Journal of Multivariate Analysis 170 (2019) 221–231

Contents lists available at ScienceDirect

Journal of Multivariate Analysis

journal homepage: www.elsevier.com/locate/jmva

Quantifying prediction uncertainty for functional-and-scalar to functional autoregressive models under shape constraints

Jacopo Rossini [*], Antonio Canale

Department of Statistical Sciences, University of Padova, Italy

ARTICLE INFO

Article history:
Received 29 November 2017
Available online 26 October 2018

Keywords:
Demand and offer model
Functional bootstrap
Functional ridge regression

ABSTRACT

Motivated by demand and supply curve forecasting in energy markets, we discuss an autoregressive functional modeling framework that preserves curve constraints, includes exogenous scalar information, and provides prediction uncertainty quantification. The model is a functional autoregressive model that relies on a non-concurrent functional autoregressive model in a non-standard pre-Hilbert space in order to satisfy the curve constraints. Prediction uncertainty is quantified by means of a novel bootstrap approach for dependent functional data where the predictive bootstrap trajectories are represented alongside the prediction to show how forecasting confidence varies in the domain. Computational and numerical details are discussed in order to replicate the model estimation process an adequate number of times during the bootstrap phase. The method is applied to Italian natural gas market data.

1. Introduction

Functional data analysis (FDA) drew a lot of attention over the past 20 years and currently features among the central research themes in statistics. Indeed, we are increasingly facing problems where the data consist of curves, images, or surfaces. FDA has been popularized through the books by Ramsay and Silverman [33,34] and the related R package [30] which is often used by practitioners. Additional important references include the books by Bosq [3], Ferraty and Vieu [13], and Horváth and Kokoszka [19], as well as the recent reviews [8,15].

In this paper we are interested in analyzing functions that are time dependent. We focus on a specific time dependence structure, i.e., functional autoregression. If $\{f_1, \dots, f_T\}$ is a set of chronologically ordered curves in L^2, the simplest functional autoregression model (FAR) can be written as

$$f_t = \alpha + \Psi(f_{t-1}) + \varepsilon_t, \tag{1}$$

where Ψ is a general operator satisfying different condition ranging from simple linear assumptions [3] to nonparametric assumptions [13]. Here we focus on simple linear FAR models for their ease of interpretation and their good performance in many applications. Linear FAR models are the natural extension of linear autoregressive models [4] where the autoregressive scalar parameters are replaced by autoregressive Hilbert–Schmidt operators.

We are motivated by a forecasting problem related to the Italian natural gas market. Specifically, we are interested in predicting future functional observations of the bivariate functional time series of demand and supply curves. These functions are observed daily and represent the outcome of a bidding dynamic. This problem has recently been studied

[*] Corresponding author.
E-mail addresses: jacopo.rossini@studenti.unipd.it (J. Rossini), canale@stat.unipd.it (A. Canale).

https://doi.org/10.1016/j.jmva.2018.10.007

in [6,7,21]. Refer to Section 4 for more details. The specific application at hand involves additional constraints that need to be considered in the model formulation. Specifically, the demand and offer curves are strictly monotone, bounded from above and below, and with an equality constraint on the lower bound of the domain.

Modeling monotonicity has been explored by several authors; see, e.g., [25,27,29,32,39]. Earlier work focused mainly on finding a set of coefficients of a basis representation that satisfies the monotonicity constraint while minimizing some error metric. This approach can give good results but can also be computationally expensive. It relies heavily on a smart specification of the basis system, because otherwise finding a suitable solution might prove impractical or unfeasible. It is a great idea for one-time analysis, but it can be tricky to implement successfully in repeated applications. Alternative methods consist in estimating the monotonic function without the monotonicity constraint and then projecting it in the convex subspace of monotonic functions [14,26].

The specific constraints of these functional data led Canale and Vantini [7] to develop a method tailored for this class of problems, in which monotonicity and constraints at the edges of the domain are imposed by means of the so-called transform/back-transform method [34]. The method exhibits good one-step-ahead forecasting potential but provides no uncertainty quantification of the functional point prediction. In addition, it discards some valuable exogenous information available at prediction time.

This paper starts from the approach in [7] and tries to solve these two issues. The idea of prediction uncertainty quantification naturally calls for a Bayesian approach. Previous work along these lines includes [6] and [21], which were motivated by the same application to the natural gas market and modeled the functional time series by means of a latent particle system. In general, under a Bayesian approach the uncertainty is naturally quantified by means of probabilities and Bayes' rule provides a mathematically elegant mechanism to update those probabilities based on the data. The Bayesian approach needs a probabilistic data generating model assumption (the likelihood) and a probabilistic prior uncertainty quantification of the unknown quantities of the models (the prior distribution).

In the present context as in many other FDA applications, a precise family of data generating processes is not assumed. Our aim here is to start from [7] where an estimator of Ψ in (1) is obtained by minimizing an objective function. To account for uncertainty quantification under this formulation, we propose a bootstrap procedure which is suitable for dependent data. Adapting the original bootstrap procedure by Efron [12] to dependent data can be done in various ways. A good overview of the most popular methods can be found in the book by Lahiri [22]. The two main approaches are the so-called block bootstrap and residual bootstrap. In the former, the data are divided into several blocks –to preserve the original time series structure within a block –that are resampled as in the standard bootstrap approach. In this paper we use this approach and specifically a moving block bootstrap specification. In the residual bootstrap approach, instead, the resampling bootstrap procedure is done on the residuals of a suitably fitted model. Recent contributions to the FDA literature that follow this approach are [31,40].

Since direct minimization of a suitable penalized error function within the bootstrap procedure is time consuming, we pay particular attention to the computational efficiency of the estimation procedure and provide a way of speeding up the process, with the ultimate goal of achieving the result in a few hours on a laptop-like machine while reducing the underlying approximation of the functional data. See Section 2.2 for additional details.

To solve the second issue of [7], we discuss how to include in the non-concurrent functional autoregressive model a set of scalar covariates and specify an estimation procedure based on the minimization of a specific penalized error function. This helps the model to capture additional information and provides an acceptable way of introducing seasonality or periodic components into the analysis. The introduction of scalar covariates in functional time series is not new; see, e.g., [1,9].

