (up to infinity).

(iii) It does not depend on the dataset scale.

Therefore, given an outlyingness threshold, samples with a depth value close to 0 can be flagged as outliers. This type of measure has been extended to functional data and used for classification [Cuevas & Febrero 2007], ranking [Fraiman & Muniz 2001, Cuevas *et al.* 2006], as well as outlier detection [Febrero *et al.* 2008].

However, most of the existing functional depths are applicable to univariate functional data only. For instance, given a functional sample, the integrated depth [Fraiman & Muniz 2001], modified band depth, and modified epigraph index [López-pintado & Romo 2009] evaluate depth pointwise, that is, at each observation point $t \in \mathcal{T}$. Then, these depth values are averaged by

[1] *statistical depth* was not specifically proposed for functional but for multivariate data [Tukey 1975, Fraiman & Muniz 2001]. However, we distinguish between *univariate functional depth* and *multivariate functional depth*, which were proposed specifically for functional data.

integration over $\mathcal{T}$ to provide a global outward-center score which can be seen as an outlyingness score. The integrated depth measures the proportion of a curve that is the closest to the median curve of the dataset, where the median curve is computed pointwise. The modified band depth measures the average proportion of the curve that takes values within the range of all pairwise sample combinations, where "proportion of a curve" refers to the size of the interval $\mathcal{T}$ where the curve outlies the dataset. The modified epigraph index has a similar principle: It measures the proportion of the curve that takes values smaller than the other values of the dataset. Thus, the functional depth intuitively measures the centrality of the curve, regarding its global shape with respect to the dataset, see [López-pintado & Romo 2009] for details. The bivariate random projection depth by Cuevas and Febrero in [Cuevas & Febrero 2007] considers specific shape information by projecting the curve and its first derivative onto random directions (*e.g.* directions generated according to a unit-variance Gaussian process), resulting in several bivariate vectors; a bivariate statistical depth function is then applied to these vectors and averaged over the random projections. Based on any of these functional depths, an outlyingness threshold is necessary for outlier detection. If the depth-value distribution is known, which is rare in practice, one can select the threshold as a small probability quantile (*e.g.* a sample with depth value lower than the 5%-quantile of this distribution is likely an outlier). [Febrero *et al.* 2008] proposed estimating this threshold as the first percentile of the empirical distribution of the depth values through a bootstrap procedure.

Unfortunately, apart from the statistical point of view, these approaches do not facilitate the understanding of the nature of outlyingness. These techniques have been developed for visually detecting univariate functional outliers. [Arribas-Gil & Romo 2014] defined the outliergram to represent each sample as a bivariate vector with the modified band-depth and epigraph values. They demonstrated that these depths are quadratically related. Hence, in a two-dimensional plot, inlier samples lie on a parabola, whereas outliers are likely to be far from it. Sun and Genton [Sun & Genton 2011] pro-

posed the functional boxplot to summarize the empirical distribution of the functional data as classical boxplots computed pointwise. It was designed to visualize a univariate functional dataset, in the same spirit as that of the classical boxplot. In their method, the central region of the pointwise boxplots is defined as the region in $\mathbb{R}$ where the 50% highest depth-score samples $\{x_i(t)\}_{i\leq n}$ (*i.e.* the most central) lie according to the band-depth ranks [López-pintado & Romo 2009]. The fences of the boxplots are defined by inflating 1.5 times the height of the central region. Thus, the continuum of the pointwise boxplots provides a functional boxplot. The outliers are then identified as samples falling outside the fences. In this functional boxplot, inliers and outliers rely heavily on curve magnitude. Thus, curve shape largely fails to be considered a potential outlyingness feature. In [Hyndman & Shang 2010], the authors applied robust principal component analysis by considering the samples to be high-dimensional vectors and represented each sample as a bivariate vector containing the first and second principal scores. Subsequently, outliers were identified as samples outside certain high-density regions that were determined using the empirical distribution of these bivariate vectors.

II.2.2 Depth-based multivariate functional outlier detection

Depth-based outlier detection methods for multivariate functional data are more recent. [Claeskens *et al.* 2014] generalized any given univariate functional depth to the case of multivariate functional data. This corresponds to a weighted sum of a given univariate functional depth applied to each variable $(\tilde{x}_1(t), ...\tilde{x}_k(t), ...\tilde{x}_p(t))$ pointwise and then integrated over $\mathcal{T}$. The authors also discussed the selection of the weight function. As a special case, in [Ieva & Paganoni 2013] proposed the multivariate band depth by using the modified band depth as the given univariate functional depth; the weights associated to the variables are constant with respect to t.

[Hubert *et al.* 2015] noted that the generalization proposed in

[Claeskens *et al.* 2014] does not allow to detect all shape outliers since low-depth samples stand near the boundary of the dataset but may not be outliers. Conversely, high-depth samples may present outlyingness in their curve shape because, pointwise, the curve does not exhibit any significant deviance in each variable, as this generalization is the sum of the individual univariate functional depths. To address this, the entire shape of the curve should be considered.

A few studies incorporate curve shape into a multivariate functional depth measure. Recently, [Kuhnt & Rehage 2016] proposed the functional tangential-angle ($FUNTA$) pseudo-depth, which considers curve shape based on the intersection angles of the centered variables (*i.e.* the variables are scaled so that their integral over $\mathcal{T}$ values is 0). More precisely, for each variable, $FUNTA$ computes the intersection angles of a given sample x_{ik} with all the other samples $x_{jk} \forall j \neq i$, and then averages these angles over the number of intersection angles of x_{ik} and over the variables $k = 1...p$. Thus, $FUNTA$ separately considers the shape for each variable with respect to t, but not the shape between the p variables.

More recently, [Dai & Genton 2019] proposed the directional outlyingness measure (*Dir.out*), which considers curve shape through the weighted pointwise direction in $\mathbb{R}^p$ of the vector $X(t)$ toward the median of the distribution of $X(t)$. The purpose of the weights is the up-weighting of the directions in which the outlyingness of $X(t)$ is likely to appear. In contrast with the aforementioned multivariate functional depths, which provide a score in $[0, 1]$, the *Dir.out* depth returns a vector in $\mathbb{R}^p \times \mathbb{R}^+$ corresponding to the concatenation of the mean directional outlyingness (in $\mathbb{R}^p$) and the total variance of the directional outlyingness (in $\mathbb{R}^+$). A final outlyingness score is computed as the robust Mahalanobis distance between this vector and a mean vector of the same type computed on a subset of independent samples. Then, the upper tail of this distance distribution is approximated by an F-distribution, and the outlyingness threshold is defined as a high-probability quantile of this F-distribution. Hence, unlike in other multivariate functional depths, the outlyingness threshold provided by the *Dir.out* approach is not data-driven,

as it is based on the (approximately) true distribution of the outlyingness scores. However, in this approach, the parameters should be tuned by simulation and are difficult to interpret beyond the statistical framework.

Multivariate functional depths are related to curve shape through the individual behavior of the curve variables. In our contribution, we adopt a different approach, as we view a curve as a path in $\mathbb{R}^p$ and process it as a geometrical shape with mapping functions. Indeed, the curve shape is not only reflected by the curve variables, individually along t, but also by the relationship between them. Our mapping functions consider the curve shape along t thus highlighting outlying correlation features between the variables.

As all the aforementioned multivariate functional depths yield an outlyingness score with unknown distribution (except for *Dir.out*), an outlyingness threshold can be computed from the resulting empirical distribution of the depth values through a bootstrap procedure as in the univariate case [Febrero *et al.* 2008]. It can also be computed from a training dataset based on the receiver operating characteristic (ROC) curve.

In the experimental study (Section II.5), we use the $FUNTA$ and *Dir.out* functional depths as baselines because they have been demonstrated to be promising for outlier detection in multivariate functional data by regarding outlyingness as a curve-shape feature.

II.2.3 Geometry-based functional outlier detection

Representing functional data in a geometric framework is a recent idea, and few studies have considered such representations for outlier detection. Recently, [Xie *et al.* 2017] proposed detecting outliers in univariate functional data by decomposing each univariate functional sample into three features: translation, phase, and amplitude. The authors defined the translation of a functional sample by its mean over the observation interval $\mathcal{T}$. Both the amplitude and phase components are functional data extracted from the original samples. The amplitude component reflects the vertical variability

of the functional data, whereas the phase component reflects the horizontal variability. Analogously to the functional boxplot of [Sun & Genton 2011], they computed on the original dataset (although the computational methods are quite different), the authors proposed a method for constructing a functional boxplot for each of the three components so that outlying features may be identified, and outliers may therefore be detected. [Xie *et al.* 2019] extended this method to multivariate functional data and added other components such as shape orientation (reflecting rotational variability). They additionally provided useful visualization techniques for identifying outlying features, in fact, they only focused on the bivariate, $p = 2$, and trivariate, $p = 3$, cases, which are shape data extracted from images. However, when the size of the dataset and the number of variables p increase, this method is computationally costly, as the shape-based component-extraction procedures include several continuous optimization problems. Moreover, in these studies, the outlier-detection methods are based purely on the empirical distribution (through the functional boxplot) of the proposed geometrical features, whereas in our work we geometrically aggregate the dimensions of the curves, resulting in univariate functional data and subsequently detect outliers from this new data representation. The latter can be seen as implicit non-parametric learning of the inlier distribution based on the functional data mapped to a geometric curve feature. Hence, we take advantage of both the geometrical mapping and the outlier-detection algorithm.

II.3 Background in functional data

This section is concerned with the handling of high-dimensional vectors of discrete noisy measurements that can be represented as smooth continuous functions; moreover, we discuss how such representations can be achieved. We use the notation introduced in Mathematical notations. The functional data representation is twofold:

(i) As the $\tilde{\mathbf{x}}_i$s are smooth functions, the reconstructed data are noiseless.

(ii) The reconstructed data are "aligned" in the sense that two reconstructed sample values $\tilde{\mathbf{x}}_1(t_j)$ and $\tilde{\mathbf{x}}_2(t_j)$ t_j at t_j are comparable, as they refer to the same evaluation point $t_j \in \tilde{\mathcal{T}}$.

This is not the case in raw data because one can have $t_{m_1} \neq t_{m_2}$ (the curves can be sampled on different grids).

II.3.1 Functional-data representation

The first step in FDA is to approximate an unknown smooth function $\mathbf{x}_i : t \rightarrow \mathbb{R}^p$, which underlies the sample i, by another smooth approximation function $\tilde{\mathbf{x}}_i(t)$, $\forall t \in \mathcal{T}$, through m_i discrete noisy measurements $\mathbf{x}_i(t_1), ..., \mathbf{x}_i(t_{m_i})$; this is referred to as the functional approximation step. Its purpose is to remove the noise, thus allowing accurate evaluations of some derived functions, such as combinations of derivatives and integral functions. This is necessary in our case, as the proposed mapping functions correspond to combinations of derivatives and integrals.

We should first select a functional representation as an approximation function. As a function is intrinsically infinite-dimensional, in FDA, it is commonly assumed that the underlying function can be approximated by a finite linear combination of non-linear basis functions. Such an approximation is called a basis expansion function [Ramsay & Silverman 2006]. We assume that x_{ik}, the k-th variable (hence a univariate function) of $\mathbf{x}_i$, is to be approximated. The intuition behind the basis expansion is to combine a small

number of "specific functions" (a set of given functions), each of which can capture some local features of the underlying function x_{ik}, so that x_{ik} could be recovered with a small approximation error. This approximation function can be formulated as

$$\forall t \in \mathcal{T}, \tilde{x}_{ik}(t) = \sum_{l=1}^{L_{ik}} \alpha_{ikl}\phi_l(t) = \boldsymbol{\alpha}_{ik}^{\top}\boldsymbol{\phi}(t) \tag{II.1}$$

where $\boldsymbol{\phi}(t) = \{\phi_l(t)\}_{1\leq l\leq L_{ik}}$ is a vector of orthonormal basis functions at t for some $L_{ik} \in \mathbb{N}^*$ (referred to as the basis size) with fewer basis functions than sampled observation points ($L_{ik} \ll m_i$), and $\boldsymbol{\alpha}_{ik} = \{\alpha_{ikl}\}_{1\leq l\leq L_{ik}}$ is the coefficient vector, the element α_{ikl} of which is the importance of the l-th basis function.

Another choice of functional representation in FDA is to use non-parametric smoothing [Ferraty & Vieu 2006], which achieves a similar approximation, but its form is less tractable than that of the basis expansion function, for instance, to compute derivatives.

According to Equation (II.1), one should select

(i) the basis $\{\phi_l\}_{1\leq l\leq L_{ik}}$ and

(ii) the basis size L_{ik}.

The choice of the basis is data-dependent. As suggested by Ramsay and Silverman [Ramsay & Silverman 2006], when the data are smooth and periodic, the Fourier basis should be selected; when the data are smooth, a B-spline basis is suitable. A B-spline is a piecewise-polynomial function of order at least three [De Boor 1978]. It is not exactly an orthonormal basis but since we only exploit the differentiability that B-splines induce, here we omit the non-orthonormality. If one requires orthonormality and smoothness, the functional principal components basis is a good solution [Ramsay & Silverman 2006].If the data have irregularities, a wavelet basis

should be preferred [Nason 2008]. See [Ramsay *et al.* 2009] for other examples and details on the choice of the basis according to the data. In turn, the choice of the basis size parameter L_{ik} depends on the selected basis. An inappropriate choice of the basis results in requiring a large L_{ik} because each basis function will focus on an irrelevant part of the data variability (low bias and high variance or, high bias and low variance); the worst case is to capture the noise, leading to over-fitting [Ramsay & Silverman 2006]. By contrast, an appropriate choice of the basis functions results in a small L_{ik}, that is, the basis is sufficiently rich to approximate an unknown function using few functions. Subsequently, once a suitable basis is selected, the bias–variance trade-off should be considered. This refers to the balance between the approximation error and a reasonable L_{ik} [Ramsay & Silverman 2006]. Such a balance is generally achieved by a grid search by cross-validation for each sample i and variable k. When $\boldsymbol{\phi}(t)$ and L_{ik} are specified, a computing method is required to estimate the coefficient vector $\boldsymbol{\alpha}_{ik}$, which is introduced in the next paragraph.

II.3.2 Functional-data fitting

The linearity of the basis expansion function with respect to the coefficient vector $\boldsymbol{\alpha}_{ik}^{\top}$ enables its efficient estimation (assuming the data were sampled with a Gaussian noise ϵ_{ij}, that is, $x_{ik}(t_{ij}) = \tilde{x}_{ik}(t_{ij}) + \epsilon_{ij}$, where ϵ_{ij} is independent of $\tilde{x}_{ik}(t_{ij})$) by minimizing the least-squares criteria:

$$\boldsymbol{J}(\boldsymbol{\alpha}_{ik}) = \sum_{j=1}^{m_i} (x_{ik}(t_{ij}) - \tilde{x}_{ik}(t_{ij}))^2 \tag{II.2}$$

or equivalently, with vector notation,

$$\boldsymbol{J}(\boldsymbol{\alpha}_{ik}) = \|x_{ik}(t_{i\bullet}) - \boldsymbol{\Phi}_{ik}\boldsymbol{\alpha}_{ik}\|_2^2 \tag{II.3}$$

where $t_{i\bullet}$ is, by abuse of notation, the vector containing all the samples $\{t_{ij}\}_{1\leq j\leq m_i}$ and $\boldsymbol{\Phi}_{ik} = (\phi_l(t_{ij}))_{1\leq j\leq m_i, 1\leq l\leq L_{ik}}$ is the $m_i \times L_{ik}$ matrix containing all the L_{ik} basis functions evaluated at the observation points. Thus, $\boldsymbol{\Phi}_{ik}$ is a discretization over $t_{i\bullet}$ of the vector of orthonormal basis functions $\boldsymbol{\phi}(t)$ in Equation (II.1). As $L_{ik} \ll m_i$ and $\boldsymbol{\Phi}_{ik}$ has all its columns linearly independent, by the orthonormality of the basis functions (and thus orthonormality of $\boldsymbol{\Phi}_{ik}$), $\boldsymbol{\Phi}_{ik}^\top \boldsymbol{\Phi}_{ik}$ is invertible. Hence, equating the gradient of $\boldsymbol{J}$ to $\mathbf{0}$ with respect to $\boldsymbol{\alpha}_{ik}$ leads to the following minimizer:

$$\boldsymbol{\alpha}_{ik}^* = \arg\min_{\boldsymbol{\alpha}_{ik}} \boldsymbol{J}(\boldsymbol{\alpha}_{ik}) = (\boldsymbol{\Phi}_{ik}^\top \boldsymbol{\Phi}_{ik})^{-1} \boldsymbol{\Phi}_{ik}^\top x_{ik}(t_{i\bullet}) \tag{II.4}$$

which is known as the classical least-squares solution [Hastie *et al.* 2009].

However, as the data are fitted according to the basis functions, the *smoothness* of $\tilde{x}_{ik}$ depends greatly on the noise influence on the basis functions. Consequently, $\tilde{x}_{ik}$ may lack smoothness and overfit the data. To analyze such a noise influence, one can compute the derivative of $\tilde{x}_{ik}$, which is "excessively" variable if a large amount of noise remains in the approximation function. To ensure smoothness, the least-squares criteria should be minimized by penalizing the derivative(s) of $\tilde{x}_{ik}$ with an amount $\lambda > 0$ as follows:

$$\boldsymbol{J}_\lambda(\boldsymbol{\alpha}_{ik}) = \sum_{j=1}^{m_i} (x_{ik}(t_{ij}) - \tilde{x}_{ik}(t_{ij}))^2 + \lambda \int_{\mathcal{T}} (D^q \tilde{x}_{ik}(t))^2 \mathrm{d}t \tag{II.5}$$

where $D^q = \frac{\mathrm{d}^q(\cdot)}{\mathrm{d}t^q}$ is the q-th derivative of $\tilde{x}_{ik}(t)$. More generally, D^q can be any linear combination of derivatives of x_{ik}, that is, a linear differential operator [Ramsay & Silverman 2006]. A penalization term including derivatives is also known as a *roughness penalty*. The parameter λ is arbitrary and can be computed by cross-validation. This is detailed in Section II.5.2. Equation (II.5) can be written using vector notation as follows:

$$\boldsymbol{J}_\lambda(\boldsymbol{\alpha}_{ik}) = \|x_{ik}(t_{i\bullet}) - \boldsymbol{\Phi}_{ik}\boldsymbol{\alpha}_{ik}\|^2 + \lambda\boldsymbol{\alpha}_{ik}^\top \boldsymbol{R}_{ik}\boldsymbol{\alpha}_{ik} \tag{II.6}$$

where $\boldsymbol{R}_{ik} = (\int_{\mathcal{T}} D^q\boldsymbol{\phi}_j(t)D^q\boldsymbol{\phi}_m(t)\mathrm{d}t)_{1\leq j\leq L_{ik},1\leq m\leq L_{ik}}$ is a $L_{ik}\times L_{ik}$ positive semi-definite matrix. The matrix $\boldsymbol{R}_{ik}$ contains the inner products of the q-th derivative of the L_{ik} basis functions. This matrix can be computed provided that the q-th derivative of the basis functions exists. In practice, it is common to choose $q = 1$ or $q = 2$ (*i.e.* to penalize the velocity or acceleration of $\tilde{x}_{ik}$, or a combination of both).

As $\boldsymbol{J}_\lambda$ remains quadratic with respect to $\boldsymbol{\alpha}_{ik}$, approximating $\tilde{x}_{ik}$ with a roughness penalty is equivalent to ridge regression [Hoerl & Kennard 1970, Hastie *et al.* 2009]. Thus, the penalty term allows $\tilde{x}_{ik}$ to

(i) be smooth, as defined by the operator D^q and,

(ii) avoid over-fitting by pushing the coefficient vector near $\mathbf{0}$.

Equating the gradient of $\boldsymbol{J}_\lambda$ to $\mathbf{0}$ with respect to $\boldsymbol{\alpha}_{ik}$ leads to the following minimizer [Hastie *et al.* 2009, Ramsay & Silverman 2006]:

$$\boldsymbol{\alpha}^*_{ik,\lambda} = \arg\min_{\boldsymbol{\alpha}_{ik,\lambda}} \boldsymbol{J}_\lambda(\boldsymbol{\alpha}_{ik,\lambda}) = (\boldsymbol{\Phi}_{ik}^\top\boldsymbol{\Phi}_{ik} + \lambda\boldsymbol{R}_{ik})^{-1}\boldsymbol{\Phi}_{ik}^\top x_{ik}(t_{i\bullet}) \tag{II.7}$$

II.3.3 Approximation functions as building blocks

Once the coefficient vectors have been estimated for the p variables of the n samples (with or without penalization), we can consider the approximations $\tilde{\mathbf{x}}_{ik}$ to be smooth multivariate functions that well recover the underlying functions. Although these functions can be theoretically evaluated at an infinite number of points in $\mathcal{T}$, in practice, there are two methods to handle the approximations computationally (*e.g.* to compute *derived functions* such as derivatives and integrals):

(i) The first method is to compute the derived functions based on the basis functions. As the basis functions are known analytically, their derived functions can also be obtained analytically. Thus, by the linearity of the basis expansion, one can easily obtain the derived functions of the approximation functions (the integral and derivative are linear operators). We illustrate this using the k-th derivative of the approximation function. We assume that an unknown function x is approximated by $\tilde{x}$ through a basis expansion with a basis size L (in Equation (II.1)), provided that the k-th derivative $\{D^k\phi_l(t)\}_{1\leq l\leq L}$ of the basis functions exists, and the coefficient vector $\{\alpha_l\}_{1\leq l\leq L}$ is available (or has been estimated as in Equation II.4). The k-th derivative of $\tilde{x}$ with respect to t is $D^k\tilde{x}$, where

$$\forall t \in \mathcal{T}, D^k\tilde{x}(t) = D^k\left(\sum_{l=1}^{L}\alpha_l\phi_l(t)\right) = \sum_{l=1}^{L}\alpha_l D^k\phi_l(t) \tag{II.8}$$

(ii) The second method is to estimate the underlying functions by evaluating all the approximation functions on the same grid $\tilde{\mathcal{T}}$. Thus, from these estimates, one can compute derived functions, such as integral or derivatives, using numerical methods, such as quadrature or finite difference schemes, respectively [Stoer & Bulirsch 2013]. These methods are easy to implement, but they do not consider the basis functions and require that the arbitrary grid be sufficiently fine (so that the approximation functions are evaluated at a large number of observation points).

Thus, if the derivatives of the basis functions are known analytically (as is the case for B-splines, Fourier basis functions, etc.), the derivatives of $\tilde{x}$ are also known and do not need to be estimated from the raw data or from the smooth reconstructions by a data-driven method such as finite differences. Equation II.8 demonstrates the flexibility of the linear basis expansion for computing derived functions in FDA. Then, a derived function, for instance $D^1\tilde{x}$, can be evaluated on an arbitrary grid. Such an approach is different

from estimating the derivatives from an evaluation of $\tilde{x}$ on the grid by using finite differences.

The first method is safer than the second because the analytic form of the basis functions is fully considered, and therefore the corresponding derived functions can be obtained accordingly. For instance, if the basis functions ϕ_l are B-splines (which are piecewise polynomial), we know the analytic form of $D^1\tilde{x}$, as $D^1\phi_l$ results in a piecewise polynomial as well. Thus, the evaluation of $D^1\tilde{x}$ by the first method provides more accurate estimates of D^1x (which is unknown) than numerical methods applied to $\tilde{x}$ evaluated on a fine grid of $\mathcal{T}$.

In the following section, we suggest some mapping functions for capturing functional outlyingness in the detection process. These mapping functions may have a complex analytical form because they involve several derivative (first and second order derivatives, as well as integral functions). Therefore, it is mandatory to have accurate evaluations of derivative functions, and we follow the first method in the computational experiments since we use B-splines and Fourier basis functions, whose derivatives are known.

