Ca' Foscari University of Venice

Department of Economics

Working Paper

Dominique Guégan
Matteo Iacopini

Nonparametric Forecasting of Multivariate Probability Density Functions

ISSN: 1827-3580
No. 15/WP/2018

Working Papers
Department of Economics
Ca' Foscari University of Venice
No. 15/WP/2018
ISSN 1827-3580

Nonparametric Forecasting of Multivariate Probability Density Functions

Dominique Guégan
Université Paris I – Panthéon-Sorbonne
Ca' Foscari University of Venice

Matteo Iacopini
Ca' Foscari University of Venice
Université Paris I – Panthéon-Sorbonne

24 May 2018

Abstract
The study of dependence between random variables is the core of theoretical and applied statistics. Static and dynamic copula models are useful for describing the dependence structure, which is fully encrypted in the copula probability density function. However, these models are not always able to describe the temporal change of the dependence patterns, which is a key characteristic of financial data.
We propose a novel nonparametric framework for modelling a time series of copula probability density functions, which allows to forecast the entire function without the need of post-processing procedures to grant positiveness and unit integral. We exploit a suitable isometry that allows to transfer the analysis in a subset of the space of square integrable functions, where we build on nonparametric functional data analysis techniques to perform the analysis.
The framework does not assume the densities to belong to any parametric family and it can be successfully applied also to general multivariate probability density functions with bounded or unbounded support. Finally, a noteworthy field of application pertains the study of time varying networks represented through vine copula models.
We apply the proposed methodology for estimating and forecasting the time varying dependence structure between the S&P500 and NASDAQ indices.

Keywords
Functional data analysis, functional PCA, functional time series, time varying dependence, time varying copula, clr transform, compositional data analysis

AMS 2000 subject classifications: Primary 62; secondary 91B84
JEL Codes: C13, C33, C51, C53

Address for correspondence:
Matteo Iacopini
Department of Economics
Ca' Foscari University of Venice
Cannaregio 873, Fondamenta S. Giobbe
30121 Venezia - Italy
e-mail: matteo.iacopini@unive.it

This Working Paper is published under the auspices of the Department of Economics of the Ca' Foscari University of Venice. Opinions expressed herein are those of the authors and not those of the Department. The Working Paper series is designed to divulge preliminary or incomplete work, circulated to favour discussion and comments. Citation of this paper should consider its provisional character

The Working Paper Series
is available only on line
(http://www.unive.it/nqcontent.cfm?a_id=86302)
For editorial correspondence, please contact:
wp.dse@unive.it

Department of Economics
Ca' Foscari University of Venice
Cannaregio 873, Fondamenta San Giobbe
30121 Venice Italy
Fax: ++39 041 2349210

Nonparametric forecasting of multivariate probability density functions

Dominique Guégan*[1,2,3,4], Matteo Iacopini†[1,2,3,4]

[1]Université Paris I - Panthéon-Sorbonne
[2]Centre d'Économie de la Sorbonne (CES)
[3]LabEx ReFi (Financial Regulation), Paris
[4]Ca' Foscari University of Venice

June 4, 2018

Abstract

The study of dependence between random variables is the core of theoretical and applied statistics. Static and dynamic copula models are useful for describing the dependence structure, which is fully encrypted in the copula probability density function. However, these models are not always able to describe the temporal change of the dependence patterns, which is a key characteristic of financial data.

We propose a novel nonparametric framework for modelling a time series of copula probability density functions, which allows to forecast the entire function without the need of post-processing procedures to grant positiveness and unit integral. We exploit a suitable isometry that allows to transfer the analysis in a subset of the space of square integrable functions, where we build on nonparametric functional data analysis techniques to perform the analysis.

The framework does not assume the densities to belong to any parametric family and it can be successfully applied also to general multivariate probability density functions with bounded or unbounded support. Finally, a noteworthy field of application pertains the study of time varying networks represented through vine copula models.

We apply the proposed methodology for estimating and forecasting the time varying dependence structure between the S&P500 and NASDAQ indices.