The proposed methodology will be presented in the context of the daily auctions needed to balance the Italian natural gas distribution system. This is the same application as originally considered in [7], but in this paper we analyze new data related to a more mature and dynamic phase of the market. From a practical point of view, the challenge is to model small fluctuations of strongly stationary curves, a context in which the penalized minimization finds its best application.

The rest of the paper is organized as follows. In Section 2, we review the non-concurrent functional autoregressive models of [7] that preserve curve constraints; we also describe the estimation method and provide a novel numerical solution relaxing the computational burden of the procedure. Section 3 introduces the moving block bootstrap procedure applied to this functional time series framework. Section 4 presents the application of the proposed method to the motivating dataset on daily demand and offer curves prediction in the Italian natural gas market. Section 5 contains some concluding remarks. Additional numerical details, data preprocessing considerations, and empirical evidence of consistency of the estimators are reported in the Appendix .

2. Model and estimation

Let $\mathscr{M}^2(a,b)$ be the family of differentiable curves –generically referred to as g –such that $g: [a,b] \to [0,1]$, $g(a) = 0$, $g(b) < 1$, and $0 < g'(s) < G < \infty$ for every $s \in [a,b]$. Following [7], we let

$$f(s) = \ln\left\{\frac{g'(s)}{1-g(s)}\right\}, \tag{2}$$

for every $s \in [a,b]$ be the image of $g \in L^2(a,b)$ with corresponding inverse

$$g(s) = 1 - \exp\left[-\int_a^s \exp\{f(u)\}du\right],$$

for every $s \in [a, b]$. Following [7], the sequence $g_1, \ldots, g_T$ is modeled with an $\mathscr{M}^2$-FAR model of order p if

$$f_t = \alpha + \sum_{j=1}^{p} \psi_j f_{t-j} + \varepsilon_t, \tag{3}$$

and f_t is obtained with (2). Equivalently, the model can be written as

$$f_t(s) = \alpha(s) + \sum_{j=1}^{p} \int_a^b \psi_j(s, u) f_{t-j}(u) du + \varepsilon_t(s).$$

The inclusion in model (3) of scalar information is done in accordance with the procedure commonly used in many FDA applications, i.e., by multiplying the scalar value by a constant function. This is consistent with analogous methods presented in [34] for the basis system, but requires additional care in the present context because of the penalization imposed in the optimization process. The $\mathscr{M}^2$-FAR model of order p with d covariates is then defined as

$$f_t = \alpha + \sum_{j=1}^{p} \psi_j f_{t-j} + \sum_{k=1}^{d} \varphi_k c_{t,k} + \varepsilon_t$$

or equivalently, for all $s \in [a, b]$, by

$$f_t(s) = \alpha(s) + \sum_{j=1}^{p} \int_a^b \psi_j(s, u) f_{t-j}(u) du + \sum_{k=1}^{d} \int_a^b \varphi_k(s, u) c_{t,k}(u) du + \varepsilon_t(s). \tag{4}$$

The functional time series f_t is obtained by mapping the time series g_t via (2). In the above equations, ψ_j and φ_k are Hilbert–Schmidt operators; $\psi_j(s, u) \in \mathscr{L}^2\{(a, b) \bigtimes (a, b)\}$ and $\varphi_j(s, u) \in \mathscr{L}^2\{(a, b) \bigtimes (a, b)\}$ are the respective kernels. Moreover, $\varepsilon_t(s)$ are zero mean innovations with finite variance and $\alpha(s)$ is a non-centrality function. The result of the multiplication of a scalar value c_t of the kth scalar variable by the constant, 1-valued function is denoted $c_{t,k}$.

2.1. Estimation

It is crucial to develop an estimation procedure capable of dealing with the radically different nature of the autoregressive and scalar-turned-function section of the model. The estimation of the Hilbert–Schmidt operators can be obtained by direct minimization of a suitable penalized error function. Following [7], we will introduce a penalization for the sum of the squared Hilbert–Schmidt norms of the operators ψ and φ. This is going to introduce a ridge-like penalization [17]. To specify it, note that in the scalar ridge regression it is common practice to standardize the variables considered. This is done to avoid penalizing larger and smaller coefficients differently, resulting in a non-homogeneous shrinkage which depends on the units in which the variables are measured. To address this issue in the functional context at hand, a second penalization term is introduced for the scalar part of the model, with the benefit of being able to tune the penalization separately for the autoregressive operators and the scalar-turned-functional operators. Clearly, the latter are going to be standardized consistently with the common practice of ridge regression.

We consider only ridge-like penalizations from the whole set of popular penalties –like the Lasso [36] or the Elastic-Net [41] –to leverage the analytical solution of the ridge problem, which makes the computations much faster. The estimation procedure is formalized below.

Theorem 1. *Let $g_1, \ldots, g_T$ be a time series in $\mathscr{M}^2(a, b)$ and $f_1, \ldots, f_T$ be the related transformed series via (2). Let $\| \cdot \|_{L^2}$ be the L^2 norm and $\| \cdot \|_{HS}$ be the Hilbert–Schmidt norm. Then, for any positive constants $\lambda_1 > 0$ and $\lambda_2 > 0$, the solution of the minimization problem*

$$\min_{\alpha \in L^2, \psi_j, \varphi_k \subseteq HS} \left\{ \sum_{t=p+1}^{T} \left\| f_t - \left(\alpha + \sum_{j=1}^{p} \psi_j f_{t-j} + \sum_{k=1}^{d} \varphi_k c_{t,k} \right) \right\|_{L^2}^2 + \lambda_1 \sum_{j=1}^{p} \|\psi_j\|_{HS}^2 + \lambda_2 \sum_{k=1}^{d} \|\varphi_k\|_{HS}^2 \right\} \tag{5}$$

is unique and provides estimators for the unknown centrality parameter $\alpha \in L^2$, and Hilbert–Schmidt operators $\psi_1, \ldots, \psi_p$ and $\varphi_1, \ldots, \varphi_d$.