II.4 Shape-based representation of curves

We regard a multivariate curve as a path lying in a p-dimensional space, $\mathbb{R}^p$, and derive mapping functions (aggregation functions over the variables), established in differential geometry, to capture shape features of the curves (*e.g.* length, velocity, or curvature) so that outlying features may be detected. These mapping functions have been used in shape analysis to extract features based on the edge (a bivariate curve) of an object in an image [Srivastava & Klassen 2016].

In this section, we investigate several mapping functions that enable the detection of multivariate functional outliers from the shape they exhibit in $\mathbb{R}^p$. Such mappings jointly consider the p variables, as they aggregate, in several ways, some derivatives (with respect to t) of the curve variables. Hence, the individual and collective variations of the variables are considered. These mapping functions take each data sample, represented by its smooth approximation function $\tilde{\mathbf{x}}_i$, as input and return a univariate curve (*i.e.* the resulting aggregation) reflecting certain shape features. Hence, they provide a means to "summarize" the shape of a multivariate curve, in the sense given by the mapping function, and reduce the number of functional variables to one. The univariate function returned by a mapping function is then fed into an outlier-detection algorithm; this is detailed in Section II.5. In the sequel, we simplify the notations by referring to a functional-data sample as an arbitrary curve $\mathbf{x} = \Big[x_1, \ldots, x_p\Big]$ instead of $\tilde{\mathbf{x}}_i = \Big[\tilde{x}_{i1}, \ldots, \tilde{x}_{ip}\Big]$.

II.4.1 Arc-length mapping

The arc-length mapping function enables analyzing the length of a curve between two points in $\mathcal{T}$ (see Figure II.2). Let $\mathbf{x}(t)$ be an arbitrary curve depending on a continuous variable $t \in \mathcal{T}$. For $t_0 \in \mathcal{T}$ and $t_0 < t$, the length $s(t)$ of the curve that $\mathbf{x}(\cdot)$ represents from t_0 to t is

$$s(t) = \int_{t_0}^{t} \|D^1\mathbf{x}(u)\|_2 \mathrm{d}u = \int_{t_0}^{t} \sqrt{\sum_{k=1}^{p} \frac{\mathrm{d}x_k(u)}{\mathrm{d}u}^2} \mathrm{d}u \tag{II.9}$$

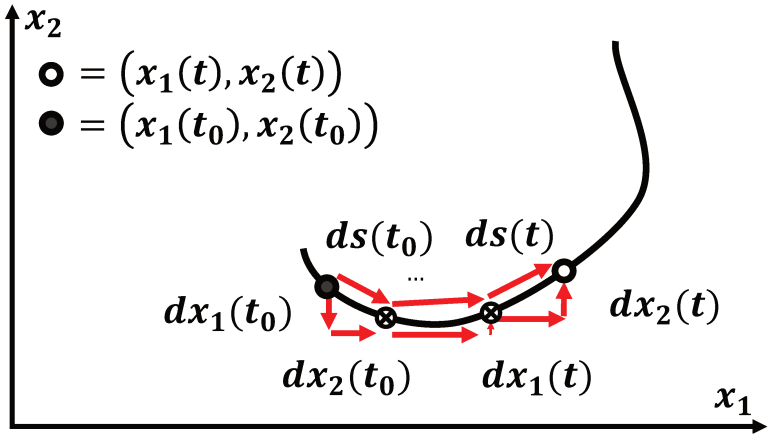

Figure II.2: The length of the curve between two observation points t_0 (dark-grey dot) and t (white dot) is defined as the sum of infinitesimal length elements $ds(t_0)...ds(t)$ along the curve (red diagonal arrows) for all t. The crossed-circle dots represent such points between t_0 and t.

Hence, the arc-length maps an original functional-data sample to univariate functional data that represent the increases in the cumulative length of the underlying curve from the starting-point $\mathbf{x}(t_0) = ((x_1(t_0), x_2(t_0))$ to an arbitrary point $\mathbf{x}(t) = ((x_1(t), x_2(t))$ for $t > t_0$. Figure II.2 shows that the length of a bivariate curve between $\mathbf{x}(t_0)$ and $\mathbf{x}(t)$ is the infinite sum from t_0 to t of infinitesimal length elements $ds(\cdot)$ (aka integral), corresponding to an infinitesimal length element in each direction (x_1 and x_2) in $\mathbb{R}^2$. This mapping always returns a positive increasing function, as it computes the cumulative length of the initial curve. Moreover, the arc-length mapping function is not influenced by a warping (*i.e.* a horizontal deformation) of the curve[2] which is usual in FDA to compare curves sampled on different grids [Srivastava & Klassen 2016]. This mapping function can discern functional samples with a shape of different size, which is a global shape feature.

[2]Let $\alpha(\cdot)$ be a differentiable warping function *i.e.* a monotone non-decreasing function defined in $\mathcal{T} \to \mathcal{T}$. The arc-length mapping function on a warped functional datum $\mathbf{x}$ is equal to the arc-length mapping function on the initial unwrapped functional datum: $s(\alpha(t)) = \int_{t_0}^{t} \|D^1\mathbf{x}(\alpha(u))\| \mathrm{d}u = \int_{t_0}^{t} \langle D^1\mathbf{x}(\alpha(u)), D^1\mathbf{x}(\alpha(u))\rangle^{1/2} \mathrm{d}u = \int_{t_0}^{t} D^1\alpha(u) \langle D^1\mathbf{x}(\alpha), D^1\mathbf{x}(\alpha)\rangle^{1/2} \mathrm{d}u$, and as $D^1\alpha(u) = \frac{\mathrm{d}\alpha}{\mathrm{d}u}$, we have $\int_{t_0}^{t} D^1\alpha(u) \langle D^1\mathbf{x}(\alpha), D^1\mathbf{x}(\alpha)\rangle^{1/2} \mathrm{d}u = \int_{t_0}^{t} \langle D^1\mathbf{x}(\alpha), D^1\mathbf{x}(\alpha)\rangle^{1/2} \mathrm{d}\alpha = \int_{t_0}^{t} \|D^1\mathbf{x}(\alpha)\|_2 \mathrm{d}\alpha$, which implies that $s(\alpha(t)) = s(t)$.

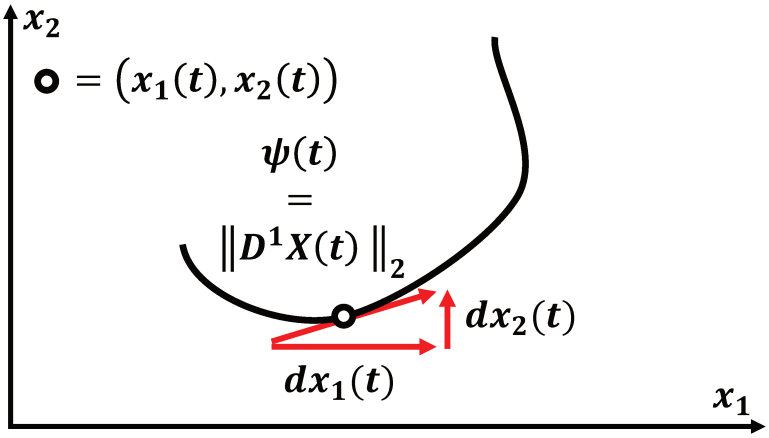

Figure II.3: The norm of the tangent vector $D^1\mathbf{x}(t)$ (red diagonal arrow), the components of which are infinitesimal variations $(dx_1(t), dx_2(t))$, shown by the horizontal and vertical red arrows of the variables of the curve allows the computation of the speed at which the curve "progresses."

Thereby, the detection of functional outliers can be improved when their underlying curve is longer or shorter than those of the other samples. For instance, an isolated outlier, which exhibits a peak for a small part of $\mathcal{T}$, induces a sharp increase in its curve length, whereas the length of other curves increases more slowly.

II.4.2 Velocity mapping

The velocity mapping function enables analyzing the instantaneous variations of the curve with respect to t. It has a simple interpretation when t corresponds to a time instant. In this case, velocity measures how fast a point moves on the curve. More generally, it can be interpreted as the norm of the projection of the curve onto $D^1Y(t)$, the tangent vector to the curve at t. In Figure II.3, the velocity mapping at t of a bivariate curve is shown as the ℓ_2-norm of the tangent vector $D^1\mathbf{x}(t)$ (vector of the first-order derivatives of the curve variables x_1 and x_2). It is defined as

$$\psi(t) = \|D^1\mathbf{x}(t)\|_2 \tag{II.10}$$

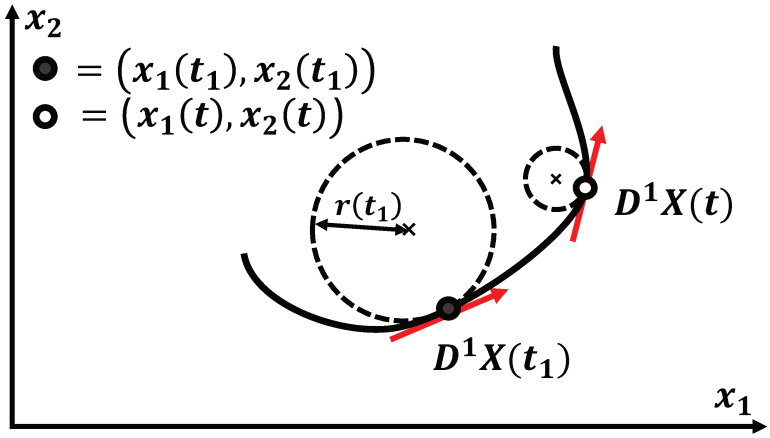

Figure II.4: Curvature is defined to be the inverse of the radius of the osculating circle. In this example, in a neighborhood of the curve at t_1 (dark-grey dot), the *tangent vector* $D^1\mathbf{x}(t_1)$ has almost the same direction; hence, the osculating circle has a large radius ($r(t_1) = \frac{1}{\kappa(t_1)}$), resulting in a small curvature. In a neighborhood of the curve at t (white dot), the tangent vector $D^1\mathbf{x}(t)$ quickly changes direction; hence, the osculating circle has a lower radius, that is, a higher curvature than at t_1.

and is related to the arc-length mapping by $\psi(t) = \frac{\mathrm{d}s}{\mathrm{d}t}$, or conversely, by $s(t) = \int_{t_0}^{t} \psi(t)\mathrm{d}t$; however, these mappings capture different features. Indeed, the arc-length mapping outputs an increasing function and thus "memorizes" the local variations of the curve as t increases, whereas the velocity mapping characterizes the local variations (*i.e.* pointwise) with respect to t. The function returned by the velocity mapping may be regarded as a measure of the variation of the arc-length mapping. Thus, the velocity mapping can be used to identify the local outlyingness of a sample that is to say an isolated outlier.

II.4.3 Curvature mapping

Curvature is a notion that relates to how "bended" a curve is, or geometrically, the degree to which a curve deviates from the tangent line at a given point. An alternative interpretation concerns the radius of the osculating circles. At a given point t, a smaller radius of the osculating circle implies larger curvature. In fact, the radius of the osculating circle is equal to the inverse of

the curvature at this point. The bivariate curve in Figure II.4 shows that at a neighborhood of t_1, where the tangent vector $D^1\mathbf{x}(t_1)$ has almost constant direction, the osculating circle has a larger radius $r(t_1)$ than the radius of the osculating circle at a neighborhood of t where the direction of the tangent vector $D^1\mathbf{x}(t)$ changes quickly. Thus, the curvature mapping function allows analyzing the change of direction of the curve with respect to t. Indeed, if the curve is a line, curvature is constant, and the curve directions remain constant as well. Curvature is defined as [Srivastava & Klassen 2016]

$$\kappa(t) = \frac{\|D^1(\frac{D^1\mathbf{x}(t)}{\|D^1\mathbf{x}(t)\|_2})\|_2}{\|D^1\mathbf{x}(t)\|_2} \tag{II.11}$$

or equivalently,

$$\kappa(t) = \frac{\sqrt{\|D^1\mathbf{x}(t)\|_2^2\|D^2\mathbf{x}(t)\|_2^2 - \langle D^1\mathbf{x}(t), D^2\mathbf{x}(t)\rangle^2}}{\|D^1\mathbf{x}(t)\|_2^3} \tag{II.12}$$

We now provide insight into the definition of κ in Equation (II.11). $\frac{D^1\mathbf{x}(t)}{\|D^1\mathbf{x}(t)\|_2}$ is the direction vector (*i.e.* the normalized tangent vector); therefore, $D^1\frac{D^1\mathbf{x}(t)}{\|D^1\mathbf{x}(t)\|_2}$ is the rate of change of the direction vector, and the normalization $\|D^1\mathbf{x}(t)\|_2$ relates to the rate of change of the direction with respect to the tangent vector. Consequently, the curvature mapping can detect functional outliers with a curve that exhibits a differently bended shape than those of the other samples.

II.5 Experimental study

We conducted an experimental study on real and synthetic datasets to demonstrate the effectiveness of the proposed mapping functions in improving outlier detection in multivariate functional data. The detection performance was evaluated in terms of the true detection rate (*i.e.* the proportion of outliers correctly detected), false detection rate (*i.e.* the proportion of outliers falsely detected), and area under the ROC curve (AUC).

II.5.1 Real data

II.5.1.1 ECG data

We tested the proposed approach on the real dataset used in [Dai & Genton 2019]. The dataset consists of electrocardiogram (ECG) time series of the electrical activity (voltage) of heart changes [Goldberger *et al.* 2000]. Such data can reveal abnormalities in heart activity. The time series are univariate and were labeled by cardiologists as *abnormal* or *normal*. This dataset has been used for time-series classification [Wei & Keogh 2006].

The ECG dataset consists in $n = 810$ time series including 208 abnormal and 602 normal cases. All the time series have an equal size of $m_i = 86$. In contrast with [Dai & Genton 2019], who only considered the time series between the time stamps $t = 6$ and $t = 80$ to avoid boundary effects, we considered the entire time series to demonstrate the robustness and applicability of the proposed approach. Dai and Genton also augmented the univariate time series to multivariate by adding the first and the second derivatives. We did not follow this, as in our proposal, these aspects are considered (*e.g.* velocity mapping in Equation (II.10) or curvature mapping in Equation (II.11)); rather, we added the squared time series. Indeed, power is proportional to the square of voltage. Thus, in terms of interpretability, this data augmentation appears to be more relevant than the second derivative of voltage. We applied the same multivariate functional data augmentation to all ECG data

experiments and for all methods. We did not apply the derivative augmentation, as this would bias the interpretation of the results, that is, it would not be possible to discern whether the results were due to the specific augmentation or to the method. This would be of interest if the focus was specifically on the ECG data, but here, we use it as a real dataset example.

As in [Dai & Genton 2019], to obtain a rare class of samples representing outliers, we randomly created a partition of 400 samples (*i.e.* the training set) out of the 810 samples by parameterizing the contamination level (*i.e.* the rate of *abnormal* samples) in this partition to 5%, 10%, 15%, 20%, and 25%. Then, for each contamination level, we evaluated the proposed method on the 410 remaining samples (*i.e.* the test set).

II.5.1.2 Pen-digits data

We tested the proposed method on the real dataset consisting of $n = 10,992$ bivariate time series representing pen digits (PenDig) [Dua & Graff 2017]. The digits are labeled according to their class, from '0' to '9'. Each digit has $m_i = 8$ observation points regularly sampled on both the horizontal and vertical coordinates. As this initial dataset cannot be considered high-dimensional, we upsampled it by linear interpolation to $m' = 200$ on the two coordinates before fitting the approximation functions.

To simulate the outlier classes, we considered a single digit to be the outlier class, and the nine other classes to be the inlier class, as in [Ruff *et al.* 2018]. The training set was generated using 75% of the entire dataset with a contamination level equal to $c = 5\%$ (*i.e.* 5% of the training set are outliers). Each digit was separately considered the outlier class, and thus the experiment was conducted in 10 independent ways. Then, for each case of outlier class, we assessed the proposed method on the test set.

II.5.1.3 Airbus flight test data

We also tested our method on a real dataset provided by an Airbus flight test expert, of $n = 7,118$ four dimensional multivariate time series wherein

each variable refers to a flight parameter. This dataset contains 85 outliers annotated by the expert. For data privacy, we cannot share the Airbus flight test data (AFT) neither give technical details on its physical meaning. We show a plot of the scaled time series in Figure I.2 where red curves corresponds to outliers.

The ECG and PenDig datasets were not used to assess the same properties of the proposed method. The ECG data were used to demonstrate the robustness of the proposed method with respect to different contamination levels for some given outliers, whereas the PenDig data were used to assess the detection performance for different outliers and a given contamination level. Thus, we only compared these two in terms of performance, in the comparison of the various methods in Section II.5.8. The AFT dataset served as an application example of outlier detection in flight test data.

II.5.1.4 Synthetic data

We simulated five multivariate functional datasets according to different models proposed in [Dai & Genton 2019]. To the best of our knowledge, this is the most recent study concerned with outlier detection in multivariate functional data providing detection rates. For each synthetic dataset, $n = 150$ curves are generated by a common simulation model with a continuous variable discretized on a regular grid of size $m = 200$ in the interval $[0, 1]$. Among the n curves, $c = 10\%$, referred to as the contamination level, were outliers generated by specific contamination models $\mathbf{x}_{c1}(t) \dots \mathbf{x}_{c5}(t)$. Each contamination model generates one type of outlier. Testing the proposed approach and the baselines on with different contamination models enables assessing the efficiency of each mapping function for a given type of outlier.

The uncontaminated model is a bivariate Gaussian process $\mathcal{GP}(\boldsymbol{\mu}(t), \boldsymbol{\Sigma}(s,t))$ [Rasmussen 2003], with a constant mean function $\boldsymbol{\mu}(t) = \mathbf{0}$, and a cross-covariance function C_{kr} between the two variables indexed by k and r, as

follows:

$$C_{kr}(s,t) = \rho_{kr}\sigma_k\sigma_r\mathcal{M}(|s-t|;\nu_{kr},\beta_{kr}) \quad k,r = 1,2 \text{ and } s,t \in [0,1] \quad \text{(II.13)}$$

where ρ_{12} is the correlation between the variables x_1 and x_2, $\rho_{11} = \rho_{22}$ is the variance of each variable, σ_1 and σ_2 are the marginal variances, $\mathcal{M}(h;\nu_{kr},\beta_{kr}) = 2^{1-\nu}\Gamma(\nu)^{-1}(\beta|h|)^{\nu}\mathcal{K}_{\nu}(\beta|h|)$ is the Matérn class function [Matérn 2013] ($\mathcal{K}_{\nu}$ is a modified Bessel function [Bowman 2012]), $\nu_{kr} > 0$ is a smoothness parameter, and $\beta_{kr} > 0$ is a range parameter. For this simulation, we used the same parameter setting as in [Dai & Genton 2019]: $\rho_{12} = 0.6$, $\rho_{11} = \rho_{22} = 1$, $\sigma_1 = \sigma_2 = 1$, $\nu_{11} = 1.2$, $\nu_{22} = 0,6$, $\nu_{12} = \nu_{21} = 1$, $\beta_{11} = 0.02$, $\beta_{22} = 0.01$, and $\beta_{12} = \beta_{21} = 0.016$. This covariance function is implemented in the R package [Schlather *et al.* 2015]. We summarize the uncontaminated model $\mathbf{u}(t) = (u_1(t), u_2(t))^{\top}$ as follows:

$$\mathbf{u}(t) \sim \mathcal{GP} = \left(\boldsymbol{\mu}(t) = \mathbf{0}; \boldsymbol{\Sigma}(s,t) = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}\right) \quad \text{(II.14)}$$

The five contamination models are (we annotate the variables with an index c referring to "contamination"):

1. Model 1 persistent magnitude outlier: $\mathbf{x}_{c1}(t) = 4\mathbf{u}(t)$.
2. Model 2 isolated outlier: $\mathbf{x}_{c2}(t) = \mathbf{u}(t)(1 + 11\mathbf{I}_{z<t<z+0.1})$, where $\mathbf{I}$ is the indicator returning 1 if the indexed condition is true, and 0 otherwise, and z is a uniform random variable in $[0, 0.9]$.
3. Model 3 persistent magnitude outlier, the contamination model is different for the two variables: $\mathbf{x}_{c3}(t) = (x_{1,c3}(t), x_{2,c3}(t))^{\top}$, with $x_{1,c3}(t) = 1.7u_1(t)$ and $x_{2,c3}(t) = 1.5u_2(t)$.
4. Model 4 isolated outlier: $\mathbf{x}_{c4}(t) = \mathbf{u}(t)(1 + 4\mathbf{I}_{z<t<z+0.1})$, with z as in model 2.

5. Model 5 persistent shape outlier, the new uncontaminated model is referred to as Y, and the contamination model as $\mathbf{x}(t)_{c5}$: $\mathbf{y}(t) = (y_1(t), y_2(t))^\top$ with $y_1(t) = u_1(t) + z_{11}\cos(4\pi t)$ and $y_2(t) = u_2(t) + z_{12}\sin(4\pi t)$, where z_{11} and z_{12} are independent uniform random variables in $[2, 3]$. The contamination model $\mathbf{x}(t)_{c5}$ is $x_{1,c5}(t) = u_1(t) + z_{21}\cos(4\pi t)$ and $x_{2,c5}(t) = u_2(t) + z_{22}\sin(4\pi t)$, where z_{21}, z_{22} are uniform random variables on $[4, 5]$.

II.5.2 Experimental protocol

II.5.2.1 Functional-data fitting

Without loss of generality, we selected $\mathcal{T} = [0, 1]$ as the domain (closed interval) of t for all the data sets. We recall that we represent all the curves in the common interval $\mathcal{T}$ because we assume that the functional samples were generated by a random function depending on t relating to the same event in $\mathbb{R}^p$. For instance, when the samples are measurements of a given process depending on t, which represents time, $\mathcal{T}$ can be viewed as the relative temporal range of the process (*i.e.* from the beginning at $t = 0$ to the end at $t = 1$), and $t \in \mathcal{T} = [0, 1]$ can be interpreted as the progress rate of the process.

Choice of the basis of functions. For the ECG, PenDig and the Airbus flight test datasets, we approximated each variable of the bivariate time series by a basis consisting of B-splines of order eight (B-splines are piecewise-polynomial functions of order at least three, and are located at a given observation point $t \in \mathcal{T}$). Indeed, we noticed that in this dataset, the curves exhibit a smooth pattern without periodicity; hence, the B-spline basis is a suitable choice, as recommended in [Ramsay & Silverman 2006].

For the synthetic dataset, we approximated each variable of the bivariate time series by a Fourier (sine and cosine functions) basis with a fundamental period of $T = \frac{1}{F} = 1$ (*i.e.* the length of $\mathcal{T}$). The Fourier basis was suitable because we noticed low-frequency periodicity (induced by the covariance function

$C_{kr}(s,t)$) over $\mathcal{T}$.

Application of the functional-data fitting procedure. We now provide the computational details of the functional-data fitting. Following the recommendations in [Ramsay *et al.* 2009, Febrero-bande & Oviedo de la Fuente 2012], for all datasets, we selected both the penalization λ_k and the basis size L_{ik} for the variable k of sample i through a leave-one-out cross-validation procedure over a given grid search for λ_k and L_{ik}. We penalized both the first- and second-order derivatives of $\tilde{x}_{ik}$ to gain smoothness in the mapping-function output. We note that for all the samples of a given variable k, we equally penalized the approximations $\tilde{x}_{ik}$ by the same λ_k to compute the coefficient vector $\boldsymbol{\alpha}^*_{ik}$. Then, by computing $\boldsymbol{\alpha}^*_{ik,\lambda_k}$ according to Equation (II.7), we selected the value of λ_k and $L_{ik} < m_i$ that minimize the leave-one-out cross-validation score $CV_{\lambda_k}(L_{ik})$,

$$CV_{\lambda_k}(L_{ik}) = \sum_{j=1}^{m_i} \Big(x_{ik}(t_j) - \tilde{x}_{ik}^{-j}(t_j)\Big)^2 \tag{II.15}$$

where $\tilde{x}_{ik}^{-j}$ corresponds to the approximation of x_{ik} by L_{ik} basis functions by omitting the pair $(t_j,\ x_{ik}(t_j))$ in the functional-fitting step, as in Equation (II.5), where the penalization is λ_k.