Keywords: Functional data analysis, functional PCA, functional time series, time varying dependence, time varying copula, clr transform, compositional data analysis.

AMS 2000 subject classifications: Primary 62; secondary 91B84.

JEL Classification: C13, C33, C51, C53

*e-mail: dominique.guegan@univ-paris1.fr

†e-mail: matteo.iacopini@unive.it, corresponding author.

1 Introduction

One of the most relevant research fields in theoretical and applied statistics is devoted to the study of the dependence between random variables. In finance, the analysis of the dependence patterns is a challenging problem and its understanding serves several purposes: control of risk clustering, credit, market and systemic risk measurement, pricing and hedging of credit sensitive instruments (such as collateralized debt obligations or CDOs) and credit portfolio management. The analysis of the relationships between economic and financial variables is crucial for the identification of causality relations (e.g., see Granger (1988), White and Lu (2010)). From a statistical perspective, the main purpose is the development of models able to describe and forecast the joint dynamic behaviour of financial variables. Moreover, these models may provide an effective support for the financial regulator (for example, in predicting and counteracting an increase of the systemic risk).

Firstly developed by Sklar (1959), copula functions have attracted signifcant attention over the last decade, particularly within the financial and econometric communities, as a flexible instrument for modelling the joint distribution of random variables (see Joe (1997), Nelsen (2013) and Durante and Sempi (2015) for an introduction and a compelling review). Let $(X_1, \ldots, X_d)$ be a random vector with continuous marginal cumulative distribution functions (cdf) $F_i(\cdot)$ and probability density function (pdf) $f_i(\cdot)$. The random vector $(U_1, \ldots, U_d) = (F_1(X_1), \ldots, F_d(X_d))$, obtained by application of the probability integral transform, has uniform marginals. The copula of $(X_1, \ldots, X_d)$ is defined as the joint cumulative distribution function $C : [0,1]^d \to [0,1]$ of $(U_1, \ldots, U_d)$, that is $C(u_1, \ldots, u_d) = \mathbb{P}(U_1 \leq u_1, \ldots, U_d \leq u_d)$. Moreover, denoting $F(\cdot)$ the joint cumulative distribution of $(X_1, \ldots, X_d)$ and by $f(\cdot)$ its probability density function, Sklar's theorem (Sklar (1959)) states that there exists a unique copula $C(\cdot)$ with probability density function $c : [0,1]^d \to \mathbb{R}_+$ such that $F(x_1, \ldots, x_d) = C(F_1(x_1), \ldots, F_d(x_d))$ and $f(x_1, \ldots, x_d) = c(F_1(x_1), \ldots, F_d(x_d)) \prod_{i=1}^d f_i(x_i)$.

The use of a copulas permits to separately deal with the marginal distributions and the dependence structure among a set of random variables, thus providing a high degree of flexibility in modelling the corresponding joint distribution. The literature on quantitative finance and financial econometrics has widely recognized the importance of this instrument, as documented by the review of Patton (2012) and the textbooks by Cherubini et al. (2004) and Cherubini et al. (2011).

A standard practice in financial econometrics, motivated by the fact that multivariate data (e.g., returns of a portfolio of assets) have non-Gaussian marginal distributon, consists in assuming a heavy-tailed distribution for the marginals (or to estimate them nonparametrically) and to join them with a parametric copula function, which fully describes the dependence structure (Deheuvels (1978), Deheuvels (1979)) through its parameters. This approach allows a parsimonious description of the dependence between two variables by means of the few parameters of a copula function.