Proof. It is a trivial extension of the proof of Theorem 1 of [7]. The uniqueness of the estimators ψ_j and φ_k can be verified by noting that, for fixed ψ_j and φ_k,

$$\hat{\alpha} = \frac{1}{T-p} \sum_{t=p+1}^{T} \left(f_t - \sum_{j=1}^{p} \psi_j f_{t-j} - \sum_{k=1}^{d} \varphi_k c_{t,k} \right) = \bar{f}_{[0]} - \sum_{j=1}^{p} \psi_j \bar{f}_{[j]} - \sum_{k=1}^{d} \varphi_k \bar{c}_k,$$

where $\bar{f}_{[j]} = (f_{p+1-j} + \cdots + f_{T-j})/(T-p)$ and $\bar{c}_k = (c_{p+1,k} + \cdots + c_{T,k})/(T-p)$. Then $\hat{\alpha}$ can be included in

$$\sum_{t=p+1}^{T} \left\| (f_t - \bar{f}_{[0]}) - \sum_{j=1}^{p} \psi_j (f_{t-j} - \bar{f}_{[j]}) - \sum_{k=1}^{d} \varphi_k (c_{t,k} - \bar{c}_k) \right\|_{L^2}^2 + \lambda_1 \sum_{j=1}^{p} \|\psi_j\|_{HS}^2 + \lambda_2 \sum_{k=1}^{d} \|\varphi_k\|_{HS}^2 .$$

For every Hilbert–Schmidt operator defined in the last equation, the last two summands can be expressed as

$$\lambda_1 \sum_{j=1}^{p} \sum_{\ell \in \mathbb{N}} \|\psi_j \phi_\ell\|_{L^2}^2 + \lambda_2 \sum_{k=1}^{d} \sum_{\ell \in N} \|\varphi_k \phi_\ell\|_{L^2}^2$$

with $\{\phi_k : k \in \mathbb{N}\}$ an arbitrary orthonormal basis of L^2. In the end, what we have is a linear combination of a positive semidefinite quadratic form and a positive definite quadratic form. This makes the expression a positive definite quadratic form in regard to the operators, admitting one single minimum.

Thanks to the Fubini–Tonelli Theorem, we can write $\|\psi_j\|_{HS}^2 = \int_a^b \{\int_a^b \psi_j^2(s,u) du\} ds$ (similarly for $\|\varphi_k\|_{HS}^2$) and carry out the minimization for every $s \in [a,b]$ independently, viz.

$$\sum_{t=p+1}^{T} \Bigg[\{f_t(s) - \bar{f}_{[0]}(s)\} - \sum_{j=1}^{p} \int_a^b \psi_j(s,u) \{f_{t-j}(u) - \bar{f}_{[j]}(u)\} du$$
$$- \sum_{k=1}^{d} \int_a^b \varphi_k(s,u) \{c_{t,k}(u) - \bar{c}_k(u)\} du \Bigg]^2 + \lambda_1 \sum_{j=1}^{p} \int_a^b \psi_j^2(s,u) du + \lambda_2 \sum_{k=1}^{d} \int_a^b \varphi_k^2(s,u) du \qquad (6)$$

on $\psi_1(s,\cdot), \ldots, \psi_p(s,\cdot)$ and $\varphi_1(s,\cdot), \ldots, \varphi_d(s,\cdot)$ for all $s \in [a,b]$. □

The relation of the estimator obtained under (5) and the ridge regression estimator leads to argue about the properties of the estimation procedure outlined in Theorem 1. For example, an issue worth investigation is related to its asymptotic behavior. Rather than prove formally the consistency of the estimator of the Hilbert–Schmidt operators, we performed a simulation study which empirically shows that the mean integrated squared error

$$\mathrm{E}\left[\iint \{\psi_j(s,u) - \hat{\psi}_j(s,u)\}^2 ds du \right],$$

goes to zero for each $j \in \{1, \ldots, p\}$ as the length of the functional time series increases. More details on this simulation study are reported in the Appendix .

When the final goal is not just parameter estimation but curve prediction, as in the motivating application discussed in Section 4, the plug-in estimator

$$\hat{g}_{T+1} = \mathrm{lnH}^{-1} \left\{ \hat{\alpha} + \sum_{j=1}^{p} \hat{\psi}_j \mathrm{lnH}(g_{T+1-j}) + \sum_{k=1}^{d} \hat{\varphi}_k \mathrm{lnH}(c_{T+1}) \right\},$$

can be used. Consistently with [7], the properties of the spaces $\mathscr{M}^2$ and L^2 ensure that $\hat{g}_{T+1}$ will satisfy all the constraints that characterize the space $\mathscr{M}^2(a,b)$.

2.2. Computational considerations

Replicating the whole minimization B times to obtain the bootstrap estimates, as discussed in the next section, quickly adds up in terms of computational cost. Eq. (6) requires several numeric evaluations of integrals, and numerically finding the minimum is impractical if not unfeasible. An approximation is presented in what follows, along with some considerations, to obtain a procedure that can be completed in a few hours on a consumer-grade laptop.

First, the integrals in Eq. (4) will be approximated by rectangles for every $s \in [a,b]$, i.e.,

$$f_t(s) \approx \alpha(s) + \sum_{j=1}^{p} \sum_{u=1}^{z} c \times \psi_j(s,u) f_{t-j}(u) + \sum_{k=1}^{d} \sum_{u=1}^{z} c \times \varphi_k(s,u) c_{t,k} + \varepsilon_t(s)$$
$$= \alpha(s) + \sum_{j=1}^{p} \sum_{u=1}^{z} \tilde{\psi}_j(s,u) f_{t-j}(u) + \sum_{k=1}^{d} \tilde{\varphi}_k(s) c_{t,k} + \varepsilon_t(s),$$

where c is the distance between two consecutive s_i and s_{i+1}, $c_{t,k}(u)$ is a constant function, independent from u, and

$$\sum_{u=1}^{z} c \times \varphi_k(s,u) c_{t,k} = \tilde{\varphi}_k(s) c_{t,k}$$

is a trivial simplification. It is immediate that approximating the integrals about $c_{t,k}$ is computationally negligible, thanks to the function being constant. Details on the specific choice of z are reported in the Appendix .