For the ECG, PenDig and AFT datasets, the grid search of λ_1 and λ_2 was fixed on logarithmic scale in $[-9,-1]$, with a thickness of 0.1. The grid search of L_{ik} was fixed as $\{35,\ldots,60\}$, that is, for a given L_{ik}, the L_{ik} B-spline functions are regularly located in $\mathcal{T}$.

For the synthetic datasets, the grid search of λ_1 and λ_2 was fixed on logarithmic scale in $[-9,-4]$, with a thickness of 0.1. The grid search of L_{ik} was fixed in $\{20,\ldots,25\}$, that is, for a given L_{ik}, the synthetic data were approximated by the first L_{ik} frequencies $2\pi F \times \ldots 2\pi F L_{ik}$. Then, for each variable, we retained the coefficient vector associated with both the optimal regularization and basis-size parameters to recover the smooth approximation function

$\tilde{\mathbf{x}}_i(t) = (\tilde{x}_{i1}, \tilde{x}_{i2})$.

Finally, we used the coefficient vector associated with both the optimal regularization and basis-size parameters to recover the smooth approximation functions $\tilde{\mathbf{x}}_i(t)$ on a given grid and applied a mapping function to them.

II.5.2.2 Applying the mapping functions

We now explain the computational application of the mapping functions and then how their output was fed to an outlier-detection algorithm.

After computing the approximation functions $\tilde{\mathbf{x}}_i(t)$, we centered and scaled each variable x_{ik} with the empirical mean and standard deviation functions computed from the training set (see [Ramsay & Silverman 2006] for details on the computation of mean standard deviation functions). This scaling prevents the mapping functions from overweighting some variables with a wider range than others. Indeed,

(i) The variables require to be scaled since the unit of the output value of the arc-length mapping function (Len_{map} in Equation (II.9)) is intrinsically a length. Then, we applied the three mapping functions introduced in Section II.4. As the arc-length mapping is the integral function of the velocity mapping, the arc-length mapping in Eq (II.9) was computed from the minimum of $\mathcal{T}$ (*i.e.* $t = 0$) and was then integrated up to t for all $t \in \mathcal{T}$. In these experiments, the integral was efficiently estimated by a Riemann sum, as in this study, all the observation points were regularly sampled in $\mathcal{T}$, and therefore the sum converges to the integral. We note that if the observation points had been irregularly sampled, the integral could have not been approximated by a Riemannian sum, and numerical techniques, such as Simpson's or the trapezoidal rule, should have been used instead [Ramsay & Silverman 2006].

(ii) Regarding the velocity mapping V_{map} in Equation (II.10), the first-order derivative of each variable of $\tilde{\mathbf{x}}_i(t)$ was computed according to Equation (II.8).

(iii) The curvature mapping ($Curv_{map}$) requires the computation of both first- and second-order derivatives. Thus, we computed them as in Equation (II.8) and combined them as in Equation (II.11).

The approximation functions recover the functional data on the entire domain $\mathcal{T}$. Thus, the approximation functions can be computed on an irregular grid, and therefore the computation of the mapping functions should be carefully performed (*e.g.* (i) in the computation of an integral function). For both V_{map} and $Curv_{map}$, which are based on derivative functions only, simple and efficient derivative estimation methods can be used, as mentioned in Section II.3.3.

Each mapping function returns a univariate function. Thus, applying a mapping function to all n approximation functions $\tilde{\mathbf{x}}_i(t)$ results in n univariate functional-data samples. We used the resulting univariate functional data in several outlier-detection algorithms. In practice, the functions returned by a mapping function should be evaluated over a grid of observation points in $\mathcal{T}$ to obtain the output samples in vector form. As we selected $\mathcal{T} = [0, 1]$ for all datasets and the observation points are regular, the grid is a regular discretization $\{t_1 ... t_j ... t_J\}$ of $\mathcal{T}$ with a thickness of $\frac{1}{J}$ ($t_1 = 0$ and for $j > 1, t_j = \frac{j}{J}$). Hence, for the outlier-detection algorithms, the data correspond to J-dimensional numerical vectors that, in turn, correspond to univariate functional data output by a mapping function. We selected the thickness of the grid as the original size of the time series for both the synthetic and ECG datasets (ECG dataset: $J = m_i = m = 86$, PenDig dataset: $J = m' = 200$, AFT dataset: $J = 1000$, synthetic data sets: $J = m = 200$). An irregular grid can also be used to evaluate the approximated functions, but the computation of the mappings should be performed cautiously, as mentioned in (i) for Len_{map}.

II.5.3 Outlier detection from the functional data output by a mapping function

We detect outliers in the functional data returned by a mapping function using a state-of-the-art outlier-detection algorithm. To this end, we selected isolation forest (iFor) [Liu *et al.* 2008] and a one-class support vector machine (OCSVM) [Schölkopf *et al.* 2001]. iFor is a bagging model that generates a large number of decision trees grown on random subspaces. A subspace corresponds to a subsample of features randomly selected from the full feature space (here, $\{1 \dots j \dots J\}$). Each tree isolates the data samples based on a random split value of a randomly selected feature from the subspace until all the data samples have been isolated, or all the features of the subspace have been selected. The sample outlyingness score returned by a tree is based on the path length between the root node and the terminal node of a tree. Outliers are samples that are easy to isolate and thus have short path length in the trees. The path length is normalized in $[0, 1]$ so that if the score is close to 1, then the sample is likely an outlier. OCSVM is a distance-based model formulated as a constrained quadratic minimization problem, the variables of which correspond to the radius and the center of the smallest hypersphere containing the data. To allow flexibility on the hypersphere boundary owing to the presence of outliers in the training data, slack variables are introduced in the objective function in addition to the two other variables. The hyperparameter ν corresponds to an upper bound on the *a priori* proportion of outliers in the training set. A sample is declared as an outlier if it lies outside the fitted hypersphere. We used the radial-basis-kernel version of OCSVM with ν equal to the exact proportion of outliers in the training set. The bandwidth hyperparameter of the radial basis kernel was optimized by a 20-fold cross-validation procedure.

For the ECG, PenDig, AFT and synthetic datasets, we set the number of trees to 1000, and the subsampling size to 32 which are recommended values in [Liu *et al.* 2008]. We randomly split each dataset into a training set and a test set. As in [Dai & Genton 2019], the training set represents 50% of

the data for the ECG dataset. The training set for the PenDig and AFT datasets consists in 75% of the entire dataset. The training set contains 60% of the data for the synthetic data. The training set was used to both fit the model (iFor and OCSVM) and select an outlyingness threshold from the Receive Operating Curve (ROC) that discriminates inliers from outliers. We then computed the outlyingness score of the test samples and achieved detection using the previously computed outlyingness threshold. Regarding OCSVM, we finetuned the bandwidth hyperparameter of the radial basis kernel on the training set through a 20-fold cross-validation procedure on the grid $\{2^{-25}...2^{-5}\}$ for the real datasets as well as the synthetic ones.

II.5.4 Result assessment

We assessed the results with the correct detection rate ρ_c (*i.e.* number of correctly detected outlier divided by the total number of outlying curves) and the false detection rate ρ_f (*i.e.* number of falsely detected outliers divided by the total number of inliers). In addition, as a measure of discrimination between outliers and inliers by the proposed approach, we also computed Area Under the ROC (AUC) from the labels of the test set. It is a standard performance measure in outlier detection [Erfani *et al.* 2016, Liu *et al.* 2008] and demonstrates that the proposed method can outperform the baselines regardless of the computed outlyingness threshold.

The threshold-selection step is simple and is not part of iFor [Liu *et al.* 2008] or OCSVM [Schölkopf *et al.* 2001], which are both unsupervised. We assume that the training data is labeled even if there are few outlier samples. In real-world applications, the user has some knowledge about the training data and can thus label inliers and some outliers.

If the training set surely has no outlier, the proposed method only requires the modification of the threshold selection rule. This modification is easy because both iFor and OCSVM are unsupervised methods and output a normalized score. Using the threshold that achieve the highest AUC, we computed the correct detection rate (*i.e.* number of correctly detected outlier divided by

the total number of outlying curves) ρ_c and ρ_f (*i.e.* number of falsely detected outliers divided by the total number of inliers) to demonstrate the complete application of the proposed method and compare it with the baselines. There are other methods for learning an outlyingness threshold, such as using a specific decision rule involving, *e.g.* an empirical quantile associated with a reference distribution of the outlyingness scores [Dai & Genton 2019], or threshold selection from the mass-volume curve [Clémençon & Thomas 2017] when no outlier label is available, but this is beyond the scope of the present study, as we assume that the training set has low non-zero contamination level.

II.5.5 Baseline comparisons

We compared the proposed method with two recent outlier-detection ones based on multivariate functional depth.

The first baseline method is FUNTA, proposed by [Kuhnt & Rehage 2016] (see Section II.2). It only requires centering each variable x_{ik} of each sample to a zero mean. As FUNTA has been demonstrated to be robust to noise and can handle curves of different size, we used it on the raw data without any functional data approximation. For the computation of the outlyingness threshold, we applied the same procedure as in the proposed method, that is, we selected the best outlyingness threshold for the training set using ROC and applied it to the test set. We used the R implementation proposed in [Rehage 2016].

The second baseline method is *Dir.out* proposed in [Dai & Genton 2019] (see Section II.2). We used the same parameter setting as in [Dai & Genton 2019] and did not perform any functional-data approximation. In this method, the outlyingness score is based on the robust Mahalanobis distance of the directional outlyingness vector computed on a subset of the data; in the present case, we computed it using the training data to obtain comparable results and to assess the performance measures on the test set. The tail of the distribution of the distances is approximated by an F-distribution with

degrees of freedom $(p+1, m-p)$, where p is the number of curve variables, and m is calculated through a simulation procedure (see [Dai & Genton 2019], p. 7 for details). Consequently, the outlyingness threshold is not data-driven and is computed as a quantile of probability $99,3\%$ of an F-distribution. Then, we used the outlyingness threshold on the test set to asses performance. We used the R implementation provided by the authors.

II.5.6 Experimental protocol application

The performance of the proposed approach was evaluated by simulation for both the real and the synthetic data. The simulation settings for the ECG and synthetic data were as in [Dai & Genton 2019]. We proceeded as follows:

(i) We randomly generated a train/test split.

(ii) We then applied the proposed and the baseline methods. Except for *Dir.out* (baseline), which does not require outlyingness-threshold learning because the outlyingness score follows a known distribution (see Section II.5.5), the outlyingness threshold was learnt on the training set based on the ROC curve.

(iii) We evaluated the performance in terms of the true detection rate (ρ_c), false detection rate (ρ_f), and AUC on the test set.

For the ECG dataset (resp., PenDig datasets), steps (i) to (iii) were repeated 50 times for each case of the five contamination levels (resp., for the 10 outlier classes) (see end of Section II.5.1), and 500 times for the synthetic data for each of the five models (Section II.5.1.4). For the AFT dataset, steps (i) to (iii) were repeated 50 times.

II.5.7 Results and discussion

We report the results for the ECG dataset in Table II.1, where for each contamination level c (columns) and for each method (rows), we provide ρ_c, ρ_f, and AUC (sub-columns). The results for the PenDig dataset are shown

in Table II.2. The results for the synthetic data are reported in Table II.4. In these tables, the value in a cell is the average of a performance measure over the number of simulations. We conducted a hypothesis testing procedure in Section II.5.8 to validate the relevance of these averages.

II.5.7.1 ECG data

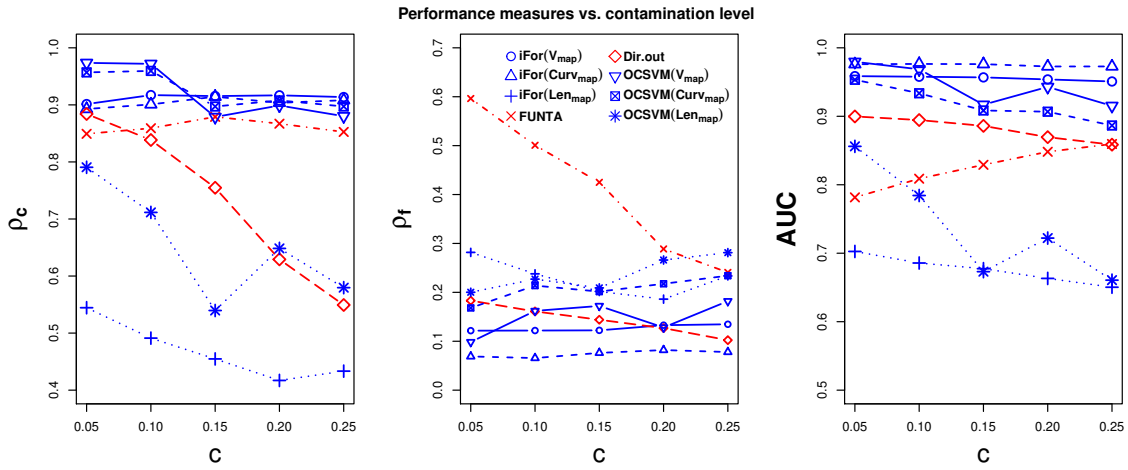

Figure II.5: The three performance measures ρ_c, ρ_f, and AUC, averaged over the simulations versus the contamination level ($c = 5\%, 10\%, 15\%, 20\%, 25\%$) for each method (proposed in *blue*, and baselines in *red*). We notice that when the contamination level c increases, the proposed method (except for iFor(Len_{map}) and OCSVM(Len_{map})) outperforms the baselines in terms of ρ_c, ρ_f and AUC.

The results for the ECG data set (Table II.1) show that the proposed method outperforms the baselines with V_{map} and $Curv_{map}$ with iFor and OCSVM. We notice that when the contamination level c increases, the proposed method (except for iFor(Len_{map}) and OCSVM(Len_{map})) outperforms the baselines in terms of ρ_c, ρ_f and AUC. Moreover, performance does not degrade as the contamination level increases, in contrast with that of the baselines. In terms of ρ_c, $FUNTA$ performs as well as V_{map} and $Curv_{map}$ when used with both iFor and OCSVM but significantly degrades in terms of ρ_f (*i.e.* it falsely detects outliers) for low contamination levels. *Dir.out* performs as well as the proposed method in terms of ρ_f but degrades in terms of ρ_c for high contamination levels. Hence, $FUNTA$ performs well when the contamination level

Table II.1: Results on the ECG dataset.

	$c = 5\%$			$c = 10\%$			$c = 15\%$			$c = 20\%$			$c = 25\%$		
Methods	ρ_c	ρ_f	AUC	ρ_c	ρ_f	AUC	ρ_c	ρ_f	AUC	ρ_c	ρ_f	AUC	ρ_c	ρ_f	AUC
$FUNTA$ (baseline)	0.85	0.60	0.78	0.86	0.50	0.81	0.88	0.42	0.83	0.87	0.29	0.85	0.85	0.24	0.86
Dir.out (baseline)	0.88	0.18	0.90	0.84	0.16	0.89	0.75	0.14	0.89	0.63	0.13	0.87	0.55	0.10	0.86
iFor(V_{map})	0.90	0.12	0.96	0.92	0.12	0.96	**0.92**	0.12	0.96	**0.92**	0.13	0.95	**0.91**	0.13	0.95
iFor($Curv_{map}$)	0.89	**0.07**	**0.98**	0.90	**0.07**	**0.98**	0.91	**0.08**	**0.98**	0.90	**0.08**	**0.97**	**0.91**	**0.08**	**0.97**
iFor(Len_{map})	0.54	0.28	0.70	0.49	0.24	0.69	0.45	0.20	0.68	0.42	0.19	0.66	0.43	0.23	0.65
OCSVM(V_{map})	**0.97**	0.10	**0.98**	**0.97**	0.16	0.97	0.88	0.17	0.92	0.90	0.13	0.94	0.88	0.18	0.92
OCSVM($Curv_{map}$)	0.96	0.17	0.95	0.96	0.21	0.93	0.90	0.20	0.91	0.91	0.22	0.91	0.90	0.23	0.89
OCSVM(Len_{map})	0.79	0.20	0.86	0.71	0.23	0.78	0.54	0.21	0.67	0.65	0.27	0.72	0.58	0.28	0.66

Outlier detection results for the *ECG data set* with five contamination levels c (columns). For each contamination level and each performance measure (sub-columns), we marked the best results in bold (*i.e.* highest correct detection rate ρ_c and AUC, and lowest false detection rate ρ_f). For all the contamination levels, the proposed method achieves the best results with V_{map} and the $Curv_{map}$. For our proposed method and a given mapping function and outlier-detection algorithm, performance does not degrade when c varies, whereas for $FUNTA$ and *Dir.out* it does. Our method outperforms state-of-the-art methods when there are few outliers.

is high, and *Dir.out* performs well when the contamination level is low. This shows that the outlying features captured by these mapping functions are more robust to the contamination level than those captured by the baselines.

II.5.7.2 PenDig data

From the results on the PenDig dataset in Table II.2, it can be seen that the proposed method always outperforms the baselines in terms of AUC. This implies that the baselines are not as effective in capturing shape outlying features. When the outliers are '0' digits, the results by the baselines are consistent with the results on the synthetic data when some shape outliers are simulated (Model 5 in Table II.4). This is not surprising, as Model 5 generates bivariate functional outliers with an elliptic shape in $\mathbb{R}^2$; hence, a zero-like shape ('0'). As an AUC value close to 0.50 implies that the detector performs as efficiently as a random method, we note that the '0' outlier case is the only in which the baselines are effective. The baseline methods cannot distinguish different shape outliers with abrupt shape irregularities such as (smooth) right angles, for example, when the outlier is the '1', '4', or '5' digit. In such cases, we obtain the best results in terms of AUC with V_{map}. For more regular shapes, such as '3', '6', '8', and '9', the best results are achieved by $Curv_{map}$. We conclude similarly for the ρ_c results. However, baselines are better in terms of ρ_f which shows that our method can confuse inliers with outliers and lack robustness of this dataset.

Table II.2: Results for the PenDig dataset

	Outlier '0'			Outlier '1'			Outlier '2'			Outlier '3'			Outlier '4'		
Methods	ρ_c	ρ_f	AUC	ρ_c	ρ_f	AUC	ρ_c	ρ_f	AUC	ρ_c	ρ_f	AUC	ρ_c	ρ_f	AUC
$FUNTA$ (baseline)	0.49	0.22	0.60	0.01	0.21	0.51	0.22	0.19	0.58	0.23	0.20	0.52	0.23	0.21	0.53
Dir.out (baseline)	0.72	**0.01**	0.82	0.24	**0.02**	0.52	0.75	0.42	0.60	0.00	**0.02**	0.55	0.00	**0.02**	0.58
iFor(V_{map})	0.78	0.05	0.87	0.44	0.38	**0.79**	0.86	0.15	0.63	**0.61**	0.45	0.66	0.74	0.09	**0.77**
iFor($Curv_{map}$)	**0.82**	0.12	**0.92**	0.43	0.60	0.61	**0.87**	0.47	0.57	0.57	0.38	**0.69**	0.81	0.33	0.63
iFor(Len_{map})	0.63	0.26	0.59	0.46	0.56	0.64	0.59	**0.12**	0.65	0.29	0.23	0.64	0.78	0.45	0.56
OCSVM(V_{map})	0.82	0.02	0.85	0.50	0.51	0.75	0.77	0.35	0.60	0.53	0.41	0.66	0.78	0.18	0.74
OCSVM($Curv_{map}$)	0.80	0.11	0.91	**0.50**	0.60	0.70	0.55	0.23	0.59	0.56	0.44	0.68	0.61	0.15	0.66
OCSVM(Len_{map})	0.81	0.10	0.75	0.37	0.42	0.70	0.84	0.18	**0.76**	0.54	0.42	0.67	**0.83**	0.25	0.69
	Outlier '5'			Outlier '6'			Outlier '7'			Outlier '8'			Outlier '9'		
Methods	ρ_c	ρ_f	AUC	ρ_c	ρ_f	AUC	ρ_c	ρ_f	AUC	ρ_c	ρ_f	AUC	ρ_c	ρ_f	AUC
$FUNTA$ (baseline)	0.49	0.22	0.60	0.01	**0.02**	0.51	0.22	**0.00**	0.58	0.23	**0.01**	0.51	0.23	0.21	0.53
Dir.out (baseline)	0.43	0.34	0.59	0.43	0.17	0.52	0.43	0.16	0.65	0.43	0.17	0.60	0.43	0.34	0.61
iFor(V_{map})	**0.69**	0.26	0.69	0.56	0.36	0.61	0.93	0.30	0.60	0.47	0.30	0.67	**0.92**	0.51	0.64
iFor($Curv_{map}$)	0.62	0.29	0.61	0.54	0.28	0.63	0.93	0.21	**0.68**	0.48	0.20	**0.77**	0.79	0.26	**0.73**
iFor(Len_{map})	0.42	0.13	0.61	0.47	0.21	**0.64**	**0.97**	0.29	0.65	0.40	0.08	0.77	0.74	0.40	0.63
OCSVM(V_{map})	0.59	**0.04**	**0.73**	0.55	0.38	0.56	0.87	0.22	0.60	**0.58**	0.45	0.63	0.70	0.25	0.70
OCSVM($Curv_{map}$)	0.58	0.18	0.64	0.61	0.40	0.61	0.86	0.19	0.62	0.56	0.44	0.66	0.62	**0.14**	0.72
OCSVM(Len_{map})	0.67	0.30	0.62	**0.62**	0.47	0.57	0.79	0.13	0.61	0.51	0.24	0.60	0.88	0.46	0.67

Outlier detection results for the *PenDig dataset* when each of the 10 classes ('0'..'9') is considered an outlier (columns), and the nine other classes inliers. For each case of outlier class and each performance measure (sub-columns), we marked the best results in bold. We note that our method achieve the best results in terms of AUC, showing that it better discriminates outliers.

II.5.7.3 AFT data

Table II.3: Results on the AFT dataset.

Methods	ρ_c	ρ_f	AUC
$FUNTA$ (baseline)	0.85	0.10	0.84
$Dir.out$ (baseline)	0.90	0.12	0.90
iFor(V_{map})	0.**98**	**0.01**	**0.97**
iFor($Curv_{map}$)	0.5	0.31	0.60
iFor(Len_{map})	**0.97**	**0.02**	**0.96**
OCSVM(V_{map})	0.95	0.03	0.96
OCSVM($Curv_{map}$)	0.52	0.34	0.55
OCSVM(Len_{map})	0.95	0.05	0.92

Outlier detection results for the AFT dataset with five contamination levels c.

We report the results on the AFT dataset in Table II.3 where one can see that our proposal outperforms the baselines with V_{map} and Len_{map} in terms of ρ_c, ρ_f and AUC. These results thus suggests that the outlying multivariate time series of the AFT dataset are underlied by a process whose duration and velocity are both abnormal. Also, since the $Curv_{map}$ does not enhance the discrimination between outliers and inliers, we can also say that they are underlied by a path whose *change* in direction (in $\mathbb{R}^{p=4}$) are similar.

II.5.7.4 Synthetic data

For all types of outliers (generated by a distinct contamination model), iFor(Len_{map}) achieves similar results to the baselines. For shape outliers (Model 5),the baselines are outperformed.