This method has some undesirable shortcomings. First, each parametric copula family is designed to describe only a specific dependence pattern (for example, see Table 1), which makes the selection of the family a crucial aspect of every modelling strategy. To this aim, we recall the definition of upper (lower) tail dependence from Cherubini et al. (2004). This concept is used to describe situations where high (low) values of a variables are likely to be observed together with high (low) values of the other. In terms of the copula pdf, this means that the probability is concentrated to the top-right (bottom-left) corner. Formally, given two random variables $X \sim G_X$ and $Y \sim G_Y$ with bivariate copula $C(\cdot)$, the upper and lower tail dependence

Copula family	**Upper** λ_U	**Lower** λ_L
Gaussian(ρ)	1 if $\rho = 1$	1 if $\rho = 1$
t-student(ν, ρ)	$\lambda > 0$ if $\rho > 1$	$\lambda > 0$ if $\rho > 1$
Gumbel(θ)	$2 - 2^{1/\theta}$	0
Clayton(θ)	0	$2^{-1/\theta}$
Frank(θ)	0	0
Fréchet(p, q)	q	q

Table 1: Upper and lower tail dependence for some copula families. In brackets the parameters of the copula family.

coefficients (upper TDC and lower TDC, respectively) are given by:

$$\lambda_U = \lim_{u \to 1^-} \mathbb{P}(G_X(X) > u | G_Y(Y) > u) = \lim_{u \to 1^-} \frac{1 - 2u + C(u,u)}{1-u}, \tag{1}$$

$$\lambda_L = \lim_{u \to 0^+} \mathbb{P}(G_X(X) < u | G_Y(Y) < u) = \lim_{u \to 0^+} \frac{C(u,u)}{u}, \tag{2}$$

which are asymptotically equivalent to:

$$\lambda_U = 2 - \lim_{u \to 1^-} \frac{\log(C(u,u))}{\log(u)}, \qquad \lambda_L = 2 - \lim_{u \to 0^+} \frac{\log(1 - 2u + C(u,u))}{\log(1-u)}. \tag{3}$$

The copula $C(\cdot)$ is said to have upper (lower) tail dependence when $\lambda_U \neq 0$ ($\lambda_L \neq 0$). Table 1 reports the tail dependence coefficients for some commonly used copula families (see Cherubini et al. (2004)). Only some of them (e.g., Gaussian, t-student and Fréchet, for some values of their parameters) have simultaneously upper and lower tail dependence: this happens only for some values of the parameter of the copula and, in any case, the tail dependence coefficients are equal, thus implying that the tail dependence is symmetric.

Second, when the copula parameter is assumed to be fixed, these constructions are able to identify only the overall, static relations and fail to account for any kind of spatial or temporal change. This constraint is particularly restrictive in time series analysis of financial data, as pointed out by Fermanian and Scaillet (2004). In fact, very often the relations between financial variables are non linear and change over time.

To address this shortcoming, Patton (2006a) and Patton (2006b) introduced dynamic copula models by assuming that the parameters of the copula function are driven by an autoregressive process. Instead, Fermanian and Wegkamp (2012) allowed the parameters to depend on past realizations of the observables. These seminal works, have opened a new avenue to research (see Manner and Reznikova (2012) for a review) and has brought outstanding improvements to the econometrician's toolbox. For example, So and Yeung (2014) and Jondeau and Rockinger (2006) incorporate dynamics into a copula-GARCH model improving its forecasting performance, Dias and Embrechts (2004) and Van Den Goorbergh et al. (2005) exploited dynamic copulas in modelling high-frequency data and option pricing, respectively, whereas Oh and Patton (2017) has recently applied this methodology to the study of systemic risk. Other relevant empirical contributions exploiting this construction include Almeida et al. (2016), Bartram et al. (2007), Weiß and Supper (2013), Hu (2010), Hafner and Reznikova (2010), Hafner and Manner (2012), Guidolin and Timmermann (2006).

Despite the greater flexibility, dynamic copulas may fail to account for the characteristics of the dependence structure among financial data. Since each copula family is constructed for

describing a specific dependence pattern, the change of its parameters may not be enough to capture other types of dependence.

The recent paper by Guégan and Zhang (2010) found empirical evidence supporting this theory. They developed a strategy for testing the null of a static copula against a dynamic copula, then upon rejection they tested for the change of the copula family over different temporal windows. The main results is that the dependence structure between the S&P500 and NASDAQ indices experienced a great variability over time, thus stressing the need for a dynamic model; nonetheless the null hypothesis of equal copula function family was rejected for several windows. This suggests that, in this dataset, the evolution of the dynamic copula parameter is insufficient to account for the variation of the dependence structure and different copula families should be used for different temporal windows.