The same kind of approximation must be applied to the Hilbert–Schmidt operators, which will be approximated by a matrix with z rows and d columns, viz.

$$\lambda_1 \|\psi_j\|_{HS}^2 = \lambda_1 \int_a^b \left\{ \int_a^b \psi_j^2(s,u) du \right\} ds \approx \lambda_1 \sum_{s=1}^{z} \left\{ \sum_{u=1}^{z} \psi_j^2(s,u) \right\},$$

$$\lambda_2 \|\varphi_k\|_{HS}^2 = \lambda_2 \int_a^b \left\{ \int_a^b \varphi_k^2(s,u) du \right\} ds = \lambda_2 \sum_{s=1}^{z} \left\{ \sum_{u=1}^{z} \varphi_k^2(s,u) \right\} \approx \lambda_2 \sum_{s=1}^{z} \tilde{\varphi_k}^2(s).$$

The sum of all terms is recognizable as an example of a classical ridge regression as discussed in Remark 1 of [7]. Hence, similarly to the classical ridge estimator, we can define, for each s on a suitable equispaced grid of size z partitioning the domain (a, b),

$$\hat{\beta}_s = (X^\top X + \Lambda)^{-1} X^\top Y_s,$$

with

$$\Lambda = \begin{bmatrix} 0 & \dots & \dots & \dots & 0 \\ \vdots & \lambda_1 & 0 & \dots & \vdots \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & 0 & \lambda_2 \end{bmatrix},$$

where λ_1 appears zp times in the diagonal of Λ, and λ_2 appears d times. All the resulting vectors are then stacked in a square matrix yielding $\tilde{\psi}_j(s,u)$ and $\tilde{\varphi}_k(s)$. Every minimization requires a single evaluation of Eq. (6) for every s, making it much more efficient than optimizing the equation numerically. This also means that parallel implementations are straightforward. Moreover, having the matrix form of the minimization makes it possible to use many computationally efficient libraries for linear algebra, speeding up the process significantly.

The other computationally burdensome operation left is the inversion of a $(zp + d + 1) \times (zp + d + 1)$ matrix. This can quickly become a problem considering how many times the minimization has to be carried out, so the matrix form just presented is adjusted to perform more efficiently at high values of z. The following identity [18] is used to restructure the minimization matrix form:

$$(A + UV)^{-1} = A^{-1} - A^{-1}U(I_{n\times n} + VA^{-1}U)^{-1}VA^{-1}.$$

The first part of the ridge matrix estimator can be expressed as

$$(\Lambda + X^\top X)^{-1} = \Lambda^{-1} - \Lambda^{-1}X^\top(I_{n\times n} + X\Lambda^{-1}X^\top)^{-1}X\Lambda^{-1},$$

so the entire estimator is

$$\hat{\beta}_s = \left(\Lambda + X^\top X\right)^{-1} X^\top Y = (\Lambda^{-1} - \Lambda^{-1}X^\top\left(I_{n\times n} + X\Lambda^{-1}X^\top\right)^{-1}X\Lambda^{-1})X^\top Y,$$

with complexity $\mathcal{O}\{(zp + d + 1)T^2\}$.

3. Moving block bootstrap for functional data

Classical bootstrap procedures fail when any form of m-dependency is introduced in the data. As discussed in the Introduction, among the possible remedies proposed to deal with this issue, the most popular are bootstrapping the residuals of a model –which should be independent –or implement the so-called block bootstrap. Both approaches require additional assumptions: the former method needs a sound model to compute the residuals for resampling, while the latter heavily depends on the block size.

As stated in the Introduction, we will exploit a block bootstrap approach. Indeed, tuning the model for the best performance already requires some trade-offs to accommodate the high dimensionality of the problem. Having the bootstrap procedure to strictly depend on this tuning and on the model specification seemed risky, which is why a more stand-alone procedure as the block bootstrap was preferred. It is worth noting that while finite dependence is often assumed in time series modeling, it may be of interest to test for the stationarity of the process, an important assumption of the block bootstrap procedure. In the context of constrained functional time series, and specifically the mature market whose supply and demand curves will be analyzed, the stationarity assumption seemed reasonable even without a formal test. Testing for functional time series stationarity is still a matter of intense research and we refer to the recent contributions [2,10] for further reading.

We specify the moving block bootstrap procedure as it follows. Functional data are divided in $N = T - \ell + 1$ blocks so that $B_t = \{f_t, \dots, f_{t+\ell-1}\}$ with $t \in \{1, \dots, N\}$. Then, R instances of resampling are created, each one having T/ℓ blocks. The

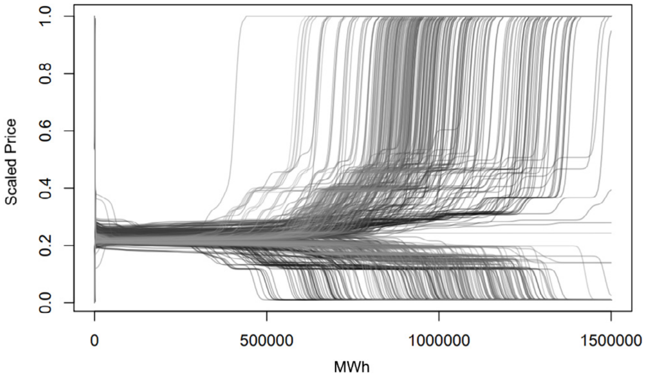

Fig. 1. Smoothed functional time series of demand and offer curves. Color denotes time, e.g., older curves are darker.

selection of an optimal block length ℓ has been addressed both from a theoretical and empirical stand-point. In particular, Hall and Horowitz [16] recommend $\ell = T^{1/3}$ when estimating bias and variance while Inoue and Shintani [20] select ℓ with an automatic procedure leading to $\ell = 3.5$ and $\ell = 6$ for $T = 64$ and $T = 128$, respectively. Also, a nonparametric plug-in method has been developed in [23] to chose an MSE-optimal ℓ.

The end result is a set of $\hat{\alpha}^*$, $\hat{\psi}_j^*$, and $\hat{\varphi}_k^*$. Every iteration of the resampling is used to estimate the needed operators and predict the curve $\hat{g}^*_{T+1}$. In this way several bootstrap forecasts are obtained, giving useful indications as to the possible trajectory of the future curve. From these B trajectories, point-wise confidence intervals can be plotted along with many other quantities. The main strength of this approach, however, is to give more detailed information about the position of the possible curve in a graphical, intuitive way. This is illustrated next.

4. Application to the Italian Natural Gas Balancing Platform

In this section we analyze data from the Italian Natural Gas Balancing Platform (PB-GAS). The PB-GAS works as follows. On a given day, the entity responsible for gas transportation (Snam, in Italy) takes care of the daily compensation of the imbalance between the gas injections and the actual consumption by submitting a demand bid (in case of gas shortage) or a supply offer (in case of gas excess) for a volume equal to the global imbalance. At the same time, each trader can submit demand and supply bids. The bids are then collected to build offer and demand curves. The intersection between the offer and demand curves provides the exchange price, and every offer/demand below/above it is accepted.