For isolated outliers (Table II.4, Model 2 and Model 4 columns), the results on the synthetic datasets show that the baseline methods perform as well as Len_{map}, and V_{map} with iFor. Moreover, since in this case the outlyingness is a short peak, the velocity quickly changes when the peak occurs; thus, the V_{map} function is an appropriate candidate for detecting isolated outliers. $Curv_{map}$

Table II.4: Results on the five synthetic datasets

	MODEL 1			MODEL 2			MODEL 3			MODEL 4			MODEL 5		
Methods	ρ_c	ρ_f	AUC	ρ_c	ρ_f	AUC	ρ_c	ρ_f	AUC	ρ_c	ρ_f	AUC	ρ_c	ρ_f	AUC
$FUNTA$ (baseline)	**1.00**	**0.00**	**1.00**	0.92	0.02	0.99	**0.96**	**0.00**	**1.00**	0.89	0.04	0.99	0.58	0.31	0.73
Dir.out (baseline)	**1.00**	**0.00**	**1.00**	**1.00**	**0.00**	**1.00**	0.91	**0.00**	**1.00**	**0.98**	**0.00**	**1.00**	0.88	**0.00**	**1.00**
iFor(V_{map})	0.99	**0.00**	**1.00**	0.91	0.02	**1.00**	0.69	0.25	0.82	0.77	0.16	0.92	0.83	0.13	0.94
iFor($Curv_{map}$)	0.61	0.30	0.75	0.57	0.48	0.60	0.59	0.39	0.67	0.57	0.48	0.61	0.73	0.24	0.85
iFor(Len_{map})	**1.00**	**0.00**	**1.00**	0.95	**0.00**	**1.00**	0.83	0.08	0.96	0.85	0.07	0.97	**0.96**	0.01	**1.00**
OCSVM(V_{map})	0.79	0.22	0.87	0.82	0.19	0.91	0.68	0.35	0.74	0.65	0.14	0.84	0.42	0.14	0.77
OCSVM($Curv_{map}$)	0.49	0.34	0.65	0.60	0.52	0.62	0.48	0.38	0.63	0.42	0.44	0.61	0.43	0.37	0.65
OCSVM(Len_{map})	0.66	0.10	0.82	0.83	0.07	0.91	0.59	0.16	0.78	0.62	0.07	0.84	0.50	0.06	0.83

Outlier detection results for the *synthetic data* generated by the five models (columns), as described in Section II.5.1.4. We compared the proposed methods, iFor($\cdot$) and OCSVM($\cdot$), with the two baselines, $FUNTA$ and *Dir.out*, in terms of three performance measures (in sub-columns): correct detection rate (ρ_c), false detection rate (ρ_f), and AUC. For each model and each performance metric, we marked in bold the best results (*i.e.* highest ρ_c and AUC, and lowest ρ_f).

shows poor performance for the two models. This implies that it is ineffective in detecting isolated outliers. Indeed, the contamination models (Model 2 and Model 4, Section II.5.1.4) generate stationary functional data (constant mean and only lag-dependent covariance) except in the part of $\mathcal{T}$ where the outlyingness (low short peak) occurs. Thus, considering the second-order variations (second-order derivatives in Equation (II.11)) is irrelevant and leads to high ρ_f values (ρ_f columns and $Curv_{map}$ rows).

For persistent magnitude outliers (Table II.4, Model 1 and Model 3 columns), *Dir.out* and $FUNTA$ yield the best results in terms of both ρ_c and ρ_f. We obtain similar results for Model 1 with V_{map}, and Len_{map} with iFor. Nevertheless, V_{map} is not as efficient for Model 3 as for Model 1. Indeed, Model 1 has high contamination (high short peak), resulting in high velocity mapping values, and we recall that velocity and curvature relate to local variations of the curves. Consequently, as here the magnitude outlyingness is a global feature, Len_{map} is better than V_{map} and $Curv_{map}$. This indicates that for detecting persistent magnitude outliers, the proposed approach is more reliable with Len_{map} than $Curv_{map}$ and V_{map}.

For persistent shape outliers (Table II.4, Model 5 column), the proposed method outperforms the baselines with iFor on Len_{map}. Furthermore, V_{map} yields results similar to those of *Dir.out* in terms of ρ_c and AUC. Table II.4 shows that the state-of-the-art $FUNTA$ totally fails to capture shape outlyingness because it is based on the intersection angles between the samples and is computed for each variable separately. Thus, it fails to consider the correlation between them (as explained in Section II.1).

For Models 1–5, among the proposed mapping functions, Len_{map} achieves the best results and appears to be suitable for outlier detection if the variables are weakly correlated, whereas V_{map} and $Curv_{map}$ are preferable if the correlation between the variables is strong.

As V_{map} and Len_{map} achieve satisfactory results, the geometric characterization (velocity and length) of the samples provides a different shape features in the outlier detection. We note that functional-data approxima-

tion affects the geometric characterization. Indeed, functional approximation enables smoothing out a curve and properly extracting derivative-based features because the induced smoothing renders the samples differentiable (see Section II.3.3); this is not a required property for the baselines *Dir.out* and $FUNTA$. Here, we carefully monitor the functional-approximation step using leave-one-out cross-validation (Equation 14). Thus, in contrast with the approximation step, the outlier-detection step depends greatly on the mapping-function computation.

Overall, considering the results on synthetic datasets, we recommend using Len_{map} in the case of (potential) persistent magnitude or shape outliers. In practice, Len_{map} does not directly indicate whether a sample is a shape or magnitude outlier. However, as shape and magnitude are quite distinctive outlyingness classes, the class of such an outlier can be known *a posteriori* by visual inspection or by setting a magnitude threshold with respect to the magnitude of the outliers detected. If the outliers are suspected to be isolated, we recommend using V_{map} and $Curv_{map}$, as both mapping functions extract local curve features in $\mathbb{R}^p$. In the case of a low contamination level, both OCSVM and iFor are suitable (even though on the ECG data, OCSVM is better for small c), whereas for high contamination levels, iFor is better.

We demonstrated that each mapping function can detect multiple classes of outliers. The results are consistent on both synthetic and real data. However, knowing the class of a outlier detected by a given mapping function is not easy, and this issue will be addressed in future work.

II.5.8 Statistical assessment of the results

We followed the hypothesis-testing procedure recommended in [Demsar 2006, García *et al.* 2010] (also used in [Erfani *et al.* 2016]) to assess the statistical significance of the results introduced above. Demsar provided an evaluation protocol for a more general assessment of the difference between several classifiers used on multiple benchmark datasets. The protocol consists of two steps: First, a global significance test is conducted to determine whether

there is a difference among the evaluated methods. Second, if this is the case, the methods are pairwise compared to evaluate the gain of one over another.

We applied Demsar's protocol because the present detection task reduces to a two-class classification in the evaluation step (outliers/inliers). We applied the protocol for the three performance measures ρ_c, ρ_f, and AUC separately. As described in [Demsar 2006, García *et al.* 2010], there are several ways of conducting the tests and we primarily applied it as in [Erfani *et al.* 2016]. We applied the protocol as follows:

(i) First, the Friedman test [Sheskin 2003] was applied to detect the global statistical significance for each of the three performance measures among all the methods on all the datasets. The Friedman test can be viewed as the non-parametric version of ANOVA (where, here, a group refers to a method, and the samples in the group refer to the performance of the method on the datasets), as it is based on the ranks and thus does not make the Gaussian assumption for the performance measures for each method [Demsar 2006]. We conducted the Friedman test with the Iman–Davenport correction [Sheskin 2003], as recommended in [Demsar 2006], to handle the well-known family-wise error rate, which can bias the p-value in a multiple-hypothesis test. We recall that in the present context, the family-wise error rate refers to the probability of erroneously asserting that one method is more reliable for detecting outliers than some of the others.

(ii) Second, if statistical significance was detected by the Friedman test, we performed a post-hoc test to determine which methods are different. More precisely, the post-hoc test is based on the p-values returned by a pairwise-comparison test applied to all pairwise comparisons of the methods. A nonparametric test can be selected for the pairwise comparisons (owing to the absence of the Gaussian assumption), such as the post-hoc Friedman's aligned ranked test [García *et al.* 2010]. As the all-pairwise-comparisons test is a special case of a multiple-hypothesis

test, it also suffers from the family error rate and requires a correction procedure. Thus, we used the Finner correction as recommended in [García *et al.* 2010].

We separately applied this evaluation protocol to the three performance measures for the five contamination levels of the ECG data, the 10 outlier classes of the PenDig dataset, and the synthetic data to compare the methods on two distinct types of data and to demonstrate the benefit of the proposed approach on real data. Moreover, this enables assessing the difference of the methods in a given context (*i.e.* when the outlier class is known). For all datasets, we used a significance level of 0.1, as in [Erfani *et al.* 2016].

We report the average ranking (vertical axis) of all methods (horizontal axis) applied to the ECG and PenDig datasets (resp., synthetic data) for each performance measure (colors) in a vertical-bar plot in Figure II.6 (resp., Figure II.7). Each bar has a height equal to its average rank (1 is the best, 8 is the worst) based on the post-hoc Friedman's aligned rank test across the five contamination levels (resp., five models). For ρ_c and AUC, the ranking is given in decreasing order, and for ρ_f, the ranking is given in increasing order. The above number of bars refers to the global ranking (*i.e.* ranks from the average ranks).

The Friedman test yielded to the rejection of equality of the methods for the ECG, PenDig and synthetic datasets, for each performance measure (p-values are given in the discussion). Therefore, we conducted pairwise comparison (post-hoc) tests. The significance of the pairwise comparison tests of ρ_c, ρ_f, and AUC for the ECG and PenDig datasets is given in Tables II.5, II.6, and II.7, and for the synthetic data, in Tables II.8, II.9, and II.10. The significance (at level 0.1) of a test is indicated by $\neq^*$, and non-significance is indicated by $=$.

ECG data. The Friedman test rejects the null hypothesis of equivalence of the methods for the three performance measures at a significance level of 0.1. The p-values are 3.0×10^{-10} for the correction detection rate ρ_c, 3.0×10^{-10}

for the false detection rate ρ_f, and 2.2×10^{-16} for AUC. Thus, we conducted a post-hoc test. Figure II.6 shows the average ranking of the methods based on the Friedman's aligned rank test (from the best 1 to the worst 8). The p-value of each pairwise comparison in the post-hoc test is given in Tables II.5, II.6, and II.7 for the correction detection rate, the false detection rate, and AUC, respectively, where a cell indicates whether the resulting p-value of the pairwise comparison test of the methods in the corresponding row and column is below of above the significance level. The symbol = indicates a p-value greater than the significance level of 0.1, allowing the *acceptance* of the null hypothesis of equivalence of the two methods; rejection is indicated by $\neq^*$.

Based on the results in Figure II.6 and Tables II.5, II.6, and II.7, it is concluded that both V_{map} and $Curv_{map}$ outperform the baselines in terms of the three performance measures. We notice that *Dir.out* is not significantly better than the methods with the worst performance (*i.e.* iFor(Len_{map}), $FUNTA$, and OCSVM(Len_{map})). $FUNTA$ is not significantly different from iFor($Curv_{map}$) and OCSVM($Curv_{map}$) (Tables II.5 and II.7, $FUNTA$ rows and columns). Thus, by considering the results on the ECG data (Table II.1 and Figure II.5), which show that $FUNTA$ is almost as effective as iFor($Curv_{map}$) and OCSVM($Curv_{map}$) in terms of ρ_c when the contamination level is high ($c \geq 15\%$), this qualitative comparison is confirmed by the non-significance of the difference with OCSVM($Curv_{map}$). However, in terms of ρ_f, $FUNTA$ is ineffective and is outperformed by iFor(V_{map}), iFor($Curv_{map}$), *Dir.out*, and OCSVM($Curv_{map}$) (Table II.6). Even though Len_{map} yields the worst results among the three proposed mapping functions with both iFor and OCSVM (Table II.1, Figure II.6), it is not significantly different from *Dir.out* (see *Dir.out* columns and Len_{map} rows in Tables II.5 and II.7).

PenDig data. The Friedman test rejects the null hypothesis of equivalence of the methods for the three performance measures at a significance level of 0.1. The p-values are 1.5×10^{-1} for the correct detection rate, 2.8×10^{-9} for the false detection rate, and 1.1×10^{-4} for AUC. We note that there

Table II.5: Statistical significance of the pairwise comparisons for the correct detection rate ρ_c on the ECG (upper table) and PenDig (lower table) datasets.

	$FUNTA$	*Dir.out*	iFor			OCSVM		
			V_{map}	$Curv_{map}$	Len_{map}	V_{map}	$Curv_{map}$	Len_{map}
$FUNTA$ (baseline)	x	=	=	=	$\neq^*$	=	=	=
Dir.out (baseline)	-	x	$\neq^*$	$\neq^*$	=	$\neq^*$	$\neq^*$	=
iFor(V_{map})	-	-	x	=	$\neq^*$	=	=	$\neq^*$
iFor($Curv_{map}$)	-	-	-	x	$\neq^*$	=	=	$\neq^*$
iFor(Len_{map})	-	-	-	-	x	$\neq^*$	$\neq^*$	=
OCSVM(V_{map})	-	-	-	-	-	x	=	$\neq^*$
OCSVM($Curv_{map}$)	-	-	-	-	-	-	x	$\neq^*$
OCSVM(Len_{map})	-	-	-	-	-	-	-	x

	$FUNTA$	*Dir.out*	iFor			OCSVM		
			V_{map}	$Curv_{map}$	Len_{map}	V_{map}	$Curv_{map}$	Len_{map}
$FUNTA$ (baseline)	x	=	$\neq^*$	$\neq^*$	$\neq^*$	$\neq^*$	$\neq^*$	$\neq^*$
Dir.out (baseline)	-	x	$\neq^*$	$\neq^*$	=	$\neq^*$	$\neq^*$	$\neq^*$
iFor(V_{map})	-	-	x	=	$\neq^*$	=	=	=
iFor($Curv_{map}$)	-	-	-	x	=	=	=	=
iFor(Len_{map})	-	-	-	-	x	=	=	=
OCSVM(V_{map})	-	-	-	-	-	x	=	=
OCSVM($Curv_{map}$)	-	-	-	-	-	-	x	=
OCSVM(Len_{map})	-	-	-	-	-	-	-	x

$\neq^*$ indicates that the corresponding methods in the row and the column of the cell are significantly different at a level of 0.1, and = indicates that they are not. The lower triangular part was replaced by dashes because it is equal to the upper part.

Table II.6: Statistical significance of the pairwise comparisons for the false detection rate ρ_f on the ECG (upper table) and PenDig datasets (lower table).

	$FUNTA$	$Dir.out$	iFor			OCSVM		
			V_{map}	$Curv_{map}$	Len_{map}	V_{map}	$Curv_{map}$	Len_{map}
$FUNTA$ (baseline)	x	=	=	=	$\neq^*$	=	=	=
$Dir.out$ (baseline)	-	x	$\neq^*$	$\neq^*$	=	$\neq^*$	$\neq^*$	=
iFor(V_{map})	-	-	x	=	$\neq^*$	=	=	$\neq^*$
iFor($Curv_{map}$)	-	-	-	x	$\neq^*$	=	=	$\neq^*$
iFor(Len_{map})	-	-	-	-	x	$\neq^*$	$\neq^*$	=
OCSVM(V_{map})	-	-	-	-	-	x	=	$\neq^*$
OCSVM($Curv_{map}$)	-	-	-	-	-	-	x	$\neq^*$
OCSVM(Len_{map})	-	-	-	-	-	-	-	x

	$FUNTA$	$Dir.out$	iFor			OCSVM		
			V_{map}	$Curv_{map}$	Len_{map}	V_{map}	$Curv_{map}$	Len_{map}
$FUNTA$ (baseline)	x	=	$\neq^*$	$\neq^*$	$\neq^*$	$\neq^*$	$\neq^*$	$\neq^*$
$Dir.out$ (baseline)	-	x	$\neq^*$	$\neq^*$	=	=	$\neq^*$	=
iFor(V_{map})	-	-	x	=	=	=	=	=
iFor($Curv_{map}$)	-	-	-	x	=	=	=	=
iFor(Len_{map})	-	-	-	-	x	=	=	=
OCSVM(V_{map})	-	-	-	-	-	x	=	=
OCSVM($Curv_{map}$)	-	-	-	-	-	-	x	=
OCSVM(Len_{map})	-	-	-	-	-	-	-	x

Notation is the same as in Table II.5.

Table II.7: Statistical significance of the pairwise comparisons for the AUC on the ECG (upper table) and PenDig (lower table) datasets.

	$FUNTA$	$Dir.out$	iFor			OCSVM		
			V_{map}	$Curv_{map}$	Len_{map}	V_{map}	$Curv_{map}$	Len_{map}
$FUNTA$ (baseline)	x	=	=	=	$\neq^*$	=	=	=
$Dir.out$ (baseline)	-	x	$\neq^*$	$\neq^*$	=	$\neq^*$	$\neq^*$	=
iFor(V_{map})	-	-	x	=	$\neq^*$	=	=	$\neq^*$
iFor($Curv_{map}$)	-	-	-	x	$\neq^*$	=	=	$\neq^*$
iFor(Len_{map})	-	-	-	-	x	$\neq^*$	$\neq^*$	=
OCSVM(V_{map})	-	-	-	-	-	x	=	$\neq^*$
OCSVM($Curv_{map}$)	-	-	-	-	-	-	x	$\neq^*$
OCSVM(Len_{map})	-	-	-	-	-	-	-	x

	$FUNTA$	$Dir.out$	iFor			OCSVM		
			V_{map}	$Curv_{map}$	Len_{map}	V_{map}	$Curv_{map}$	Len_{map}
$FUNTA$ (baseline)	x	=	$\neq^*$	$\neq^*$	$\neq^*$	$\neq^*$	$\neq^*$	$\neq^*$
$Dir.out$ (baseline)	-	x	$\neq^*$	=	=	=	=	=
iFor(V_{map})	-	-	x	=	=	=	=	=
iFor($Curv_{map}$)	-	-	-	x	=	=	=	=
iFor(Len_{map})	-	-	-	-	x	=	=	=
OCSVM(V_{map})	-	-	-	-	-	x	=	=
OCSVM($Curv_{map}$)	-	-	-	-	-	-	x	=
OCSVM(Len_{map})	-	-	-	-	-	-	-	x

Notation is the same as in Table II.5

is consistency with respect to the ECG data except for the false detection rate ρ_f. Indeed, both V_{map} and $Curv_{map}$ outperform the baselines in terms of ρ_c and AUC (Tables II.5 and II.7). Moreover, among the three mapping functions, Len_{map} yields the worst results and is not different from *Dir.out*. However, there is an inconsistency ranking regarding ρ_f in the PenDig data with respect to the ECG data (Figure II.6 and Table II.6). Indeed, as the proposed method is not ranked first in terms of the false detection rate, it may be claimed that it recognizes the outliers but tends to be excessively severe.

We note that this conclusion regarding the correct and false detection rates is drawn according to the adopted outlyingness thresholding rule, which can be modified, as discussed at the end of Section II.5.3.

From the global ranking (Figure II.6) and the pairwise comparison tests, it may be concluded that the proposed method outperforms the baselines on both the ECG and PenDig datasets at the significance level 0.1.

Synthetic data. Regarding the synthetic data, the Friedman test rejects the null hypothesis of equivalence of the methods for the three performances measures at a significance level of 0.1. The p-value is 2.4×10^{-10} for the correct detection rate, 2.4×10^{-10} for the false detection rate, and 1.0×10^{-6} for AUC. As these p-values are below 0.1, we conducted a post-hoc test to compare the methods pairwise and assess the gain of one over another. Figure II.7 shows the average ranking of the methods according to the post-hoc Friedman's aligned rank test.

The p-values of each pairwise comparison test is given in Tables II.8, II.9, and II.10 for ρ_c, ρ_f, and AUC, respectively. We notice that *Dir.out* is significantly equivalent to iFor(Len_{map}), OCSVM(Len_{map}), $FUNTA$, and iFor($Vmap$), and these methods are ranked first, second, and third on average, respectively (Figure II.7). Thus, on the synthetic dataset, the baseline methods are slightly better than the proposed method; however, based on the pairwise comparison tests, the best methods ($iFor(Len_{map})$ and $OCSVM(Len_{map})$)

are statistically equivalent. As discussed in the two previous paragraphs, the proposed method is superior on real datasets. Moreover, in the iFor rows and OCSVM columns, it can be seen that there is a pairwise equivalence between iFor and OCSVM for (Len_{map}) and (V_{map}), that is, these two outlier-detection algorithms are empirically consistent for a given mapping function. Therefore, we have equivalent methods to achieve state-of-the-art results (which cannot be improved, except for MODEL 5) for the synthetic data.

Overall assessment. Tables II.5, II.6, and II.7 (in the iFor rows and OCSVM columns) show the pairwise consistency between the iFor and OCSVM algorithms for each mapping function. The same holds for the synthetic data. Thus, for a given dataset and mapping function, iFor and OCSVM achieve *statistically* the same performance results. This implies that the detection performance relies more on the outlying features provided by the mapping function than on the capacity of the outlier-detection algorithm to discover outlying features itself.

Table II.8: Statistical significance of the pairwise comparisons for the correct detection rate ρ_c on the synthetic datasets.

	FUNTA	*Dir.out*	iFor			OCSVM		
			V_{map}	$Curv_{map}$	Len_{map}	V_{map}	$Curv_{map}$	Len_{map}
FUNTA (baseline)	x	=	=	$\neq^*$	=	=	$\neq^*$	=
Dir.out (baseline)	-	x	=	$\neq^*$	=	$\neq^*$	$\neq^*$	$\neq^*$
iFor(V_{map})	-	-	x	=	=	=	$\neq^*$	=
iFor($Curv_{map}$)	-	-	-	x	$\neq^*$	=	=	=
iFor(Len_{map})	-	-	-	-	x	=	$\neq^*$	$\neq^*$
OCSVM(V_{map})	-	-	-	-	-	x	=	=
OCSVM($Curv_{map}$)	-	-	-	-	-	-	x	=
OCSVM(Len_{map})	-	-	-	-	-	-	-	x

Notation is the same as in Table II.5

Table II.9: Significance of the pairwise comparisons for the false detection rate ρ_f on the synthetic dataset.

	$FUNTA$	*Dir.out*	iFor(V_{map})	iFor($Curv_{map}$)	iFor(Len_{map})	OCSVM(V_{map})	OCSVM($Curv_{map}$)	OCSVM(Len_{map})
$FUNTA$ (baseline)	x	=	=	$\neq^*$	=	=	$\neq^*$	=
Dir.out (baseline)	-	x	=	$\neq^*$	=	$\neq^*$	$\neq^*$	=
iFor(V_{map})	-	-	x	=	=	=	$\neq^*$	=
iFor($Curv_{map}$)	-	-	-	x	$\neq^*$	=	=	=
iFor(Len_{map})	-	-	-	-	x	=	$\neq^*$	=
OCSVM(V_{map})	-	-	-	-	-	x	=	=
OCSVM($Curv_{map}$)	-	-	-	-	-	-	x	$\neq^*$
OCSVM(Len_{map})	-	-	-	-	-	-	-	x

Notation is the same as in Table II.5

Table II.10: Significance of the pairwise comparisons for AUC on the synthetic dataset.