To overcome these limitations, we propose a methodology that has the whole function as the object of interest, instead of a finite-dimensional parameter vector. We do this by exploiting some results developed in the literature on functional data analysis, which we briefly introduce in the following.

Starting from the seminal work of Bosq (2000), functional data analysis (see Ramsay and Silverman (2005) and Ferraty and Vieu (2006) for a thorough treatment) has found applications also in finance and financial econometrics (see Hörmann and Kokoszka (2012), Kokoszka (2012) and Kidziński (2015) for an introduction to the topic). Different models have been proposed for time series of functional data: for example, Sen and Klüppelberg (2015) assumed stationarity and estimated a VARMA functional process for electricity spot prices, conversely Liebl (2010) used the same data but proposed a method for dealing with non-stationary data and applied it in Liebl (2013) for forecasting. Within the same stream of research, Horváth et al. (2010) and Horváth et al. (2014) designed a testing procedure for detecting non-stationarity. Aue et al. (2015) and Kargin and Onatski (2008), instead, used time series functional for improving on the forecasting performance of multivariate on forecasting. More recently, Kidziński et al. (2016) and Klepsch et al. (2017) extended the theory of univariate ARMA models to the functional framework, by introducing also seasonal effects. Finally, Petris (2013) and Canale and Ruggiero (2016) developed an inferential procedure following the Bayesian paradigm, following a parametric and non-parametric approach, respectively.

The literature on functional time series modelling can be partitioned into two main classes, according to the methodology developed. The parametric framework, firstly introduced by Bosq (2000), hinges on the linear functional autoregressive model (FAR) which can be considered an infinite-dimensional analogue of vector autoregressive (VAR) processes, widely used in times series analysis. By contrast, the non-parametric approach (see Ferraty and Vieu (2006) and Ramsay and Silverman (2005) for an overview) relies on functional principal component analysis (fPCA). See Appendix A for a short introduction.

Unfortunately, none of the previously mentioned approaches is suited for dealing with constrained functions, such as probability density functions (pdfs), which must be positive (on their support) and have unit integral. A statistical model for the analysis of a time series of pdfs should include a mechanism for dealing with these constraints. In the literature, three main approaches have been proposed to address this issue. One possibility consists in ignoring the constraints and treating the pdfs as an unrestricted functions, then after the estimation and forecasting steps, the output is re-normalized in order to get a probability density function. Sen and Ma (2015) adopted this approach for studying a time series of pdfs of financial data from the S&P500 and the Bombay Stock Exchange.

More appealing alternatives do not need to post-process the output and allow to perform the analysis taking into account the constraints. The seminal works by Egozcue et al. (2006), van der Boogaart et al. (2010), van der Boogaart et al. (2014) and Egozcue and Pawlowsky-Glahn (2015)

introduced the notion of Bayes space, that is a Hilbert space of probability density functions. They borrowed from compositional data analysis and Aitchison's geometry (see Aitchison (1986)) and interpreted probability density functions as infinite-dimensional compositional vectors. They replaced the pointwise sum and multiplication by the operations of perturbation and powering which, for $f(\cdot), g(\cdot) \in \mathbb{D}(I)$, $I \subset \mathbb{R}^n$, and $\alpha \in \mathbb{R}$, are defined by:

$$f(\mathbf{x}) \oplus g(\mathbf{x}) = \frac{f(\mathbf{x})g(\mathbf{x})}{\int_I f(\mathbf{x})g(\mathbf{x})\,\mu(\mathrm{d}\mathbf{x})}\,, \qquad \alpha \odot f(\mathbf{x}) = \frac{f(\mathbf{x})^\alpha}{\int_I f(\mathbf{x})^\alpha\,\mu(\mathrm{d}\mathbf{x})}\,. \tag{4}$$