The method described in Sections 2 and 3 is now applied with the final goal of providing useful information to elaborate bidding strategies. The available data span October 2016 to September 2017 and represent a fresher version of the dataset analyzed in [7]. The raw data are depicted in Fig. 1, where the quantity exchanged is on the x-axis, while the price –rescaled between 0 and 1 –is on the y-axis. Color denotes time dependence, with older curves being darker. Details on the data preprocessing are reported in the Appendix .

To test the model performance, the last 20 days of the available data were used as a test set. The model is estimated day-by-day, obtaining the prediction of the following day and measuring the discrepancy between this result and the real curve. This approach mimics the practical use of the model in the real world.

The following testing procedure was used for the selection of every parameter in the model, i.e., the autoregressive order p, and the two penalty parameters λ_1 and λ_2. Moreover, the selection process was done considering the first third of the domain, since from a practical standpoint it is by far the most useful section, where the intersection between the curves is obtained, while also being less noisy. Most of the offers, in fact, happen in this section, and every intersection of the curves is in this region. In other contexts and applications, however, it may be more natural and reasonable to use the whole domain of the curves.

First, we discuss the choice of the autoregressive order p. Two models, with $p = 1$ and $p = 2$, were compared. High values of p should be avoided if possible, given the computational burden that they entail. From [7], it appears that $p = 1$ should be sufficient, even though this choice was based on older data (from 2013). For both models, an independent selection of the penalization parameters –as discussed later –was carried out to obtain the best performing candidate models. On the whole test set, the model with $p = 2$ performed 3% worse than the model with $p = 1$ regarding curve prediction. As far as the prediction of the daily price, a byproduct of the procedure easily obtainable as the intersection of the demand and supply curve, the model with $p = 2$ performed 16% worse. This degradation in effectiveness of the model with higher p, along with computational considerations, made testing the performance of $p = 3$ unattractive. The reference model was thus chosen to have $p = 1$.

The most critical aspect of the whole model is the selection and the effect of parameters λ_1 and λ_2. They tune how close to interpolation the model gets, a critical trade-off when considering forecasting as a priority.

Table 1
Error based on λ_1 (row) and λ_2 (columns) for the demand and supply curve, multiplied by a factor of 100.

	Demand curves								
	λ_2								
		10^7	1000	100	10	1	0.1	0.01	10^{-15}
λ_1	10^{13}	5.22	2.09	0.60	0.65	0.68	0.68	0.68	0.68
	10^{12}	5.22	2.09	**0.59**	0.65	0.68	0.68	0.68	0.68
	10^{11}	5.22	2.09	0.60	0.65	0.68	0.68	0.68	0.68
	10^{10}	5.22	2.09	0.60	0.65	0.68	0.68	0.68	0.68
	10^9	5.22	2.09	0.60	0.65	0.68	0.68	0.68	0.68
	10^8	5.22	2.09	0.60	0.65	0.68	0.68	0.68	0.68
	10^7	5.24	2.11	0.61	0.65	0.69	0.69	0.69	0.69
	10^6	5.33	2.25	0.69	0.74	0.77	0.78	0.78	0.78
	10^5	5.61	3.01	1.36	1.39	1.42	1.43	1.43	1.43
	10^4	6.58	4.63	2.78	2.69	2.72	2.73	2.73	2.73
	10^3	7.69	6.34	4.45	4.10	4.08	4.08	4.08	4.08
	Supply curves								
	λ_2								
		10^7	1000	100	10	1	0.1	0.01	10^{-15}
λ_1	10^{13}	9.06	9.42	10.06	10.00	9.99	9.99	9.99	9.99
	10^{12}	**9.06**	9.42	10.06	10.00	9.99	9.99	9.99	9.99
	10^{11}	9.06	9.42	10.06	10.00	9.99	9.99	9.99	9.99
	10^{10}	9.06	9.42	10.06	10.00	9.99	9.99	9.99	9.99
	10^9	9.07	9.42	10.06	10.00	9.99	9.99	9.99	9.99
	10^8	9.11	9.46	10.06	10.00	9.99	9.99	9.99	9.99
	10^7	9.34	9.63	10.06	9.97	9.96	9.96	9.96	9.96
	10^6	9.47	9.66	9.89	9.76	9.74	9.74	9.74	9.74
	10^5	9.19	9.30	9.50	9.37	9.34	9.34	9.34	9.34
	10^4	9.60	9.65	9.71	9.54	9.52	9.51	9.51	9.51
	10^3	11.44	11.35	11.18	10.96	10.94	10.95	10.95	10.95

Table 2
Scalar covariates available at prediction time in the PB-GAS application.

	Value	Minimum	Mean	Maximum
Volume (MWh)	Real	1169	43 600	216 600
Number of active operators	Integer	2	8.092	16
Imbalance type (1 = demand)	Binary	0	0.4375	1
Positive unbalance (MWh)	Real	0	12 060	216 600
Negative unbalance (MWh)	Real	0	12 780	173 300
Daily price (€/MWh)	Real	16	18.70	22.38
Mean temperature (°C)	Real	−1.22	15.19	34.02
Month	Dummy	–	–	–
Weekday	Dummy	–	–	–

Using the same procedure as before, several techniques were used in order to find the best combination of λ_1 and λ_2. An initial research conducted with common optimization procedures such as Bayesian optimization [28] and particle swarm optimization [11] led to the selection of very specific parameters. To check the effectiveness of these values, a grid search on a wide range of values was also conducted, leading to similar results.

Table 1 shows the results on a fairly wide range of values to give a sense of the optimization surface. The search conducted with the two above mentioned algorithms was carried out on a significantly wider range of values in order to avoid leaving out any potential candidate. The optimization was carried out separately for the demand and supply time series. The comparisons were based on an approximation of the L_2 mean squared errors computed on the 20 curves of the test, viz.

$$L_2\text{-MSE}(\lambda_1, \lambda_2) = \frac{1}{20}\sum_{D=1}^{20}\int \{f_D(s) - \hat{f_D}(s; \lambda_1, \lambda_2)\}^2 ds$$

where $\hat{f_D}(s; \lambda_1, \lambda_2)$ is the predicted curve for day D, for fixed λ_1 and λ_2.