	$FUNTA$	$Dir.out$	iFor(V_{map})	iFor($Curv_{map}$)	iFor(Len_{map})	OCSVM(V_{map})	OCSVM($Curv_{map}$)	OCSVM(Len_{map})
$FUNTA$ (baseline)	x	=	=	$\neq^*$	=	=	$\neq^*$	=
$Dir.out$ (baseline)	-	x	=	$\neq^*$	=	$\neq^*$	$\neq^*$	$\neq^*$
iFor(V_{map})	-	-	x	$\neq^*$	=	=	$\neq^*$	=
iFor($Curv_{map}$)	-	-	-	x	$\neq^*$	=	=	=
iFor(Len_{map})	-	-	-	-	x	$\neq^*$	$\neq^*$	$\neq^*$
OCSVM(V_{map})	-	-	-	-	-	x	=	=
OCSVM($Curv_{map}$)	-	-	-	-	-	-	x	=
OCSVM(Len_{map})	-	-	-	-	-	-	-	x

Notation is the same as in Table II.5

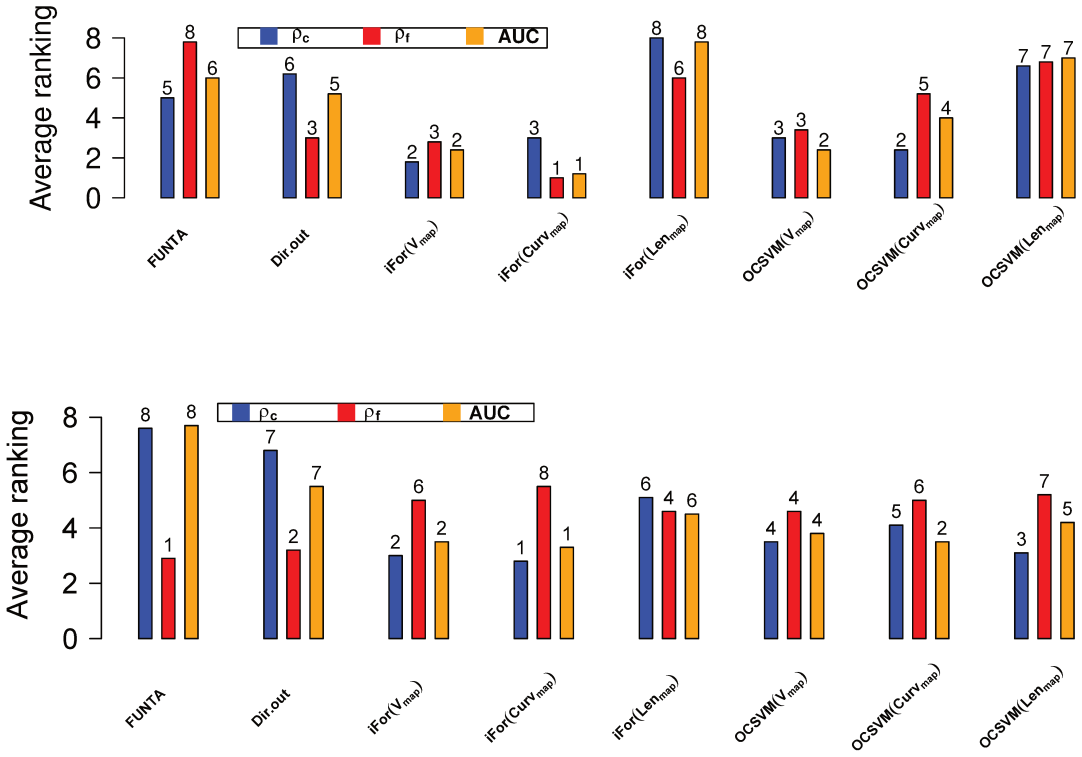

Figure II.6: (1 is the best, 8 the worst) for ρ_c, ρ_f, and AUC based on the post-hoc Friedman's aligned rank test, considering the five contamination levels in the *ECG data* (upper bar plot) and the *PenDig data* (lower bar plot). For ρ_c and AUC, the ranking is given in decreasing order (*i.e.* for high ρ_c and AUC values, the rank tends to 1); for ρ_f, the ranking is given in increasing order (*i.e.* for low ρ_f values, the rank tends to 1). The y-axis represents the average ranking over the five models, and the integers on the top of the bars represent the final ranking. If there are ties, we take the average ranking.

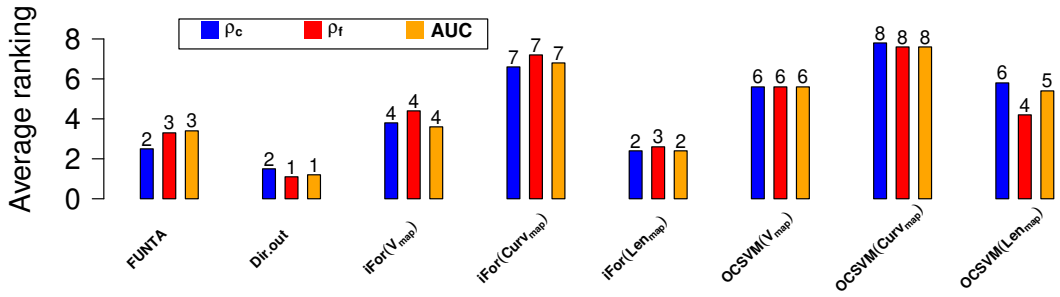

Figure II.7: (1 is the best, 8 is the worst) for ρ_c, ρ_f, and AUC based on the post-hoc Friedman's aligned rank test. For ρ_c and AUC, the ranking is given in decreasing order (*i.e.* for high ρ_c and AUC values, the rank is close to 1), and for ρ_f, the ranking is given in increasing order (*i.e.* for low ρ_f values, the rank is close to 1). The y-axis represents the average ranking over the five models, and the integers on the top of the bars represent the final ranking. If there are ties, we take the average ranking. The baseline methods are slightly better than the proposed method, but the best results by the proposed method ($iFor(Len_{map})$ and $OCSVM(Len_{map})$) are statistically equivalent to those by the baseline, as showed by the pairwise comparison tests in Tables 8,9, and 10.

II.6 Conclusion

In this chapter, we proposed a method to improve the detection of different types of outliers in multivariate functional data, based on curve shape. We assumed that the original discrete curves can be well approximated by finite functional basis expansions, where the basis is specified. Based on the smooth reconstruction provided by the fitted basis expansion, we used the arc-length, velocity, and curvature mapping functions to capture latent shape features. Then, we detected the outliers from the mapped curves using outlier-detection algorithms.

Through an experimental study on real and synthetic datasets, we showed that the proposed approach outperforms multivariate functional depth baselines on real data and can perform similarly on synthetic data (except for persistent shape outliers, where the proposed method performs better). We demonstrated that, compared with the baselines, the proposed approach is robust to the variation of the contamination level.

We did not assume any weighting of the curve variables in the mapping functions; this is left as future work. This weighting could be user-driven, as proposed for functional depth in [Claeskens *et al.* 2014], or data-driven. It is conceivable that this can enhance outlier detection in the presence of non-outlying curve variables (when p increases). Another possible improvement would be to combine mapping functions in the same detector so that multiple outlier classes may be detected in the same dataset.

II.7 Publications

We shall mention that the work of Chapter II has been published in two venues. We first published the short paper [Lejeune *et al.* 2020b] in the international Conference *Extended Data Base Technology*. In this paper, we used the curvature mapping function on the ECG data set and showed empirically that our method is more robust than some baselines in terms of contamina-

tion rate. From repeated trials, we also showed that our method provides predictions with lower uncertainties than the baselines. We secondly published the journal paper [Lejeune *et al.* 2020a] in the *Knowledge-based Systems* journal. In this paper, we extended our work in [Lejeune *et al.* 2020b] by proposing the arc-length and the velocity mapping functions. We validated our method on synthetic datasets and on the ECG and PenDig real datasets.

Chapter III

Data-driven discovery of systems of ordinary differential equations with nonconvex multi-task learning

Abstract.

In this chapter, we tackle the problem of analytical discovery of systems of ordinary differential equations (SODE) from a multivariate time series. Once known, the analytical form of such model provides a quantitative representation of the dynamics underlying an observed phenomenon. The problem is encompassed by the recent framework known as the *sparse identification of nonlinear dynamics* [Brunton *et al.* 2016]. In state of the art, the problem is formalized as a linear regression whose weights are learned with the sparse convex LASSO penalty. However, this penalty does not take into account the coupling between the equations of the SODE and, due to its convexity, gives a biased estimate of the weights. To address these two limitations, we re-cast the problem as a multi-task learning one involving a nonconvex penalty. Numerical experiments, on known SODEs, show that both the multi-task and the nonconvexity features of our method outperform state of the art. We also apply our algorithm on an Airbus flight test dataset.

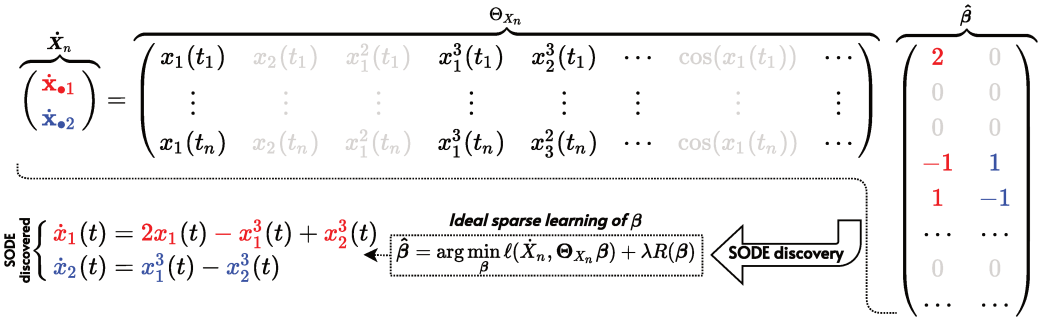

Figure III.1: (from top to center to bottom-left) Illustration of the data-driven discovery of a two dimensional SODE with two (joint) linear regressions (linear with respect to $\boldsymbol{\beta}$). Top: based on samples of state-variables time derivative ($\dot{\boldsymbol{X}}_n$'s columns), and resultant of nonlinear transformations of samples of the state-variables (columns of the dictionary $\boldsymbol{\Theta}_{X_n}$), a linear model is assumed between the two latters. Center: the discovery a the SODE is performed by minimizing a learning criterion involving a sparse penalty R. Bottom-left: the SODE is identified with the resulting learned coefficient matrix $\boldsymbol{\beta}^*$.

III.1 Introduction

Governing equations are mathematical models, such as partial differential equations (*e.g.* Navier-Stokes equation for fluid dynamic modeling) or systems of ordinary differential equations (SODE), widely used in sciences and engineering to model dynamic systems [Butcher 2016]. A governing equation models the dependency relationship between several parameters, named *state-variables*, (*e.g.* velocity, chemical concentration, temperature, etc.), in a dynamic system.

A SODE is a system of equations where each one models the dependency relationship between several scalar-dependent state-variables and one state-variable first order derivative. To illustrate a SODE, we give an example of a two dimensional SODE in Figure III.1 bottom-left wherein one can see that the two equations entail both state variables x_1 and x_2 through cubic monomials and the first equation has in addition a linear monomial. The solution of a governing equation depicts the temporal and/or spatial evolution

of the state-variables in a dynamic system. Famous examples of SODEs are for example the FitzHugh-Nagumo equation to model neural excitement in biology, the damped harmonic oscillator equation in mechanics or the Lotka-Volterra equation to model population dynamics [Ramsay & Hooker 2017].

Governing equations are traditionally derived from principled rules themselves formalized from general empirical observations under certain hypotheses. For instance, the SODE of the damped harmonic oscillator is derived from the Newton's second law under the constant-mass hypothesis [Greiner 2006].

However, there remains some complex systems (*e.g.* in mechanical engineering, fluid mechanics) whose dynamic behavior is poorly understood and quite hard to be modeled within a governing equation derived from existing principled rules. Hence, the SODE underlying such systems is unknown. Accessing to the model, like a SODE, that governs an unknown dynamic is a challenging task of scientific interest to improve the understanding of a physical system as well as of practical interest to get a simulation model *e.g.* for prototype design [Brunton *et al.* 2016, Schaeffer & McCalla 2017]. In this chapter, we address such a challenge and propose a data-driven method to solve it based on multi-task learning (MTL). In Figure III.1, we illustrate the core of both our and state of the art methods: (top) based on data ($\dot{\boldsymbol{X}}_n$ and $\boldsymbol{\Theta}_{X_n}$) sampled from a dynamic system, solving for the optimization problem involving a sparse penalty R (bottom-center) leads to the identification of a SODE (bottom-left). The learned SODE strongly relies on the chosen penalty. We propose a nonconvex sparse penalty to learn the matrix coefficient $\boldsymbol{\beta}$ that accounts for correlations within the SODE. The nonconvexity of such a regularizer results in a penalty with stronger sparsity than convex regularizers leading to better selection of the candidate functions. Furthermore, this penalty provides in a SODE learned more accurately than with state of the art penalty.

Thanks to the development of sensor technology, many data can be sampled from a dynamic system. Such an amount of data gives new opportunities to

extract important knowledge on the underlying physical behavior of a dynamic system. Consequently, there has been a growing interest over the last years in the development of data-driven methods for the discovery of governing equations [Brunton *et al.* 2016, Long *et al.* 2018, Bhat & Rawat 2019, Schaeffer 2017, Zhang & Schaeffer 2019].

State-of-the-art methods for the discovery of SODE are based on the matrix-extended LASSO (the $\|\cdot\|_{1,1}$ norm) sparsity-promoting learning methods [Brunton *et al.* 2016, Schaeffer & McCalla 2017, Rudy *et al.* 2019, Tishbirani 1996]. In Figure III.1, we illustrate and generalize the core idea of these methods: such kind of methods takes as input samples of state-variables time derivative and a dictionary of resultants of (arbitrary) nonlinear transformations of state-variables samples. Then the algorithm outputs the relevant elements of the dictionary that best model the relationship between the (samples of) state-variables and their associated first order derivatives. Actually, learning in this way, *i.e.* with the $\|\cdot\|_{1,1}$ norm penalty, reduces to single-task learning, where a task refers to the discovery of a single equation. However, (i) single-task learning does not accommodate the relatedness between the equations of the SODE [Obozinski *et al.* 2010]. We give an example of such relatedness in Figure III.1 bottom-left where the occurrence of x_1^3 and x_2^3, within both equations, makes the SODE *coupled*. Consequently, issue (i) results in an over-complete (*i.e.* not sparse enough) or under-complete (too sparse) set of selected dictionary elements to identify the SODE.

Moreover, since the LASSO penalty is convex, (ii) the learned coefficients associated to each element of the dictionary result biased [Zhang 2010, Fan & Li 2001].

To remedy both (i) and (ii), we propose to cast the problem of discovering a SODE as a MTL problem [Argyriou *et al.* 2008, Caruana 1997, Obozinski *et al.* 2010] involving nonconvex regularizer that promotes sparsity, relatedness between the equations of the SODE and encourages unbiasedness.

Our contribution is to learn a SODE from data samples of a dynamic system

with sparsity-promoting algorithm in the same spirit as [Brunton *et al.* 2016, Schaeffer & McCalla 2017, Rudy *et al.* 2019], while preventing both inconsistent sparsity and bias issues. Actually, learning a SODE can be seen as a problem made up of p regression tasks. The k-th learning task consists to learn the k-th component of the SODE from a large dictionary of linear and nonlinear functions built from the raw data *i.e.* the samples of the state-variables of the dynamic system under study. Since, by definition of a SODE, there are as many tasks as state-variables, the regression problem results in a MTL problem. Contrary to single-task learning, MTL takes benefit from the correlation between the tasks to learn a consistent set of predictors across tasks in order to improve predictive performances of each of of them [Argyriou *et al.* 2008, Obozinski *et al.* 2010]. In other words, when applied to the discovery of a SODE, MTL amounts to learn a sparse set of elements from a dictionary of candidate functions for each equation by considering the coupling feature of the SODE.

We sum up our two-fold contribution:

- We cast the discovery of SODE as a nonconvex MTL problem. We formalize the learning as an optimization problem involving a matrix-structured, sparse and nonconvex regularizer to account both for task relatedness and unbiasedness in the learned SODE. To perform the learning, we instantiate an efficient generic algorithm [Gong *et al.* 2013].
- Through experiments on benchmark of reference SODEs, we show that learning with a nonconvex multi-task penalty enables a better recovery of the underlying equations than learning with a convex single-task penalty.

III.2 Related work

We focus on the case where the state-variables are sampled from a dynamic system along a scalar variable $t \in \mathbb{R}$ that, without loss of generality, refers to time. Thus each one of the sampled state-variables form a time series. To discover the dynamic relationship between the state-variables, one can assume that the data in underlied by the solution of an unknown SODE as proposed by [Brunton *et al.* 2016]. Discovering the SODE from the data enables the practitioner to better understand the underlying dynamic of the phenomenon. We emphasize that, contrary to recent work on deep learning algorithm driven by SODE solver [Chen *et al.* 2018], we aim to discover a SODE from data samples and not to solve a given SODE since the later is unknown by definition of the problem.

For clarity, let's recall how a SODE is formalized. Let $f = \Big[f_1, \ldots, f_p\Big]^\top : \mathbb{R}^p \to \mathbb{R}^p$ be a Lipschitz continuous map defining the evolution of a state-variable $\mathbf{x}(t) = \Big[x_1(t), \ldots, x_p(t)\Big]^\top \in \mathbb{R}^p$, a SODE is expressed as $D\mathbf{x}(t) = \dot{\mathbf{x}}(t) = f(\mathbf{x}\Big(t\Big)\Big)$, where $\dot{\mathbf{x}}$ refers to the first-order time derivative of $\mathbf{x}$ (we use the 'dot' notation instead of D for a better readiness), or equivalently:

$$\begin{cases} \dot{x}_1(t) = f_1\Big(x_1(t), \ldots, x_p(t)\Big) \\ \vdots \\ \dot{x}_p(t) = f_p\Big(x_1(t), \ldots, x_p(t)\Big) \end{cases} \tag{III.1}$$

In Figure III.1 bottom-left, we instantiate Equation (III.1) with $p = 2$ and f polynomial in $\mathbf{x}$. Hence, discovering a SODE boils down to learn f in Equation (III.1) from noisy samples of $(\dot{\mathbf{x}}, \mathbf{x})$. Since the state-variables are related to each other, they can appear in multiple equations of the SODE. Thus, the p equations are coupled each other. We improve the discovery of a SODE by taking benefit of this feature through MTL.

The seminal work of learning a SODE from data samples traces back to [Schmidt & Lipson 2009]. In their paper, Schmidt and Lipson proposed a

combinatorial approach based on genetic-programming to select the parsimonious model, among a large set of candidate models, that best recovers the data. As mentioned in [Brunton *et al.* 2016], genetic-programming methods do not scale to large data sets and are inclined to over-fitting. To remedy it, Brunton et al. re-casted the learning of a SODE as sparse regression problem. Such a modern problem formulation has recently formed a new general framework referred as the sparse identification of nonlinear dynamics in the literature.

III.2.1 Building block for sparse learning of a SODE

State-of-the-art methods for the discovery of SODE, [Brunton *et al.* 2016, Rudy *et al.* 2019], assume that each one of $\dot{x}_1, \ldots, \dot{x}_p$ in Equation (III.1) are independent targets which can be predicted by a sparse combination of elements comprised in a dictionary of candidate functions. See Figure III.1 for an example of a two-dimensional SODE and where the dictionary is denoted $\boldsymbol{\Theta}_{X_n}$. The dictionary is first specified by the user by building linear as well as nonlinear candidate functions (*e.g.* polynomials) from the noisy samples $\boldsymbol{X}_n$ of $\mathbf{x}$. This dictionary reflects the prior knowledge on the observed phenomenon and is possibly over-complete. In [Brunton *et al.* 2016, Schaeffer & McCalla 2017, Rudy *et al.* 2019], f is assumed to be linear with respect to (w.r.t) the elements of the dictionary (and not w.r.t t). The linear assumption on f w.r.t $x_1, \ldots, x_p$ makes it easy to learn and to interpret. Then to learn the SODE, a sparsity-promoting algorithm, *e.g.* LASSO [Tishbirani 1996], elastic-net [Zou & Hastie 2005], learns $f_1, \cdots, f_p$ separately by both selecting the relevant elements (*i.e.* a small set of candidate functions) of the dictionary and estimating their associated coefficient in the linear model.

III.2.2 Discovery of a SODE by sparse linear regression

The first step of the approach introduced above can be described as follows [Brunton *et al.* 2016]. Starting from n noisy samples of a p-dimensional state-variable comprised in $\boldsymbol{X}_n$ and the associated time-derivatives samples $\dot{\boldsymbol{X}}_n = \left[\dot{\mathbf{x}}_{\bullet 1}, \dot{\mathbf{x}}_{\bullet 2}, \ldots, \dot{\mathbf{x}}_{\bullet p}\right]$ (which can be computed numerically if they were not sampled), one first builds an arbitrary dictionary of m candidate functions $\boldsymbol{\Theta}_{X_n} = \left[\mathbf{x}_{\bullet 1}, \mathbf{x}_{\bullet 2}, \ldots, \mathbf{x}_{\bullet 1}^2, \mathbf{x}_{\bullet 2}^2, \ldots, \cos \mathbf{x}_{\bullet 1}, \ldots\right] \in \mathbb{R}^{n \times m}$. Then from the linear assumption on f, *i.e.* $\dot{\boldsymbol{X}}_n = f\left(\boldsymbol{X}_n\right) = \boldsymbol{\Theta}_{X_n}\boldsymbol{\beta}$ where $\boldsymbol{\beta} \in \mathbb{R}^{m \times p}$ is a matrix wherein the q-th column refers to the coefficient-vector associated to the candidate functions of the q-th SODE component (see Equation (III.1)), one can find a sparse $\boldsymbol{\beta}^*$ by minimizing a loss (data fidelity term) plus a sparsity-promoting term:

$$\boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^{m \times p}} \ell(\dot{\boldsymbol{X}}_n, \boldsymbol{\Theta}_{X_n}\boldsymbol{\beta}) + \lambda R(\boldsymbol{\beta}) \tag{III.2}$$

$\lambda > 0$ is the sparsity amount hyperparameter. We illustrate our approach in Figure III.1 with a two-dimensional SODE.

To learn the SODE, [Brunton *et al.* 2016, Rudy *et al.* 2019, Schaeffer 2017] instantiate Problem (III.2) by choosing R as the $\ell_{1,1}$ norm *i.e.* $\boldsymbol{\beta}^*$ is the solution of the following problem:

$$\boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^{m \times p}} \frac{1}{2}\|\dot{\boldsymbol{X}}_n - \boldsymbol{\Theta}_{X_n}\boldsymbol{\beta}\|_F^2 + \lambda\|\boldsymbol{\beta}\|_{1,1} \tag{III.3}$$

which is a special case of Problem (III.2) where $\|\cdot\|_F^2 = \|\cdot\|_{2,2}^2 = \sum_j^p \sum_i^n (\cdot)^2$ is the squared Frobenius norm that serves in the loss $\ell(\cdot, \cdot)$ and $\|\cdot\|_{1,1}$ is the sparsity-promoting regularizer $R(\cdot)$. Note that, since $\boldsymbol{\beta} = \left[\boldsymbol{\beta}_{\bullet 1}, \cdots, \boldsymbol{\beta}_{\bullet p}\right]$ and $\|\cdot\|_{1,1}$ acts independently on each entry of $\boldsymbol{\beta}$, solving Problem (III.3) reduces to p independent LASSO [Tishbirani 1996] sub-problems where each one is a learning task that consists to estimate, for $k = 1 \ldots p$, $\boldsymbol{\beta}_{\bullet k}$ from $\left(\dot{\mathbf{x}}_{\bullet k}, \mathbf{x}_{\bullet k}\right)$ with the ℓ_1 norm weighted by λ. In this way, learning the SODE is performed in a single-task learning manner.

III.3 Learning algorithm

[Schaeffer 2017] formalized the discovery of nonlinear dynamics in the case of partial differential equations and thus the dictionary used in his algorithm is different to our $\boldsymbol{\Theta}_{X_n}$ as it involves partial derivatives. It turns out that the learning criterion that the author used was formalized similarly to Problem (III.3) for SODE. Since Problem (III.3) is convex in $\boldsymbol{\beta}$, Schaeffer solved it with the Douglas-Rachford algorithm [Combettes & Pesquet 2011] (Algorithm III.1) which is a proximal algorithm [Parikh & Boyd 2013]. One can see that the main steps of Algorithm III.1 are in lines 4 and 5. These steps consist to iterate between the shrinkage of $\boldsymbol{W}$ (sparsity promotion) followed by the regularized inversion of $\boldsymbol{Z}$ (loss minimization). Step 6 maintains sparsity in $\boldsymbol{W}$. The algorithm can be applied in low as well as in high-dimension settings, $m \gg n$, *i.e.* when there are more candidate functions than samples in $\boldsymbol{\Theta}_{X_n}$. The notion of proximal operator is at the core of this learning algorithm as well as in our proposal for MTL, so we recall its formal definition.