One of the innovations of the proposed method, over the original contribution [7], is the inclusion of scalar exogenous variables into the model. This is motivated by the application at hand where, at prediction time, all the quantities described in Table 2 are available.

To test the performance of the proposed functional model with scalar covariates, we compared it with the original approach in [7], i.e., fitting a model with the same procedure but without scalar information. The difference of L_2-MSE errors is presented in Table 3. The performance gain is appreciable for the demand curve but it is negligible for the supply curve. Again, as a byproduct of the better prediction of the curves, the forecasting of the price is also positively affected.

Table 3
Prediction errors with or without scalar information: L_2-MSE for the curves and scalar MSE for the intersection price.

	With covariates	Without covariates
Demand curve	0.0059	0.0522
Supply curve	0.0906	0.0906
Price (€/MWh)	0.2051	0.4754

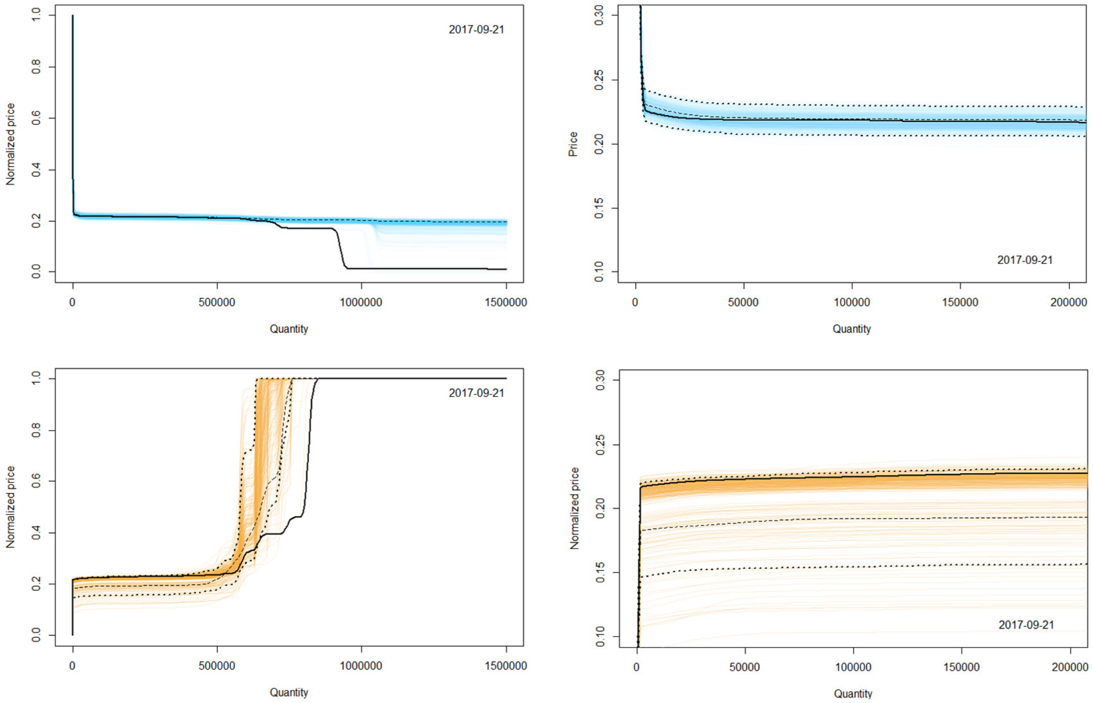

Fig. 2. Functional forecasting for September 21, 2017 of the demand (upper plots) and offer (lower plots) curves: real curve (solid), functional prediction (dashed), and 95% confidence bands (dotted). The colored lines are the bootstrap trajectories. The right panels zoom in on the curves' domain where all the intersections happen.

The other main innovation that we propose is to provide a blocked bootstrap solution to quantify the prediction uncertainty. In what follows the size of the block bootstrap procedure described in Section 3 is $\ell = 10$. This prediction uncertainty quantification can be obtained by means of graphical solutions, such as those represented in Fig. 2 reporting the prediction for September 21, 2017 in the test set. The left column represents the curves in their entire domain while the right column zooms in on the first part of their domain, i.e., to the portion of the domain where all the exchanges happen and where it is critical to get a good prediction. From the upper panels of Fig. 2, it is clear that the point prediction for the demand curve for this specific day (dashed line) is precise if compared to the real curve (solid line) with most of the bootstrap trajectories being concentrated around the real curve. Considering the lower panels of Fig. 2, instead, we can see that the prediction for the supply curve (dashed line) is a bit off the real curve (solid line), but the bootstrap trajectories clearly mark a denser area in which there is a good chance of finding the real curve, and indeed the real curve is really close to this area. Hence, the bootstrap trajectories give to traders the power to adapt their conclusions based on the uncertainty they convey.

Comparing the whole method to other functional approaches proved to be unfeasible for the reasons presented in the Introduction. The functions of the R FDA package to estimate monotonic curves failed to converge to a solution because of the peculiar shape of the functions in our problem. This is because the implementation of the FDA package tries to find a set of coefficients for a basis representation while minimizing the approximation error (and preserving the monotonicity constraint), but we were unable to find a sufficiently robust basis representation.

While the absence of applicable methods was the most compelling reason for the development of the method, we can compare the performance of our FAR approach regarding the price prediction in Table 4.

Our functional approach performs slightly worse than univariate forecasting on price alone, at the benefit of being able to use the shape of the curve while crafting the bidding strategy. The slightly worse numerical error is however minimal in the range of the price of this market — between 0 and 82.8 €/MWh. If the goal of the analyst is to produce more accurate price predictions, different λ_1 and λ_2 can be selected to minimize the price prediction error. However, these results need to

Table 4
Mean absolute error (MAE) on the stepwise CV for the last 20 points of data.

Method	MAE
FAR(1) without covariates	0.47
FAR(1) with covariates	0.20
scalar AR(1)	0.14
scalar AR(2)	0.22
scalar ARMA(1,1)	0.19

be interpreted thinking that, while price forecasting may be the central aspect of the auction system presented, our model provides much more insight than the mere analysis of the scalar series of price. Specifically, the operators in the field are energy companies whose interest is not merely to model the market, but to find a way to influence it. The demand and supply curves are built upon the offers of single operators, meaning that knowing the shape of the function can lead to a better understanding of the consequences of the bidding strategy adopted. For example, simulations and what-if studies can be carried out to pick a course of action. Moreover, modeling the whole curve brings useful insights to the analyst, who can interpret the changes and trends in shape with economical expertise. Considering how tested and proven the demand and supply interaction is, we believe that modeling and representing the curves directly fits much better the problem at hand and can lead to a significant advantage in the competitive setting of the market.

5. Discussion

The purpose of this work was to provide the user with additional tools for modeling constrained functional time series. To achieve this, a suitable way to include scalar information in the model was shown, which may include seasonal indicators as well as structural information about the underlying dynamic. The method introduced is computationally light while producing a sizable improvement of the performance in the example provided.

The bootstrap procedure presented performed surprisingly well in augmenting the conclusions about the demand and supply curves from a graphical standpoint. The insights given by the bootstrap trajectories were most of the time useful if not outright critical. The computational burden of the procedure is justified by its ability to provide another way of keeping the model tendency to overfit the data in check. The output is also easy to read and to interpret, making it a useful and accessible tool. A great addition to the method would be an empirical comparison of different block lengths in this functional application, thereby assisting the analyst in picking the best ℓ. Moreover, the bootstrap trajectories may be used to generate a more robust prediction. In fact, it is not uncommon for high-variance and low-bias estimators to benefit from the ensembling of many predictions, similarly to how Random Forests work [5]. However, formalizing this possible improvement must be approached with care considering the sizable amount of bias that the ridge regression introduces, making this topic worth of future investigations.

To make the bootstrap procedure more appealing, a better way of computing the result was presented. This greatly diminished the computational costs of the whole minimization. The matrix form of the optimization problem opens a great variety of possibilities, especially from a computational standpoint. This also leads to a lesser degree of approximation while dealing with the functional data, which translates to better modeling performance.

It is worth stressing that the results discussed in this paper are related to one-step-ahead forecasts only. In theory, as in standard ARIMA models for scalar time series, we can also produce general h-step-ahead forecasts by iterative one-step-ahead forecasts. In doing this, however, our experience shows a general deterioration of the quality of the predictions, as expected.

Acknowledgments

The authors are grateful for useful comments from the Editor-in-Chief, Christian Genest, the Guest Editor and three referees which improved the presentation of the manuscript. This work was initially developed by Jacopo Rossini as his MSc thesis in statistical sciences at University of Padova, Italy. Antonio Canale's work was supported by a SID2017 grant from the University of Padova, Italy. Antonio Canale is also affiliated with the Interdepartmental Centre "Giorgio Levi Cases" for Energy Economics and Technology of the Università di Padova.

Appendix. Additional numerical details

Integration and differentiation. Due to the nature of the transformation (2), it is critical that the numerical differentiation and integration are carried out in a consistent manner. This means that the assumptions used in the numerical differentiation must be compatible with the ones used in the numerical integration to achieve identical curves when transformed back and forth. The numerical integration was carried out using a sum of rectangles, while the numerical differentiation was handled by the `KernSmooth` `R` package [38] as explained in the next paragraph.

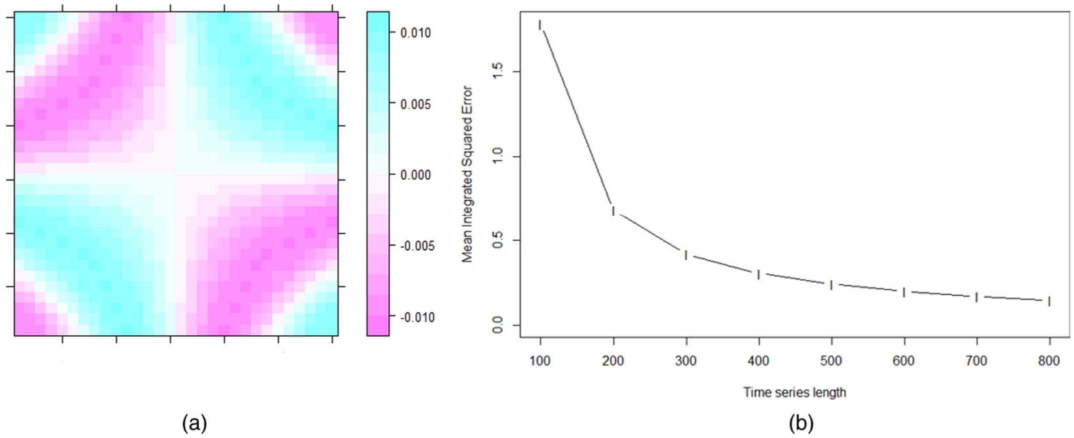

Fig. A.3. True autoregressive operator used in the simulation study (a) and mean integrated squared error of its estimation in function of the functional time series length.

Table A.5
Average point-wise error at different values of grid size z, on the original price scale.

Grid size (z)	500	1500	2500
Average error	0.36	0.10	0.08

Smoothing. The smoothing of the original, step-wise function was done with the help of the `KernSmooth` R package [38]. The choice of the bandwidth (10,000) for the estimation of the smoothed function was done by eye, which leads to satisfactory performance in many situations; see Section sec3.1 in [37]. To estimate the derivatives of the functions, `KernSmooth` was again used. This time, an adaptive bandwidth was used: starting from a baseline of 240, for the initial 15 points of evaluation the bandwidth could increase up to 8 times to guarantee the transformation and back-transformation consistency. The bandwidth chosen was the one used when the error of transformation and back-transformation was deemed acceptable (less that 0.001% of relative error). This guarantees a satisfactory performance even in difficult situations, a feat that could not be achieved with any other differentiation method; see [24].

Consistency of the estimating procedure. A small simulation study was conducted to check the behavior of the proposed estimation procedure. Different functional time series of different length ranging from 100 to 800 were generated following the autoregressive data generating process induced by a simple FAR(1) model with the Hilbert–Schmidt operator reported in the left panel of Fig. A.3 and random functional noise generated from a Gaussian process [35] with zero mean function and squared exponential kernel. For each sample size, we simulated $R = 100$ independent replicated functional time series, and each of them was used in turn as a training set to check the consistency of estimator of the Hilbert–Schmidt autoregressive operator described in Theorem 1. For each sample size, we computed the mean integrated squared error, viz.

$$\frac{1}{R}\sum_{r=1}^{R}\iint\{\psi(s,u)-\hat{\psi}^{(r)}(s,u)\}^2 dsdu,$$

where $\hat{\psi}^{(r)}$ is the estimates of the Hilbert–Schmidt operator ψ obtained in the rth replicated sample. The right panel of Fig. A.3 reports the results of this simulation experiment. As expected, the mean integrated squared error decreases to zero as the sample size increases, thus providing empirical evidence of consistency.