Algorithm III.1: Douglas-Rachford algorithm for Problem (III.3)

Input: data samples $\dot{\boldsymbol{X}}_n, \boldsymbol{X}_n$, sparsity amount $\lambda > 0$, initial guess $\boldsymbol{\beta}_0$, $0 < \mu < 2$

1: build $\boldsymbol{\Theta}_{X_n}$ arbitrarily from $\boldsymbol{X}_n$
2: $\boldsymbol{W} \leftarrow \boldsymbol{\beta}_0$, $\boldsymbol{\beta} \leftarrow \boldsymbol{W}$
3: **while** $\boldsymbol{\beta}$ has not converged **do**
4: $\quad \boldsymbol{Z} \leftarrow 2\text{prox}_{\lambda\|\cdot\|_{1,1}}(\boldsymbol{W}) - \boldsymbol{W}$
5: $\quad \boldsymbol{W} \leftarrow \mu(I + \lambda\boldsymbol{\Theta}_{X_n}^\top \boldsymbol{\Theta}_{X_n})^{-1}(\boldsymbol{Z} - \lambda\boldsymbol{\Theta}_{X_n}^\top \dot{\boldsymbol{X}}_n) - \boldsymbol{Z}$ $+(1 - \frac{\mu}{2})\boldsymbol{W}$
6: $\quad \boldsymbol{\beta} \leftarrow \text{prox}_{\lambda\|\cdot\|_{1,1}}(\boldsymbol{W})$
7: **end while**

Output: $\boldsymbol{\beta}$

Definition III.1. *(proximal operator [Parikh & Boyd 2013]) The proximal operator associated to a closed proper convex function of a Hilbert space,*

$h : \mathcal{H} \to \mathbb{R}$, is defined for every $\mathbf{y} \in \mathcal{H}$, with $\lambda > 0$ as:

$$\text{prox}_{\lambda h}(\mathbf{y}) = \arg\min_{\mathbf{u} \in \mathcal{H}} \frac{1}{2}\|\mathbf{y} - \mathbf{u}\|_{\mathcal{H}}^2 + \lambda h(\mathbf{u})$$

Remark III.3.0.1. *Here, depending on the context, $\mathcal{H}$ reduces either to $\mathbb{R}^p$ or $\mathbb{R}^{n \times p}$. By the strong convexity of the two terms in the optimization problem involved in Definition III.1, the optimizer $\text{prox}_{\lambda h}(\mathbf{u})$ is unique and thus is single-valued in $\mathcal{H}$ [Parikh & Boyd 2013].*

Remark III.3.0.2. *When h is separable, i.e. for any vector or matrix $\boldsymbol{W}$, $h(\boldsymbol{W}) = \sum_i \sum_j h_{ij}(w_{ij})$ with $h_{ij} : \mathbb{R} \to \mathbb{R}$, computing $\text{prox}_{\lambda h}(\boldsymbol{W})$ reduces to compute the proximal operator of h_{ij} for every i, j and then to concatenate $\left\{\text{prox}_{\lambda h_{ij}}(w_{ij})\right\}_{ij}$ according to the dimensions of $\boldsymbol{W}$. In other words, in the case of a separable vector or matrix function, evaluating the proximal operator on a given element boils down to evaluate the proximal operator for each of the separable parts. Hence, since the $\ell_{1,1}$ matrix norm is separable, $h_{ij}(w_{ij}) = |w_{ij}|$, its proximal operator acts entrywisely and $\text{prox}_{\lambda|\cdot|}(w_{ij}) = \text{sign}(w_{ij}) \max(0, |w_{ij}| - \lambda)$ which is known as the soft-thresholding operator [Tishbirani 1996].*

Roughly speaking, in the learning algorithm of a linear model, the proximal operator of a convex non-differentiable function (*e.g.* $\|\cdot\|_{1,1}$) serves as a shrinkage operator which assigns zero to coefficients in $\boldsymbol{\beta}$ that do not decrease the learning criterion enough. Actually, the exact shrinking to zero, of the coefficients associated to irrelevant candidate functions for each task, is due to the non-differentiability of the regularizer. Note that $\|\cdot\|_{1,1}$ is used as a sparsity-promoting regularizer in Problem (III.3), is separable and thus does not consider any matrix structure. Thus, for MTL, the separability of regularizer is not desirable. Indeed, one can permute any element within the coefficient matrix $\boldsymbol{\beta}$ to learn, the resulting $\ell_{1,1}$ norm remains unchanged. Consequently, the regularizer acts as if $\boldsymbol{\beta}$ were a vector in $\mathbb{R}^{mp}$.

III.3.1 Shortcomings

Considering a simple vector structure on $\boldsymbol{\beta}$ through a fully separable norm regularizer, rather than a matrix-structured one, amounts to omit task relatedness *i.e.* correlations between columns $\left[\boldsymbol{\beta}_{\bullet 1}, \cdots, \boldsymbol{\beta}_{\bullet p}\right]$. Furthermore, the convexity of a norm in $\mathbb{R}^{m \times p}$ induces a bias in the learned coefficients [Fan & Li 2001, Boyd & Vandenberghe 2004] and can degrade the identification of the SODE. Indeed, the soft-thresholding operator evaluated on a regression coefficient, *i.e.* $\text{prox}_{\lambda|\cdot|}(\beta_{ik}) = \tilde{\beta}_{ik} = \text{sign}(\beta_{ik})(|\beta_{ik}| - \lambda)$ if $|\beta_{ik}| - \lambda > 0$. Hence, if the i-th candidate function is relevant for the k-th equation of the SODE and if the sparsity amount λ must be large, the learned coefficient $\tilde{\beta}_{ik}$ is underestimated (in absolute value) with a bias amount of $\lambda \operatorname{sign} \beta_{ik}$.

III.3.2 Building-block of MTL of linear regressions

Since we address the discovery of a SODE with multiple linear regressions, herbelow we describe the building block of MTL in this framework. The core idea of MTL consists to regularize all the tasks jointly during the learning. Here a task refers to the discovery of a single equation of the SODE. We discuss stat of the art sparsity-promoting regularizers that can deal with sparsity across tasks.

MTL consists in learning p functions $\left[f_1, \cdots, f_p\right]$ jointly by assuming that they are close to each other in some similarity metric [Argyriou *et al.* 2008] and share a common set of features. For the k-th task, one is given a data set $\{\mathbf{y}_{ik}; z_{ik}\}_{i \leq n_k}$ of n_k samples with m features. Therefore, the p regression coefficient vectors can be represented in a matrix $\boldsymbol{\beta} = \left[\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_p\right] \in \mathbb{R}^{m \times p}$. The task similarity is reflected within the learning criterion by a regularizer applied on this matrix. For p linear regressions, MTL can be formulated as computing $\boldsymbol{\beta}^*$ as:

$$\boldsymbol{\beta}^* = \arg \min_{\boldsymbol{\beta} = [\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_p]} \sum_{k=1}^{p} \sum_{i=1}^{n_k} \frac{1}{2n_k} (z_{ik}; \mathbf{y}_{ik}^\top \boldsymbol{\beta}_k)^2 + \lambda R(\boldsymbol{\beta}) \qquad \text{(III.4)}$$

where R is the regularizer that may take into account for task relatedness and $\lambda > 0$ is the regularization amount. Note that when $n_k = n$ and R is $\|\cdot\|_{1,1}$, Problem (III.4) reduces to Problem (III.3). Thus by choosing a regularizer, more appropriate than the $\ell_{1,1}$, to considers relatedness between tasks, the discovery of a SODE can be reformulated as a MTL problem. Hence solving for Problem (III.3) corresponds to learn $f_1, \cdots, f_p$ independently.

III.3.2.1 Considering task relatedness

To account for task relatedness in the learning and not to perform p independent single-task learning, the regularizer has to consider a matrix structure on the coefficients. For instance, solving Problem (III.4) with R chosen as $R_{\ell_{2,1}}$, *i.e.* $\|\boldsymbol{\beta}\|_{2,1} = \sum_i^m \|\boldsymbol{\beta}_{i\bullet}\|_2$ matrix norm (the so-called *group-lasso* [Yuan & Lin 2006]) makes $\boldsymbol{\beta}$ row-sparse *i.e.* some of its rows are nonzero and all the others exactly equal zero. Such a regularizer enforces all the components of the SODE to have the same nonzero coefficients and thus to share the same candidate functions specified in $\boldsymbol{\Theta}_{X_n}$ [Obozinski *et al.* 2010].

III.3.2.2 Considering task specific elements

Rather than only considering specific candidate functions with the entry-wise $\|\cdot\|_{1,1}$ norm, or oppositely, only common candidate functions across the components of the SODE with the $\|\cdot\|_{2,1}$ norm, it is more realistic to consider both. Such a compromise can be achieved by taking R as a convex combination of these two norms *i.e.* $R_{\ell_{2,1}+\ell_{1,1}} = \alpha\|\cdot\|_{2,1} + (1-\alpha)\|\cdot\|_{1,1}$ with $\alpha \in [0, 1]$. For single-task linear regression, it is formulated as an extension of the group-lasso when groups of features are known and some of the features do not act at the group level [Simon *et al.* 2013]. When $\alpha > 0.5$, the learning criterion attaches more importance on common candidate functions across tasks than task specific ones, and conversely when $\alpha < 0.5$. In practice, α is set around 0.9 to allow few task specific components but it can be fine-tuned by cross-validation.

III.3.2.3 Shortcomings

Despite being able to select relevant candidate functions, the norms $R_{\ell_{2,1}}$ and $R_{\ell_{2,1}+\ell_{1,1}}$ induce a bias in the learned coefficients within $\boldsymbol{\beta}$. Similarly to $\|\cdot\|_{1,1}$, this bias is induced by the convexity of such regularizers through their associated proximal operator which are available analytically (no optimization step is needed to evaluate them) and given in [Simon *et al.* 2013]. We consider these regularizers as baselines in our numerical experiments in Section III.5 and show that the nonconvexity is important to get an accurate estimate of the SODE.

In Section III.2.1 and Section III.3.2, task relatedness and nonconvexity are showed to be two important weaknesses not addressed by the $\ell_{1,1}$ regularizer used in state-of-the-art algorithms. Our contribution leverages both the nonconvexity and task relatedness within a single regularizer to improve the discovery of a SODE.

III.4 Nonconvex matrix-structured regularizer

In this section, we expose our contribution and propose a nonconvex regularizer which considers both relatedness between the tasks, through sparsity, and unbiasedness. To learn a SODE with such a regularizer, we instantiate a generative iterative thresholding algorithm [Gong *et al.* 2013] which can be used with convex as well with nonconvex regularizers which can be expressed as the difference of two convex functions.

III.4.1 Nonconvex separable regularizer

We first introduce the single task learning version of the nonconvex regularizer that we extend to MTL by applying it entrywisely on a matrix, *i.e.* as the ℓ_1 norm to $\ell_{1,1}$ norm. This nonconvex regularizer, the Smoothly Clipped Absolute Deviation (SCAD) [Fan & Li 2001], does not account for task relatedness but serves as a building block for the regularizer we propose and incorporates unbiasedness.

Definition III.2. *Let $\lambda > 0$ and $\theta > 2$ be two hyperparameters that serve as the sparsity amount and unbiasedness level respectively. The SCAD penalty is defined for any $w \in \mathbb{R}$ as:*

$$r_{\lambda,\theta}^{SCAD}(w) = \begin{cases} \lambda|w| & \text{if } |w| \leq \lambda \\ -\frac{\lambda^2 - 2\theta\lambda|w| + w^2}{2(\theta-1)} & \text{if } \lambda < |w| \leq \theta\lambda \\ \frac{(\theta+1)\lambda^2}{2} & \text{if } |w| > \theta\lambda \end{cases} \tag{III.5}$$

Remark III.4.1.1. *$r_{\lambda,\theta}^{SCAD}$ is not differentiable for $x = 0$ and nonconvex (actually concave in $\mathbb{R}^+$). Like any convex regularizer used in a learning algorithm, the non-differentiability at 0 involves sparsity which is a desirable property. Also, when θ increases, the SCAD regularizer approximates the ℓ_1 norm [Fan & Li 2001].*

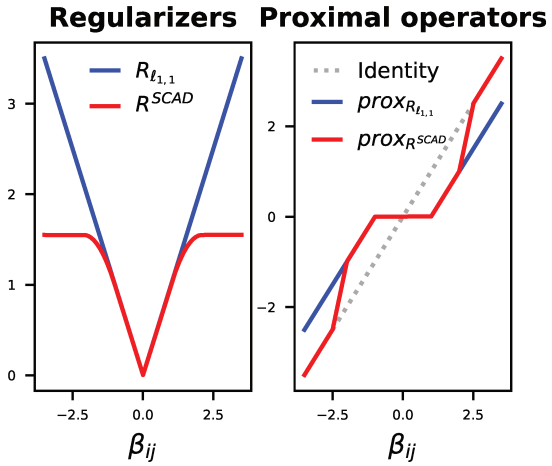

Figure III.2: Illustration of the (convex) $\ell_{1,1}$ and (nonconvex) Smoothly Clipped Absolute Deviation penalties (left) and associated proximal operators (right). $\lambda = 1$ and $\theta = 2.01$. The bias is the difference between $\text{prox}_R(\beta_{ij})$ and β_{ij} (*i.e.* the identity, in grey dots).

We illustrate $r^{SCAD}_{\lambda,\theta}$ in Figure III.2 (left).

Although $r^{SCAD}_{\lambda,\theta}$ is nonconvex, one can compute its proximal operator (which is not necessarily single-valued since $r^{SCAD}_{\lambda,\theta}$ is nonconvex) by solving analytically multiple optimization problems, depending on the value of w in Equation (III.5), involved by Definition III.1. The proximal operator of $r^{SCAD}_{\lambda,\theta}$ is given in [Fan & Li 2001] by:

$$\text{prox}_{r^{SCAD}_{\lambda,\theta}}(w) = \begin{cases} \text{sign}(w)\max(0, |w| - \lambda) & \text{if } |w| \leq 2\lambda \\ \frac{\theta-1}{\theta-2}\text{sign}(w)\max(0, |w| - \frac{\theta}{\theta-1}\lambda) & \text{if } 2\lambda < |w| \leq \theta\lambda \\ w & \text{if } |w| > \theta\lambda \end{cases} \tag{III.6}$$

We illustrate $\text{prox}_{r^{SCAD}_{\lambda,\theta}}(\cdot)$ in Figure III.2 (right), showing that for large entry w, it returns unbiased output since such a proximal operator converges to the identity function.

$R^{SCAD}_{\lambda,\theta}$, the vector or matrix version of $r^{SCAD}_{\lambda,\theta}$ can be easily built by applying it entrywisely and summing the resulting outputs along the input dimen-

sion(s) *i.e.* for any vector or matrix $\boldsymbol{W}$, $R^{SCAD}_{\lambda,\theta}(\boldsymbol{W}) = \sum_i \sum_j r^{SCAD}_{\lambda,\theta}(w_{ij})$. One can note that such an extension results, by construction, in a regularizer having the separability property. Thus, the proximal operator of $R^{SCAD}_{\lambda,\theta}$ is obtained by evaluating $\text{prox}_{r^{SCAD}_{\lambda,\theta}}$ entrywisely. However, since a separable regularizer cannot take into account relatedness between tasks, like the $\ell_{1,1}$ norm, the benefit gained by learning with $R^{SCAD}_{\lambda,\theta}$ over the latter is only unbiasedness. Indeed, the first summand in $R^{SCAD}_{\lambda,\theta}$ acts both at the row and column levels of $\boldsymbol{\beta}$. In other words, it acts for the coefficient β_{ij} associated to i-th candidate function (i-th column of $\boldsymbol{\Theta}_{X_n}$) of the j-th task, and therefore independently across the p equations of the SODE. We propose to "un-separate" $R^{SCAD}_{\lambda,\theta}$ to both leverage from non-separability as well as nonconvexity of the SCAD penalty.

III.4.2 Nonconvex non-separable regularizer

We propose to learn $\boldsymbol{\beta} = \left[\boldsymbol{\beta}_{1\bullet}, \cdots, \boldsymbol{\beta}_{m\bullet}\right]^\top$ in $\dot{\boldsymbol{X}}_n = \boldsymbol{\Theta}_{X_n}\boldsymbol{\beta}$, by solving Problem III.4 with our proposed regularizer $R^{SCAD-\ell_1}$. To make the proposed regularizer non-separable, such that it accounts for task relatedness as well as unbiasedness, the key point is to replace the second summation of $R^{SCAD}_{\lambda,\theta}$ by the evaluation of $r^{SCAD}_{\lambda,\theta}$ of the ℓ_1 norm of $\boldsymbol{\beta}_{i\bullet}$ for every i:

$$R^{SCAD-\ell_1}_{\lambda,\theta}(\boldsymbol{\beta}) = \sum_{i=1}^{m} r^{SCAD}_{\lambda,\theta}\Big(\|\boldsymbol{\beta}_{i\bullet}\|_1\Big) \tag{III.7}$$

In this manner, since the regularizer acts for the i-th candidate function onto the coefficient vector $\boldsymbol{\beta}_{i\bullet} \in \mathbb{R}^p$ through the SCAD penalty of the ℓ_1 norm, $R^{SCAD-\ell_1}$ enforces the coefficients of the i-th candidate function to be sparse, unbiased and correlated across the p tasks. Note the analytical similarity with the $R_{\ell_{2,1}} = \sum_i \|\boldsymbol{\beta}_{i\bullet}\|_2$ norm (Section III.3.2) which does not allow each equation of the SODE to have a specific candidate function. Contrary to $R_{\ell_{2,1}}$, our proposal $R^{SCAD-\ell_1}$ enforces each $\boldsymbol{\beta}_{i\bullet}$ to have a small ℓ_1 norm and thereby enables the components of the SODE to have specific candidate functions (this is due to the sparsity induced by the ℓ_1 norm). $R^{SCAD-\ell 1}$

was reformulated as a sparse group penalty to learn single-task linear models when groups of correlated predictive variables are known but some of the groups may have been misspecified [Jiang & Huang 2015]. To the best of our knowledge, this regularizer has never been extended to MTL.

Like $R^{SCAD}_{\lambda,\theta}$, the regularizer $R^{SCAD-\ell_1}_{\lambda,\theta}$ is nonconvex but its proximal operator can be computed according the ℓ_1 norm of the rows of $\boldsymbol{W} = \left[\mathbf{w}_{1\bullet}, \cdots, \mathbf{w}_{m\bullet}\right]^\top$ as, for $i \in \{1, \ldots, m\}$:

$$\operatorname{prox}_{R^{SCAD-\ell_1}_{\lambda,\theta}}(\boldsymbol{W}) = \begin{cases} \operatorname{sign}(\mathbf{w}_{i\bullet}) \max(0, |\mathbf{w}_{i\bullet}| - \lambda) & \text{if } |\mathbf{w}_{i\bullet}| \leq 2\lambda - \|\mathbf{w}_{i\bullet}\|_1 \\ \frac{\theta-1}{\theta-2} \operatorname{sign}(\mathbf{w}_{i\bullet}) \max(0, |\mathbf{w}_{i\bullet}| - \frac{\theta}{\theta-1}\lambda) & \text{if } 2\lambda - \|\mathbf{w}_{i\bullet}\|_1 < |\mathbf{w}_{i\bullet}| \leq \theta\lambda - \|\mathbf{w}_{i\bullet}\|_1 \\ \mathbf{w}_{i\bullet} & \text{if } |\mathbf{w}_{i\bullet}| > \theta\lambda - \|\mathbf{w}_{i\bullet}\|_1 \end{cases} \tag{III.8}$$

where sign, max and $|\cdot|$ act entrywisely. We will see in the next section that knowing, analytically, this proximal operator is essential to perform the learning.

III.4.3 MTL with a nonconvex regularizer

In this section, we instantiate the Generative Iterative Sequential Thresholding (GIST) algorithm of [Gong *et al.* 2013] to learn $\boldsymbol{\beta} \in \mathbb{R}^{m \times p}$ with nonconvex regularizers that can be expressed as a difference of two convex functions, see [Gasso *et al.* 2009] for examples of such regularizers. We present the GIST algorithm in Algorithm III.2 with a regularizer R that can be either convex, either nonconvex and can be written a difference of two convex functions, see [Gong *et al.* 2013] for convergence guarantee of GIST. This is a generalization of the Fast Iterative Shrinkage-Thresholding algorithm [Beck & Teboulle 2009] designed to minimize (convex) least-squares plus ℓ_1 norm problems. It turns out that $R^{SCAD-\ell_1}$, and R^{SCAD} as well, have this property, thereby the GIST algorithm encompasses Problem III.4. We also used GIST with R convex for baseline comparisons in numerical experiments in Section III.5.

GIST algorithm (Algorithm III.2) consists in two nested loops. The outer

Algorithm III.2: GIST for Problem (III.4) with R possibly non-convex

Input: data samples $\dot{\boldsymbol{X}}_n$ and $\boldsymbol{X}_n$, sparsity amount $\lambda > 0$, unbiasedness level $\theta > 2$, initial guess $\boldsymbol{\beta}_0$, step size γ, step size minimum γ_{min}.

1: build $\boldsymbol{\Theta}_{X_n}$ arbitrarily from $\boldsymbol{X}_n$
2: $\boldsymbol{\beta} \leftarrow \boldsymbol{\beta}_0$, $\gamma \leftarrow \gamma_{min}$
3: $\boldsymbol{W} \leftarrow \boldsymbol{\beta} - \gamma_{min}\boldsymbol{\Theta}_{X_n}^{\top}(\boldsymbol{\Theta}_{X_n}\boldsymbol{\beta} - \dot{\boldsymbol{X}}_n)$
4: **while** $\boldsymbol{\beta}$ has not converged **do**
5: **while** line search criterion unsatisfied **do**
6: $\boldsymbol{\beta} \leftarrow \text{prox}_{\gamma\lambda R}(\boldsymbol{\beta})$
7: $\gamma \leftarrow 0.8\gamma$
8: **end while**
9: $\boldsymbol{W} \leftarrow \boldsymbol{\beta} - \gamma\boldsymbol{\Theta}_{X_n}^{\top}(\boldsymbol{\Theta}_{X_n}\boldsymbol{\beta} - \dot{\boldsymbol{X}}_n)$
10: $\boldsymbol{\beta} \leftarrow \text{prox}_{\gamma\lambda R}(\boldsymbol{W})$
11: $\gamma \leftarrow \gamma_{min}$
12: **end while**

Output: $\boldsymbol{\beta}$

loop (lines 4-12) consists in performing a loss gradient descent step (line 9) followed by a shrinkage operator step (which here learns the coefficients of the underlying SODE). In practice, the outer loop (line 4) is stopped if the $\ell_{2,2}$ norm of the relative change of $\boldsymbol{\beta}$ between two consecutive iterations is less than a low value like 10^{-5} or the total number of iterations is greater than 5000. The inner loop (lines 5-8) consists to perform a line search to compute a gradient step size γ so that it ensures a sufficient decrease of the loss along the negative gradient direction. Here we used the backtracking line search criterion with parameter value 0.8 (line 7), which is commonly used, and serves as the 'slow-rate' of the line search (must lie in $]0, 1[$) [Boyd & Vandenberghe 2004]. Note that learning with $R_{\ell_{1,1}}$ in Algorithm III.2 approximately reduces to learning with Algorithm III.1. In the next section we instantiate Algorithm III.2 with different regularizers and compare their ability to recover a SODE from noisy data.

III.5 Numerical experiments

III.5.1 Experimental setting

III.5.1.1 Synthetic SODEs

To show the efficiency of our approach, we evaluated it on three reference two-dimensional SODEs: the Damped Oscillator with Cubic dynamic (DOC) (used to model damped behaviors with non-linearity), the Lotka-Volterra (LV) system (used for predator-prey interactions modeling) and the Lorenz Attractor (LAT) system (used for excitation systems modeling like neurons). Each one of these systems has common functions as well as specific functions across their two equations. We used the same settings as [Schaeffer & McCalla 2017]. We generated the time series by solving numerically the true SODEs with the explicit Runge-Kutta-45 method. We corrupted the time series and their time derivative with a Gaussian noise ($\sigma = 0.1$). Based on these noisy samples, we built the dictionary with monomial candidate functions (up to degree five, with first and second order interactions *e.g.* $x_1x_2, x_1^2x_2$).