Selecting the grid size for the estimation procedure. In this section we discuss the level of resolution of the finite grid used to evaluate the functions. A broad comparison is feasible, thanks to the computational gains obtained with the application of the results of Section 2.2. Table A.5 reports a measure of distance between the original step-wise function and the smoothed one, for different values of the grid size z. In the application presented in Section 4, we used $z = 1500$, being the best trade-off between speed and acceptable performance.

References

[1] G. Aneiros-Pérez, P. Vieu, Nonparametric time series prediction: A semi-functional partial linear modeling, J. Multivariate Anal. 99 (2008) 834–857.
[2] A. Aue, A. van Delft, Testing for stationarity of functional time series in the frequency domain, ArXiv e-prints, January 2017.
[3] D. Bosq, Linear Processes in Function Spaces: Theory and Applications, Springer, New York, 2010.
[4] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, G.M. Ljung, Time Series Analysis: Forecasting and Control, Wiley, New York, 2015.

[5] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32.
[6] A. Canale, M. Ruggiero, Bayesian nonparametric forecasting of monotonic functional time series, Electron. J. Stat. 10 (2016) 3265–3286.
[7] A. Canale, S. Vantini, Constrained functional time series: Applications to the Italian gas market, Int. J. Forecast. 32 (2016) 1340–1351.
[8] A. Cuevas, A partial overview of the theory of statistics with functional data, J. Statist. Plann. Inference 147 (2014) 1–23.
[9] J. Damon, S. Guillas, The inclusion of exogenous variables in functional autoregressive ozone forecasting, Environmetrics 13 (2002) 759–774.
[10] A. van Delft, P. Bagchi, V. Characiejus, H. Dette, A nonparametric test for stationarity in functional time series, ArXiv e-prints, August 2017.
[11] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, IEEE, 1995, pp. 39–43.
[12] B. Efron, R.J. Tibshirani, An Introduction to the Bootstrap, Chapman & Hall, New York, 1993.
[13] F. Ferraty, P. Vieu, Nonparametric Functional Data Analysis: Theory and Practice, Springer, New York, 2006.
[14] J. Friedman, R.J. Tibshirani, The monotone smoothing of scatterplots, Technometrics 26 (1984) 243–250.
[15] A. Goia, P. Vieu, An introduction to recent advances in high/infinite dimensional statistics, J. Multivariate Anal. 146 (2016) 1–6.
[16] P. Hall, J.L. Horowitz, B.-Y. Jing, On blocking rules for the bootstrap with dependent data, Biometrika 82 (1995) 561–574.
[17] T. Hastie, R.J. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, second ed., Springer, New York, 2009.
[18] N.J. Higham, Accuracy and Stability of Numerical Algorithms, second ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 2002.
[19] L. Horváth, P. Kokoszka, Inference for Functional Data with Applications, Springer, New York, 2012.
[20] A. Inoue, M. Shintani, Bootstrapping GMM estimators for time series, J. Econometrics 133 (2006) 531–555.
[21] G. Kon Kam King, A. Canale, M. Ruggiero, Bayesian functional forecasting with locally-autoregressive dependent processes, Preprint, 2018.
[22] S.N. Lahiri, Resampling Methods for Dependent Data, Springer, New York, 2003.
[23] S.N. Lahiri, K. Furukawa, Y. Lee, A nonparametric plug-in rule for selecting optimal block lengths for block bootstrap methods, Stat. Methodol. 4 (2007) 292–321.
[24] D. Levy, Introduction to Numerical Analysis, Department of Mathematics and Center for Scientific Computation and Mathematical Modeling (CSCAMM) University of Maryland, 2010.
[25] E. Mammen, Estimating a smooth monotone regression function, Ann. Statist. 19 (1991) 724–740.
[26] E. Mammen, J.S. Marron, B.A. Turlach, M.P. Wand, A general projection framework for constrained smoothing, Stat. Sci. 16 (2001) 232–248.
[27] E. Mammen, C. Thomas-Agnan, Smoothing splines and shape restrictions, Scand. J. Stat. 26 (1999) 239–252.
[28] J. Močkus, On Bayesian methods for seeking the extremum, in: Optimization Techniques IFIP Technical Conference, Springer, New York, 1975, pp. 400–404.
[29] E. Passow, J.A. Roulier, Monotone and convex spline interpolation, SIAM J. Numer. Anal. 14 (1977) 904–909.
[30] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2017.
[31] P. Raña, G. Aneiros, J. Vilar, P. Vieu, Bootstrap confidence intervals in functional nonparametric regression under dependence, Electron. J. Stat. 10 (2016) 1973–1999.
[32] J.O. Ramsay, Monotone regression splines in action, Stat. Sci. 3 (1988) 425–441.
[33] J.O. Ramsay, B.W. Silverman, Applied Functional Data Analysis: Methods and Case Studies, Springer, New York, 2002.
[34] J.O. Ramsay, B.W. Silverman, Functional Data Analysis, second ed., Springer, New York, 2005.
[35] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, MIT Press, Cambridge, CA, 2006.
[36] R.J. Tibshirani, Regression shrinkage and selection via the Lasso, J. R. Stat. Soc. Ser. B Stat. Methodol. 58 (1996) 267–288.
[37] M.P. Wand, M.C. Jones, Kernel Smoothing, Chapman & Hall/CRC, New York, 1994.
[38] M.P. Wand, B. Ripley, KernSmooth R package, 2015.
[39] S. Winsberg, J.O. Ramsay, Monotonic transformations to additivity using splines, Biometrika 67 (1980) 669–674.
[40] T. Zhu, D.N. Politis, Kernel estimates of nonparametric functional autoregression models and their bootstrap approximation, Electron. J. Stat. 11 (2017) 2876–2906.
[41] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. R. Stat. Soc. Ser. B Stat. Methodol. 67 (2005) 301–320.