For clarity, we give the analytic form of the DOC, LV and LAT SODE:

DOC

$$\begin{cases} \dot{x}_1(t) = -0.1x_1^3(t) + 2x_2^3(t) \\ \dot{x}_2(t) = -2x_2^3(t) - 0.1x_1^3(t) \end{cases} \tag{III.9}$$

with $\mathbf{x}_{DOC}(0) = \Big[0, 2\Big]$, $T_{DOC} = 25$ and $n_{DOC} = 5.10^3$. The numerical solution of the DOC SODE is plotted in Fig III.3.

LV

$$\begin{cases} \dot{x}_1(t) = 1.5x_1(t) - x_1(t)x_2(t) \\ \dot{x}_2(t) = -3x_2(t) - x_1(t)x_2(t) \end{cases} \tag{III.10}$$

with $\mathbf{x}_{LV}(0) = \Big[0, 2\Big]$, $T_{LV} = 4.5$ and $n_{LV} = 5.10^3$. The numerical solution of

the LV SODE is plotted in Fig III.4.

LAT

$$\begin{cases} \dot{x}_1(t) = -10x_1(t) + 10x_2(t) \\ \dot{x}_2(t) = 28x_1(t) - x_2(t) - x_1(t)x_3(t) \\ \dot{x}_3(t) = -\frac{8}{3}x_3(t) + x_1(t)x_2(t) \end{cases} \tag{III.11}$$

with $\mathbf{x}_{LAT}(0) = \Big[-5, 1, 20\Big]$, $T_{LAT} = 10$ and $n_{LAT} = 5.10^3$ as in [Schaeffer & McCalla 2017]. The numerical solution of the LAT SODE is plotted in Fig III.5.

III.5.1.2 Airbus flight test data

We experimented the baselines and our method on a multivariate time series provided by an Airbus flight test expert. For data confidentiality, we cannot share details on the data meaning. The dataset includes the time series of the sampled state-variables (Figure III.7, blue curves), as well as their sampled time derivatives (Figure III.8 blue curves). We built the dictionary with monomial candidate functions (up to degree five, with first, second and third order interactions terms *e.g.* x_1x_2, $x_1^2x_2$, $x_1^3x_2$). Based on the knowledge of a flight test expert, we also included $\cos(2\pi x_1)$, $\cos(2\pi 0.5x_1)$, $\sin(2\pi x_1)$ and $\sin(2\pi 0.5x_1)$. Also, to account for temporal variability of the SODE, regardless of the state-variables, we included exponential terms $e^{-0.5t}, e^{-t}, e^{-1.5t}, e^{-2t}$ in the dictionary. Since it is a real dataset, the true SODE is unknown thus we can only report the ϵ_T error between the sampled state-variables and the numerical solution of the learnt SODE. For data confidentiality, we cannot report the analytical SODE found but a plot of the numerical solution. To solve the SODE, we used the explicit Runge-Kutta-45 method implemented in the SciPy library [Oliphant *et al.* 2001].

III.5.1.3 Implementation

We learnt the SODEs with GISTA with $R^{SCAD-\ell_1}$ and R^{SCAD}. We set $\theta = 2.01$, near its lower bound. The best value of λ was computed as the one

minimizing the Bayesian information criterion (no need for train/test splits with cross-validation)[1] in the logarithmic grid $\{\lambda_0 = 10^3, \ldots, \lambda_{N-1} = 10^{-2}\}$ containing $N = 10^3$ values. As this requires one learning for each λ, we performed warm-start to estimate the models along λ [Friedman *et al.* 2010]. Warm-start consists to first estimate $\hat{\boldsymbol{\beta}}_{\lambda_0}$ with the largest sparsity amount λ_0, such that $\hat{\boldsymbol{\beta}}_{\lambda_0} = \mathbf{0}$, and then to (sequentially) estimate $\hat{\boldsymbol{\beta}}_{\lambda_i}$ by initialising GISTA with $\boldsymbol{\beta}_0 = \hat{\boldsymbol{\beta}}_{\lambda_{i-1}}$.

III.5.2 Comparison with baseline regularizers

We compared our approach by learning with: $R_{\ell_{1,1}}$, $R_{\ell_{2,1}}$, $R_{\ell_{2,1}+\ell_{1,1}}$ ($\alpha = 0.9$), for which the proximal operators are available analytically [Chierchia *et al.* 16]. For $R_{\ell_{1,1}}$ and $R_{\ell_{2,1}}$, we used Algorithm III.1($\mu = 1.5$) and the multi-task-LASSO (MTLa) from Scikit-Learn, respectively. We used the relative squared norm $\epsilon_\beta = \frac{\|\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}^*\|_{2,2}^2}{\|\boldsymbol{\beta}^*\|_{2,2}^2}$, $\epsilon_T = \sum_t^T \frac{\|\hat{\mathbf{x}}(t)-\mathbf{x}(t)^*\|_2^2}{\|\mathbf{x}(t)^*\|_2^2}$ and the rate of misidentified candidate functions ϵ_{MIS} *i.e.* the number of misidentified functions divided by the total number of candidates functions for every tasks $p \times m$. ϵ_β measures the unbiasedness w.r.t the true coefficient matrix $\boldsymbol{\beta}^*$. ϵ_T measures the relative total error, along t, between the (numerical) solution of the learnt SODE w.r.t the true one. The lower ϵ_β, ϵ_T and ϵ_{MIS}, the better the recovery of the SODE.

III.5.3 Results

III.5.3.1 Synthetic SODEs

For each SODE, we repeated the experiment ten times. We report the average and standard deviation of ϵ_β, ϵ_T and ϵ_{MIS} in Table III.1. We show the simulated SODEs, with their closed form, from their best estimate over the ten trials in Fig III.6. The results show that on the DOC, LV and LAT datasets, learning with $R^{SCAD-\ell_1}$ outperforms the convex baselines (Table III.1 $R_{\ell_{1,1}}$, $R_{\ell_{2,1}}$, $R_{\ell_{2,1}+\ell_{1,1}}$ rows) both in terms of SODE identification (smallest ϵ_{MIS}) as

[1] However, without training set, by abuse of language we keep on using "learn" that refers to compute an estimate of $\hat{\boldsymbol{\beta}}$

Table III.1: Results (average (%) ± standard deviation (%), over ten trials) of the learning of the DOC, LV and LAT SODEs with five regularizers.

SODE	DOC ($p=2$)			LV ($p=2$)			LAT ($p=3$)		
Reg/Error	ϵ_β	ϵ_{MIS}	ϵ_T	ϵ_β	ϵ_{MIS}	ϵ_T	ϵ_β	ϵ_{MIS}	ϵ_T
$R_{\ell_{1,1}}(LASSO)$	$10^{-3} \pm 10^{-3}$	15 ± 8.7	14 ± 10^{-3}	$\mathbf{6.4 \pm 2.10^{-1}}$	$\mathbf{38 \pm 4.1}$	$\mathbf{1.2} \pm 10^{-3}$	$2.10^{-3} \pm 3.10^{-4}$	4.1 ± 2.9	37 ± 10^{-3}
$R_{\ell_{2,1}}$ (MTLa)	$10^{-3} \pm 10^{-4}$	13 ± 10	$7.10^{-1} \pm 10^{-3}$	11 ± 1.0	65 ± 13	$10^{162} \pm 10^{-3}$	$1.10^{-3} \pm 2.10^{-4}$	27 ± 6.1	44 ± 10^{-3}
$R_{\ell_{2,1}+\ell_{1,1}}$	$10^{-3} \pm 10^{-4}$	12 ± 9.6	1.3 ± 10^{-3}	10 ± 1.0	65 ± 13	18 ± 10^{-3}	$1.10^{-3} \pm 2.10^{-4}$	19 ± 5.8	44 ± 10^{-3}
R^{SCAD}	$\mathbf{10^{-4} \pm 10^{-4}}$	$\mathbf{6.4 \pm 3.1}$	$\mathbf{3.10^{-1} \pm 10^{-3}}$	14 ± 6.10^{-1}	50 ± 10	38 ± 10^{-3}	$\mathbf{4.10^{-5} \pm 3.10^{-5}}$	$\mathbf{1.2 \pm 0.0}$	$\mathbf{32 \pm 10^{-3}}$
$R^{SCAD-\ell_1}$ (our)	$\mathbf{10^{-4} \pm 10^{-4}}$	$\mathbf{3.5 \pm 3.5}$	$\mathbf{8.10^{-2} \pm 10^{-3}}$	$\mathbf{2.10^{-3} \pm 1.10^{-3}}$	$\mathbf{7.8 \pm 2.1}$	$\mathbf{1.10^{-4} \pm 10^{-3}}$	$\mathbf{4.10^{-5} \pm 3.10^{-5}}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{13 \pm 10^{-3}}$

ϵ_β measures unbiasedness. ϵ_{MIS} is the misidentification error of the SODE. ϵ_T is the error between the solution of a learnt SODE and the ground truth. For each SODE, the two smallest errors are in bold.

well as unbiasedness (smallest ϵ_β). Fig III.6 (LAT simulations) shows that even if the SODE is well identified (small ϵ_{MIS}) with a convex regularizer, the bias of the coefficients leads to a degradation of the forecasting performances of the SODE. Moreover, despite the *a priori* inability of R^{SCAD} to consider the coupling within an SODE, the results nevertheless (Table III.1 penultimate row) show that it is good for variable selection (except for LV whose dictionary $\mathbf{\Theta}_{X_n}$ involves more correlations than for DOC and LAT). Such a result can be attributed to the (semi) concavity of R^{SCAD} that leads to a penalty with stronger sparsity (*i.e.* nearer to the ℓ_0 oracle) than convex regularizers. As a global result, our experiments show that leveraging both from MTL and nonconvexity improves the discovery of the SODE in closed form.

III.5.3.2 Airbus flight test data

Table III.2: Results (%) of the SODE discovery experiment on an Airbus flight test dataset with five regularizers. The two smallest errors are in bold.

SODE	Flight test dataset ($p = 2$)
Reg/Error	ϵ_T
$R_{\ell_{1,1}}(LASSO)$	67.41
$R_{\ell_{2,1}}$ (MTLa)	55.87
$R_{\ell_{2,1}+\ell_{1,1}}$	**55.29**
R^{SCAD}	62.56
$R^{SCAD-\ell_1}$ (our)	**44.96**

We report the error ϵ_T, for each SODE discovered by the five regularizers, in Table III.2. The results show that learning with $R_{SCAD-\ell_1}$ outperforms the baselines. We note that the second best result is achieved by learning with $R_{\ell_{2,1}}$, meaning that considering the coupling within the SODE, regardless of the bias induced by convexity, is a first benefit of our MTL-based approach. Thus, the results are consistent with the ones of the synthetic SODEs, which

shows that learning with a nonconvex MTL-based regularizer improves the discovery.

In addition, in Figure III.7 and Figure III.8), for each learned SODE, we provide the numerical solutions and their time derivative respectively.

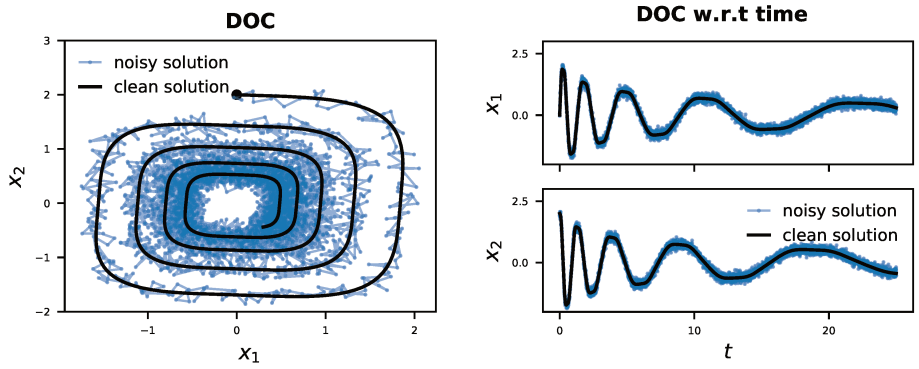

Figure III.3: Left: plot in $\mathbb{R}^2$ of the clean solution (*i.e.* ground truth) (black) of the DOC SODE and the noisy version (blue). The black circle is the initial state $\mathbf{x}_{DOC}(0)$. Right: same data plotted in the temporal domain.

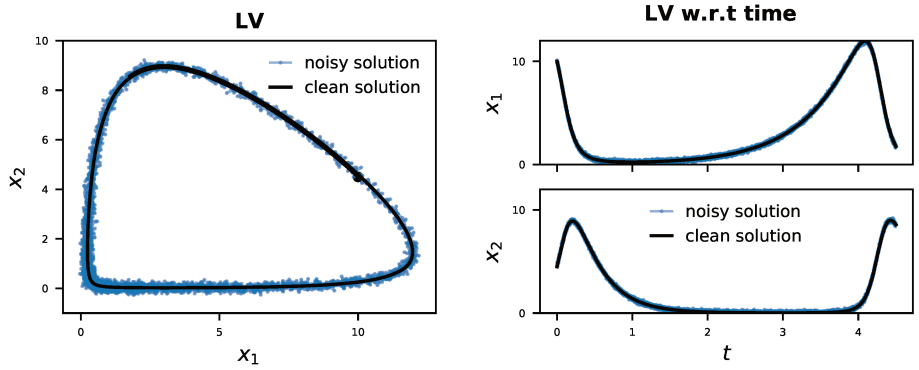

Figure III.4: Left: plot in $\mathbb{R}^2$ of the clean solution (*i.e.* ground truth) (black) of the LV SODE and the noisy version (blue). The black circle is the initial state $\mathbf{x}_{LV}(0)$. Right: same data plotted in the temporal domain.

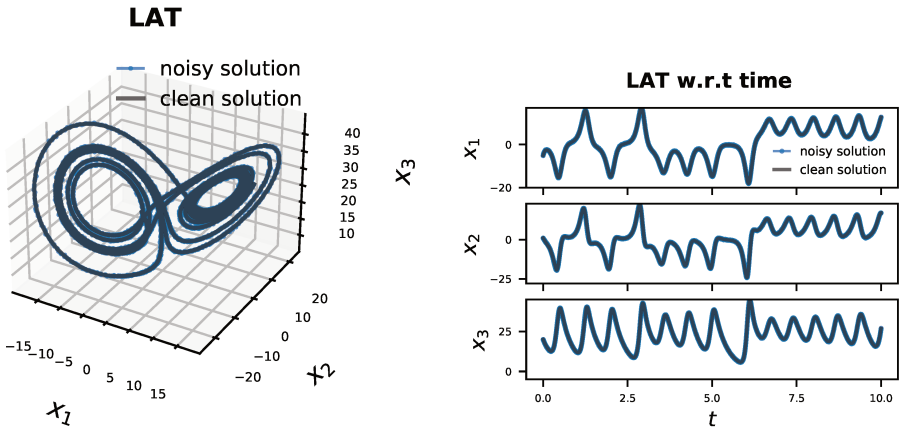

Figure III.5: Left: plot in $\mathbb{R}^3$ of the clean solution (*i.e.* ground truth) (black) of the LAT SODE and the noisy version (blue). The black circle is the initial state $\mathbf{x}_{LAT}(0)$. Right: same data plotted in the temporal domain.

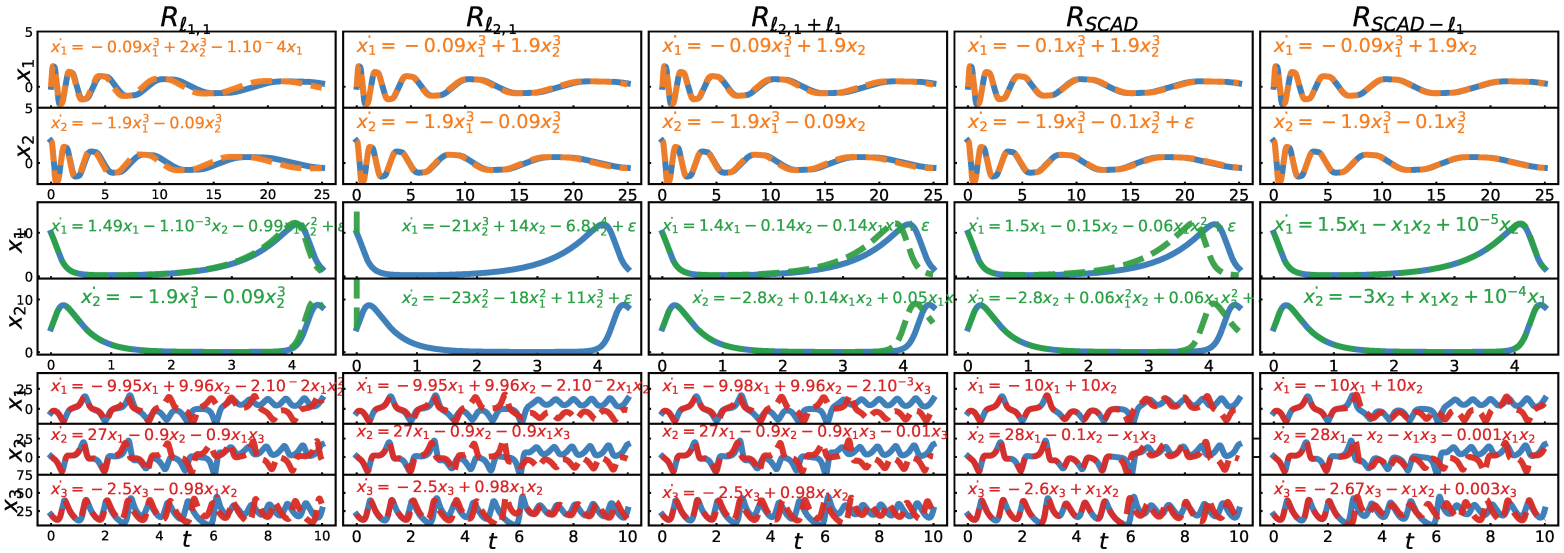

Figure III.6: Time series simulated from $\dot{\mathbf{x}} = f(\mathbf{x})$ where f, displayed analytically in each panel, is the DOC (orange), LV (green), LAT (red) learnt with different regularizers and the ground truth (blue). "$+\epsilon$" refers to misidentified functions with a small coefficient.

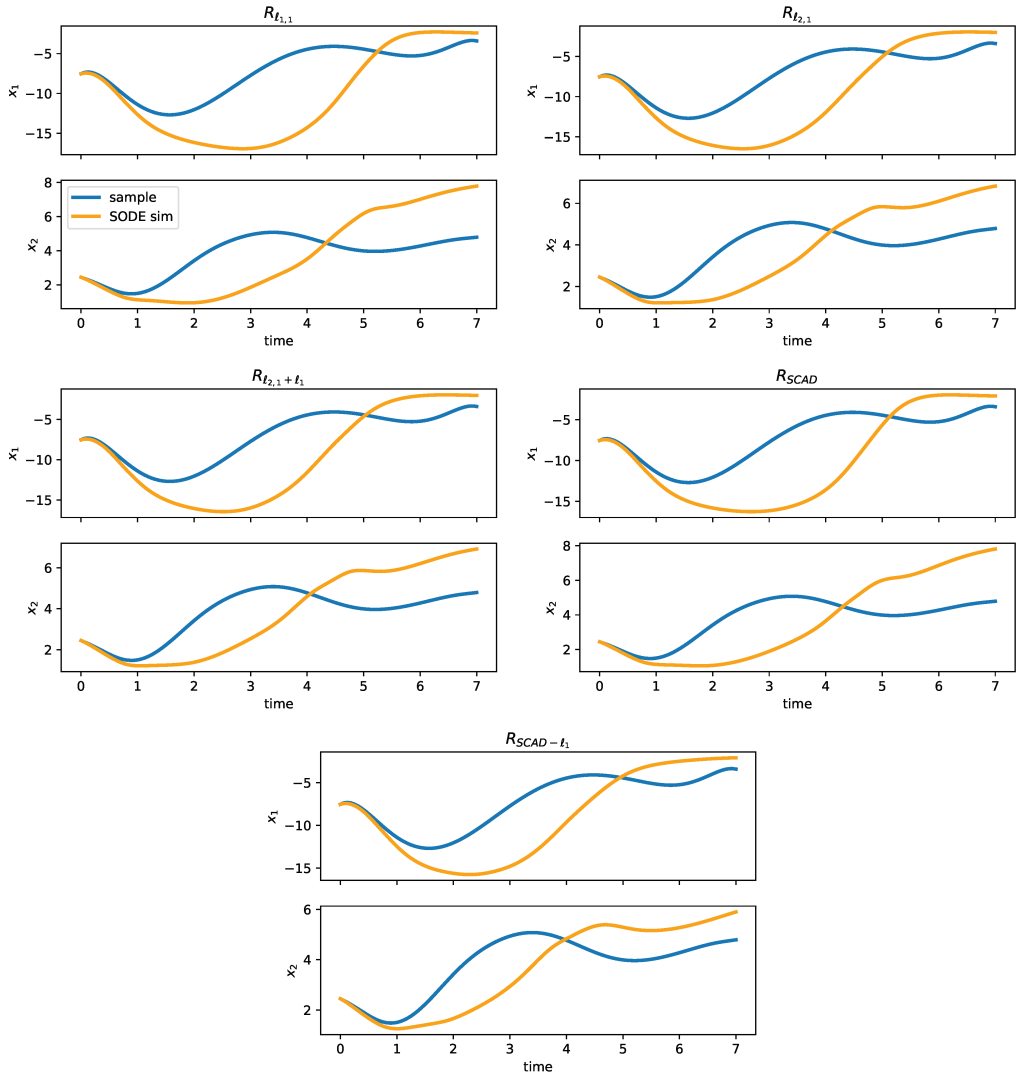

Figure III.7: Sampled state-variables x_1, x_2 (blue) and numerical solution (orange) of the SODEs discovered with the five regularizers.

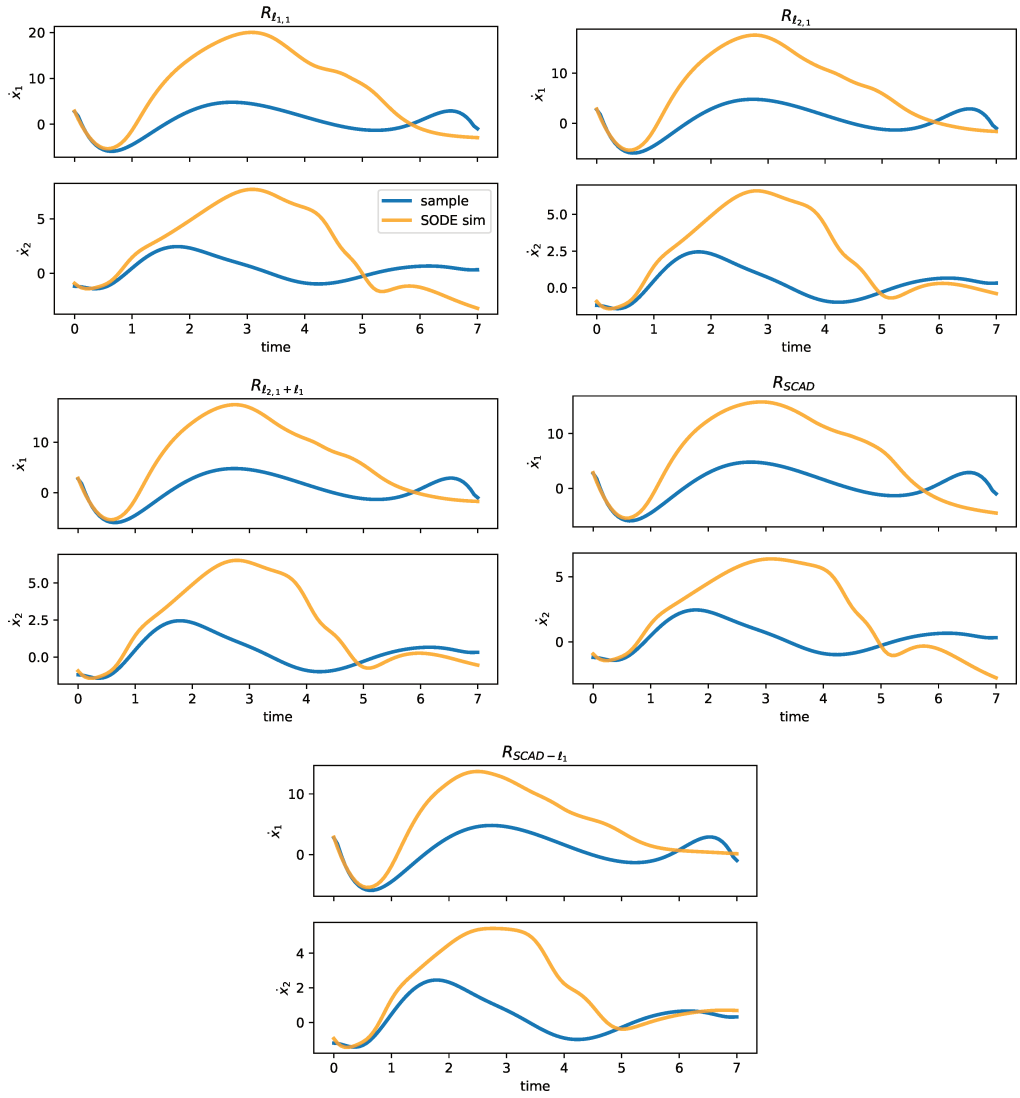

Figure III.8: Sampled state-variable derivatives $\dot{x}_1, \dot{x}_2$ (blue) and derivatives of the numerical solution (orange) of the SODEs discovered with the five regularizers.

III.6 Conclusion

We re-casted the learning of an unknown SODE from noisy data as a MTL problem. We proposed a nonconvex regularizer that better (i) accommodates the coupling of the equations of the underlying SODE than convex regularizers and (ii) results in unbiased coefficients. To learn the SODE with our nonconvex regularizer, we instantiate a generic algorithm from the literature. Numerical experiments on real and synthetic datasets confirm that both the MTL feature and the nonconvexity of our proposal outperform learning with state of the arts regularizers.

General conclusion and perspectives

In this thesis, we tackled two problems in multivariate time series analysis: the detection of outliers and the discovery of SODE. To address both problems, we represented a time series as function over time.

In Chapter II, we used the functional data representation as a pre-processing step. Then, we proposed to extract shape features with the arc-length, velocity and curvature mapping functions which are well established in differential geometry. We used the obtained shape-based data representation as input of an outlier detection algorithm. Through numerical experiments on synthetic and real datasets, we showed the improvement of our approach, in terms of detection and robustness to the contamination level, w.r.t to state of the art.

As future work, a possible improvement is to combine mapping functions so that multiple classes of outliers in the same dataset might be detected. For instance, such a combination can be made with ensemble models, wherein each model has to detect outliers from the representation returned by a single mapping function. Another line of research is the outlier class recognition (in addition to the detection), which would enhance the interpretability of our approach. Recognizing the type of outlyingness can inform the user on the abnormality of the behavior *e.g.* a delay if the outlier is horizontally

shifted w.r.t to the inlier class. This could also be addressed by ensemble models wherein each model is trained on a single outlier class. Then, one can decompose the global outlyingness score in multiple sub-scores, where each one refers to an outlier class. Such a decomposition would inform on the contribution of each outlier class to an outlier.

In Chapter III, we represented the time-dependent function underlying a noisy time series as the solution of an unknown SODE. Accessing to the analytical form of the SODE is of a great interest since it can give an explicit, and functional, understanding on a particular dynamic system. This inverse problem is formulated as a sparse regression problem. However, state of the art algorithms are single-task based and thus omit the coupling feature within the SODE. Furthermore, they involve the convex ℓ_1 sparse penalty that results in biased estimate of the regression weights. To remedy these two limitations, we re-casted the problem with MTL involving a nonconvex penalty. Our numerical experiments showed the improvement of our algorithm w.r.t to state of the art.

A limitation of our approach, also not addressed by state of the art, is its restriction to discover the SODE underlying a *single* multivariate time series. Therefore, for a dataset comprising multiple multivariate time series, this is required to learn each SODE independently. In this way, as the size of the dataset increases, the learning can be costly and does not consider the potential similarities between the time series. A future work, would be to jointly discover multiple SODEs with a constraint of similarity between the regression weights across all the SODEs.

Publications

[Baril *et al.* 2018] Xavier Baril, Oihana Coustie, Clément Lejeune, Josiane Mothe, Adil Soubki et Olivier Teste. *A Study on Time Series Dimensionality Reduction for Data Mining*. In Research Summer School on Statistics for Data Science (S4D 2018), Caen, France, Juin 2018.

[Lejeune *et al.* 2020a] Clément Lejeune, Josiane Mothe, Adil Soubki et Olivier Teste. *Shape-based outlier detection in multivariate functional data*. Knowledge-Based Systems, vol. 198, page 105960, 2020.

[Lejeune *et al.* 2020b] Clément Lejeune, Josiane Mothe et Olivier Teste. *Outlier detection in multivariate functional data based on a geometric aggregation*. In Proceedings of the 23rd International Conference on Extending Database Technology, EDBT 2020, Copenhagen, Denmark, March 30 - April 02, 2020, pages 383–386, 2020.

[Lejeune *et al.* 2021] Clément Lejeune, Josiane Mothe, Adil Soubki et Olivier Teste. Data-driven discovery of systems of ordinary differential equations through nonconvex multi-task learning (under review). 2021.

Bibliography

[Aggarwal & Yu 2001] Charu C Aggarwal et Philip S Yu. *Outlier Detection for High-Dimensional Data*. In SIGMOD, volume 30, pages 37–46. ACM, 2001.

[Argyriou *et al.* 2008] Andreas Argyriou, Theodoros Evgeniou et Massimiliano Pontil. *Convex multi-task feature learning*. Machine Learning, vol. 73, no. 3, pages 243–272, 2008.

[Arribas-Gil & Romo 2014] Ana Arribas-Gil et Juan Romo. *Shape outlier detection and visualization for functional data: The outliergram*. Biostatistics, vol. 15, no. 4, pages 603–619, 2014.

[Baril *et al.* 2020] Xavier Baril, Oihana Coustié, Josiane Mothe et Olivier Teste. *Application Performance Anomaly Detection with LSTM on Temporal Irregularities in Logs*. In Proceedings of the 29th ACM International Conference on Information and Knowledge Management, CIKM '20, page 1961–1964. Association for Computing Machinery, 2020.

[Beck & Teboulle 2009] Amir Beck et Marc Teboulle. *A Fast Iterative Shrinkage-Thresholding Algorithm*. SIAM Journal of Imaging Sciences, vol. 2, no. 1, pages 183–202, 2009.

[Bhat & Rawat 2019] Harish S Bhat et Shagun Rawat. *Learning Stochastic Dynamical Systems via Bridge Sampling*. In European Conference on Machine Learning, 2019.

[Boulfani *et al.* 2020] Feriel Boulfani, Xavier Gendre, Anne Ruiz-Gazen et Martina Salvignol. *Anomaly detection for aircraft electrical generator using machine learning in a functional data framework*. In GMC-

ElecEng2020, Valence, Spain, Septembre 2020.

[Bowman 2012] Frank Bowman. Introduction to bessel functions. Courier Corporation, 2012.

[Boyd & Vandenberghe 2004] Stephen Boyd et Lieven Vandenberghe. Convex Optimization. Cambridge University Press, 2004.

[Brunton *et al.* 2016] Steven L. Brunton, Joshua L. Proctor et J. Nathan Kutz. *Discovering governing equations from data by sparse identification of nonlinear dynamical systems.* Proceedings of the National Academy of Sciences, vol. 113, no. 15, pages 3932–3937, 2016.

[Bunge *et al.* 2015] Roberto A. Bunge, Felipe Munera Savino et Ilan M. Kroo. *Stall/spin flight test techniques with COTS model aircraft and flight data systems.* In AIAA, pages 1–20, 2015.

[Butcher 2016] J.C Butcher. Numerical Methods for Ordinary Differential Equations. John Wiley & Sons, 2016.

[Caruana 1997] Rich Caruana. *Multitask Learning.* Machine Learning, vol. 28, no. 4, pages 41–75, 1997.

[Chandola *et al.* 2009] Varun Chandola, Arindam Banerjee et Vipin Kumar. *Anomaly detection: A survey.* ACM Computing Surveys, vol. 41, no. 3, pages 1–58, 2009.

[Chen *et al.* 2018] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt et David Duvenaud. *Neural Ordinary Differential Equations.* In NeurIPS, 2018.

[Chierchia *et al.* 16] G. Chierchia, E. Chouzenoux, P. L. Combettes et J.-C. Pesquet. *The Proximity Operator Repository. User's guide*, 2016–.

[Claeskens *et al.* 2014] Gerda Claeskens, M Hubert, Leen Slaets et K Vakili. *MFHD: Multivariate Functional Halfspace Depth.* Journal of the American Statistical Association, vol. 109, no. 505, pages 411–423, 2014.

[Clémençon & Thomas 2017] Stephan Clémençon et Albert Thomas. *Mass Volume Curves and Anomaly Ranking.* Electronic Journal of Statistics, 2017.

[Combettes & Pesquet 2011] Patrick-Louis Combettes et Jean-Christophe Pesquet. *Proximal Splitting Methods in Signal Processing.* In Fixed-

point algorithms for inverse problems in science and engineering, pages 185–212. Springer, 2011.

[Cuevas & Febrero 2007] Antonio Cuevas et Manuel Febrero. *Robust estimation and classification for functional data via projection-based depth notions*. Computational Statistics, vol. 22, no. 3, pages 481–496, 2007.

[Cuevas *et al.* 2006] Antonio Cuevas, Manuel Febrero et Ricardo Fraiman. *On the use of the bootstrap for estimating functions with functional data*. Computational Statistics and Data Analysis, vol. 51, pages 1063–1074, 2006.

[Dai & Genton 2019] Wenlin Dai et Marc G. Genton. *Directional outlyingness for multivariate functional data*. Computational Statistics and Data Analysis, vol. 131, pages 50–65, 2019.

[De Boor 1978] Carl De Boor. A practical guide to splines, volume 27. springer-verlag New York, 1978.

[Demsar 2006] Janez Demsar. *Statistical Comparisons of Classifiers over Multiple Data Sets*. Journal of Machine Learning Research, vol. 7, pages 1–30, 2006.

[Dos Santos *et al.* 2016] Ludovic Dos Santos, Ali Ziat, Ludovic Denoyer, Benjamin Piwowarski et Patrick Gallinari. *Modelling Relational Time Series using Gaussian Embeddings*. 2016.

[Dua & Graff 2017] Dheeru Dua et Casey Graff. *UCI Machine Learning Repository*, 2017.

[Erfani *et al.* 2016] Sarah M. Erfani, Sutharshan Rajasegarar, Shanika Karunasekera et Christopher Leckie. *High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning*. Pattern Recognition, vol. 58, pages 121–134, 2016.

[Fan & Li 2001] Jianqing Fan et Runze Li. *Variable Selection via Nonconcave Penalized*. Journal of the American Statistical Association, vol. 96, no. 456, pages 1348–1360, 2001.

[Febrero-bande & Oviedo de la Fuente 2012] Manuel Febrero-bande et Manuel Oviedo de la Fuente. *Statistical Computing in Functional Data Analysis : The R package fda.usc*. Journal of Statistical Software, vol. 51, no. 4, pages 1–28, 2012.

[Febrero *et al.* 2008] Manuel Febrero, Pedro Galeano et Wenceslao González-

Manteiga. *Outlier detection in functional data by depth measures , with application to identify abnormal NO x levels*. Environmetrics, vol. 19, no. August 2007, pages 331–345, 2008.

[Ferraty & Vieu 2006] Frédéric Ferraty et Philippe Vieu. Nonparametric functional data analysis: theory and practice. Springer Science & Business Media, 2006.

[Fraiman & Muniz 2001] Ricardo Fraiman et Graciela Muniz. *Trimmed means for functional data*. Test, vol. 10, no. 2, 2001.

[Friedman *et al.* 2010] Jerome Friedman, Trevor Hastie et Rob Tibshirani. *Regularization paths for generalized linear models via coordinate descent*. Journal of Statistical Software, vol. 33, no. 1, pages 1–22, 2010.

[García *et al.* 2010] Salvador García, Alberto Fernández, Julián Luengo et Francisco Herrera. *Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power*. Information Sciences, vol. 180, no. 10, pages 2044–2064, 2010.

[Gasso *et al.* 2009] Gilles Gasso, Alain Rakotomamonjy et Stéphane Canu. *Recovering sparse signals with a certain family of non-convex penalties and DC programming*. IEEE Transactions on Signal Processing, vol. 57, no. 12, pages 4686–4698, 2009.

[Goldberger *et al.* 2000] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng et H Eugene Stanley. *PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals*. Circulation, vol. 101, no. 23, pages e215–e220, 2000.

[Gong *et al.* 2013] Pinghua Gong, Changshui Zhang, Zhaosong Lu, Jianhua Z. Huang et Jieping Ye. *A General Iterative Shrinkage and Thresholding Algorithm for Non-convex Regularized Optimization Problems*. In International Conference on Machine Learning, 2013.

[Greiner 2006] Walter Greiner. Classical mechanics: point particles and relativity. Springer Science & Business Media, 2006.

[Hastie *et al.* 2009] Trevor Hastie, Robert Tibshirani et Jerome Friedman. The Elements of Statistical Learning. 2009.

[Hawkins 1980] Douglas M Hawkins. Identification of outliers, volume 11.

Springer, 1980.

[Hoerl & Kennard 1970] Arthur E Hoerl et Robert W Kennard. *Ridge regression: Biased estimation for nonorthogonal problems.* Technometrics, vol. 12, no. 1, pages 55–67, 1970.

[Hubert *et al.* 2015] Mia Hubert, Peter J. Rousseeuw et Pieter Segaert. *Multivariate functional outlier detection.* Statistical Methods and Applications, vol. 24, no. 2, pages 177–202, 2015.

[Hyndman & Shang 2010] Rob J Hyndman et Han Lin Shang. *Rainbow Plots , Bagplots , and Boxplots for Functional Data.* Journal of Computational and Graphical Statistics, vol. 19, no. 1, pages 29–45, 2010.

[Ieva & Paganoni 2013] Francesca Ieva et Anna M Paganoni. *Depth Measures for Multivariate Functional Data.* Communications in Statistics - Theory and Methods, vol. 42, no. 7, pages 1265–1276, 2013.

[Japkowicz & Stephen 2002] Nathalie Japkowicz et Shaju Stephen. *The class imbalance problem: A systematic study.* Intelligent data analysis, vol. 6, no. 5, pages 429–449, 2002.

[Jiang & Huang 2015] Dingfeng Jiang et Jian Huang. *Concave 1-norm group selection.* Biostatistics, vol. 16, no. 2, pages 252–267, 2015.

[Kuhnt & Rehage 2016] Sonja Kuhnt et André Rehage. *An angle-based multivariate functional pseudo-depth for shape outlier detection.* Journal of Multivariate Analysis, vol. 146, pages 325–340, 2016.

[Lafabregue *et al.* 2019] B. Lafabregue, J. Weber, P. Gançarski et G. Forestier. *Deep constrained clustering applied to satellite image time series.* In ECML/PKDD, Sep 2019.

[Liu *et al.* 2008] Fei Tony Liu, Kai Ming Ting et Zhi-Hua Zhou. *Isolation Forest.* In ICDM, pages 413–422, 2008.

[Long *et al.* 2018] Zichao Long, Yiping Lu, Xianzhong Ma et Bin Dong. *PDE-Net : Learning PDEs from Data.* In International Conference on Machine Learning, 2018.

[López-pintado & Romo 2009] Sara López-pintado et Juan Romo. *On the Concept of Depth for Functional Data.* Journal of the American Statistical Association, vol. 104, no. 486, pages 718–734, 2009.

[López-pintado *et al.* 2014] Sara López-pintado, Yin Sun, Juan K Lin et

Marc G. Genton. *Simplicial band depth for multivariate functional data*. Advances in Data Analysis and Classification, vol. 8, no. 3, pages 321–338, 2014.

[Mangan *et al.* 2017] Niall M Mangan, J Nathan Kutz, Steven L Brunton et Joshua L Proctor. *Model selection for dynamical systems via sparse regression and information criteria*. Proceeding of the Royal Society A, vol. 473, no. 2204, 2017.

[Markou & Singh 2003] Markos Markou et Sameer Singh. *Novelty detection: a review—part 1: statistical approaches*. Signal processing, vol. 83, no. 12, pages 2481–2497, 2003.

[Matérn 2013] Bertil Matérn. Spatial variation, volume 36. Springer Science & Business Media, 2013.

[Nason 2008] G.P. Nason. Wavelet methods in statistics with r. Springer, 2008.

[Obozinski *et al.* 2010] Guillaume Obozinski, Ben Taskar et Michael I. Jordan. *Joint covariate selection and joint subspace selection for multiple classification problems*. Statistics and Computing, vol. 20, no. 2, pages 231–252, 2010.

[Oliphant *et al.* 2001] Travis Oliphant, Pearu Peterson et Eric Jones. *Python for Scientific Computing*. Computing in Science & Engineering, vol. 9, no. 90, 2001.

[Parikh & Boyd 2013] Neal Parikh et Stephen Boyd. *Proximal Algorithms*. Foundations and Trends in Optimization, vol. 1, no. 3, pages 123–231, 2013.

[Pimentel *et al.* 2014] Marco A.F. Pimentel, David A. Clifton, Lei Clifton et Lionel Tarassenko. *A review of novelty detection*. Signal Processing, vol. 99, pages 215–249, 2014.

[Ramsay & Hooker 2017] James Ramsay et Giles Hooker. Dynamic Data Analysis. Springer Series in Statistics, 2017.

[Ramsay & Silverman 2006] Jim Ramsay et Bernard W Silverman. Functional Data Analysis. Wiley Online Library, 2006.

[Ramsay *et al.* 2009] James O. Ramsay, Giles Hooker et Spencer Graves. Functional Data Analysis with R and MATLAB. Springer Science\& business Media, 2009.

[Rasmussen 2003] Carl Edward Rasmussen. *Gaussian processes in machine learning*. In Summer School on Machine Learning, pages 63–71. Springer, 2003.

[Rehage 2016] Andre Rehage. *Functional Tangential Angle Pseudo-Depth*, 2016. R package version 0.1.0.

[Rousseeuw & Hubert 2018] Peter J Rousseeuw et Mia Hubert. *Anomaly detection by robust statistics*. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 8, no. 2, page e1236, 2018.

[Rudy *et al.* 2019] Samuel Rudy, Alessandro Alla, Steven L. Brunton et J. Nathan Kutz. *Data-driven identification of parametric partial differential equations*. SIAM Journal on Applied Dynamical Systems, vol. 18, no. 2, pages 643–660, 2019.

[Ruff *et al.* 2018] Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Uller et Marius Kloft. *Deep One-Class Classification*. In ICML, pages 4390–4399, 2018.

[Schaeffer & McCalla 2017] Hayden Schaeffer et Scott G McCalla. *Sparse model selection via integral terms*. Physical Review E, vol. 96, no. 2, page 023302, 2017.

[Schaeffer *et al.* 2018] Hayden Schaeffer, Giang Tran et Rachel Ward. *Extracting sparse high-dimensional dynamics from limited data*. SIAM Journal of Applied Dynamical Systems, vol. 78, no. 6, 2018.

[Schaeffer 2017] Hayden Schaeffer. *Learning partial differential equations via data discovery and sparse optimization*. Proceeding of the Royal Society A, vol. 573, 2017.

[Schlather *et al.* 2015] Martin Schlather, Alexander Malinowski, Peter J Menck, Marco Oesting, Kirstin Strokorb*et al.* *Analysis, simulation and prediction of multivariate random fields with package RandomFields*. Journal of Statistical Software, vol. 63, no. 8, pages 1–25, 2015.

[Schmidt & Lipson 2009] Michael D. Schmidt et Hod Lipson. *Distilling Free-Form Natural Laws from Experimental Data*. Science, vol. 324, pages 81 – 85, 2009.

[Schölkopf *et al.* 2001] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola et Robert C Williamson. *Estimating the support*

of a high-dimensional distribution. Neural computation, vol. 13, no. 7, pages 1443–1471, 2001.

[Sheskin 2003] David J Sheskin. Handbook of parametric and nonparametric statistical procedures. crc Press, 2003.

[Simon *et al.* 2013] Noah Simon, Jerome Friedman, Trevor Hastie et R O B Tibshirani. *A sparse-group lasso*. Computational and Graphical Statistics, pages 1–13, 2013.

[Srivastava & Klassen 2016] Anuj Srivastava et Eric P Klassen. Functional and Shape Data Analysis. Springer Series in Statistics, 2016.

[Stoer & Bulirsch 2013] Josef Stoer et Roland Bulirsch. Introduction to numerical analysis, volume 12. Springer Science & Business Media, 2013.

[Sun & Genton 2011] Ying Sun et Marc G. Genton. *Functional Boxplots*. Journal of Computational and Graphical Statistics, vol. 20, no. 2, pages 316–334, 2011.

[Tishbirani 1996] Robert Tishbirani. *Regression shrinkage and selection via the Lasso*. Journal of the Royal Statistical Society. Series B, vol. 58, no. 1, pages 267–288, 1996.

[Torres *et al.* 2011] J. Martínez Torres, P.J. Garcia Nieto, L. Alejano et A.N. Reyes. *Detection of outliers in gas emissions from urban areas using functional data analysis*. Journal of Hazardous Materials, vol. 186, no. 1, pages 144 – 149, 2011.

[Tuddenham & Snyder 1954] Read D. Tuddenham et Margaret M. Snyder. *Physical growth of California boys and girls from birth to eighteen years*. Publications in child development. University of California, Berkeley, vol. 1, no. 2, page 183—364, 1954.

[Tukey 1975] John W Tukey. *Mathematics and the picturing of data*. In Proceedings of the International Congress of Mathematicians, Vancouver, 1975, volume 2, pages 523–531, 1975.

[Wei & Keogh 2006] Li Wei et Eamonn Keogh. *Semi-supervised time series classification*. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 748–753. ACM, 2006.

[Xie *et al.* 2017] Weiyi Xie, Sebastian Kurtek, Karthik Bharath et Ying Sun.

A Geometric Approach to Visualization of Variability in Functional Data. Journal of the American Statistical Association, vol. 112, no. 519, pages 979–993, 2017.

[Xie *et al.* 2019] Weiyi Xie, Oksana Chkrebtii, Sebastian Kurtek et Senior Member. *Visualization and Outlier Detection for Multivariate Elastic Curve Data*. IEEE Transactions on Visualization and Computer Graphics, 2019.

[Yuan & Lin 2006] Ming Yuan et Yi Lin. *Model selection and estimation in regression with grouped variables*. Journal of the Royal Statistical Society. Series B: Statistical Methodology, vol. 68, no. 1, pages 49–67, 2006.

[Zhang & Schaeffer 2019] Linan Zhang et Hayden Schaeffer. *On the Convergence of the SINDy Algorithm*. Multiscale Modeling and Simulation, vol. 17, no. 3, pages 948–972, 2019.

[Zhang 2010] Cun Hui Zhang. *Nearly unbiased variable selection under minimax concave penalty*. The Annals of Statistics, vol. 38, no. 2, pages 894–942, 2010.

[Zou & Hastie 2005] Hui Zou et Trevor Hastie. *Regularization and variable selection via the elastic net*. Journal of the Royal Statistical Society. Series B, vol. 67, no. 2, pages 302–320, 2005.

[Zuo & Serfling 2000] Yijun Zuo et Robert Serfling. *General Notions of Statistical Depth Function*. The Annals of Statistics, vol. 28, no. 2, pages 461–482, 2000.