Instead, the analogue of subtraction is given by $f(\cdot) \ominus g(\cdot) = f(\cdot) \oplus [-1 \odot g(\cdot)]$. They showed that the tuple $(\mathbb{D}(I), \oplus, \odot)$ is a space and that the subset $\mathbb{D}^*(I) \subset \mathbb{D}(I)$ of probability density functions whose logarithm is square integrable is a Hilbert space. These remarkable results permit to conduct the analysis directly in $\mathbb{D}^*(I)$, provided that it is possible to re-define the necessary statistical models by means of the new operations $\oplus, \odot$. van der Boogaart et al. (2014) proved that $\mathbb{D}^*(I)$ is isomorphic to the space $\mathcal{L}_2^*(I)$ of functions on I with square integrable logarithm (written $\mathbb{D}^*(I) \cong_{\mathrm{clr}} \mathcal{L}_2^*(I)$) via the centred log-ratio isometry defined as follows (see Section 2.1 for the notation).

Definition 1.1 (Centred log-ratio)
Let ν be a measure on $\mathbb{R}^n$ and $f : I \to \mathbb{R}_+$ be a probability density function supported on a set $I \subset \mathbb{R}^n$ of finite ν-measure, that is $\nu(I) < \infty$ and $\nu(I) \neq 0$. The centred log-ratio (clr) transformation is an invertible map defined as:

$$\mathrm{clr}(f)(\mathbf{x}) = g(\mathbf{x}) = \log(f)(\mathbf{x}) - \frac{1}{\nu(I)} \int_I \log(f)(\mathbf{y})\,\nu(\mathrm{d}\mathbf{y})\,, \tag{5}$$

with inverse given by:

$$\mathrm{clr}^{-1}(g)(\mathbf{x}) = f(\mathbf{x}) = \frac{\exp(g)(\mathbf{x})}{\int_I \exp(g)(\mathbf{y})\,\nu(\mathrm{d}\mathbf{y})}\,. \tag{6}$$

Consequently, by definition 1.1 it follows that the clr transform of a pdf supported on I has to satisfy the following constraint (which we will call zero integral constraint in the rest of this paper):

$$\int_I \mathrm{clr}(f)(\mathbf{x})\,\mu(\mathrm{d}\mathbf{x}) = \int_I \log(f)(\mathbf{x})\,\mu(\mathrm{d}\mathbf{x}) - \int_I \frac{1}{\mu(I)} \left[\int_I \log(f)(\mathbf{y})\,\mu(\mathrm{d}\mathbf{y})\right] \mu(\mathrm{d}\mathbf{x}) = 0\,. \tag{7}$$

The spaces $\mathbb{D}(I), \mathbb{D}^*(I), \mathcal{L}_1(I), \mathcal{L}_2(I), \mathcal{L}_2^*(I)$ are defined with respect to a reference measure ν and contain equivalence classes of functions which are proportional each other, that is we implicitly defined $\mathbb{D}(I) = \mathbb{D}_\nu(I)$, $\mathbb{D}^*(I) = \mathbb{D}_\nu^*(I)$, $\mathcal{L}_{1,\nu}(I) = \mathcal{L}_{1,\nu}(I)$, $\mathcal{L}_2(I) = \mathcal{L}_{2,\nu}(I)$, $\mathcal{L}_2^*(I) = \mathcal{L}_{2,\nu}^*(I)$. In this paper we consider to be the Lebesgue reference measure, i.e. $\nu = \mu$. In order to single out a specific element it is necessary to normalize the reference measure. This can be easily done if the set I is ν-finite, whereas if $\nu(I) = \infty$, normalization can be done using the centring procedure (see van der Boogaart et al. (2014)). Moreover, the following relations hold: $\mathbb{D}^*(I) \subset \mathbb{D}(I) \subset \mathcal{L}_1(I)$ and $\mathbb{D}^*(I) \cong_{\mathrm{clr}} \mathcal{L}_2^*(I) \subset \mathcal{L}_2(I) \subset \mathcal{L}_1(I)$.

It is possible to use the clr transform to project a pdf into the Hilbert space $\mathcal{L}_2^*(I)$ (provided that its logarithm is square integrable), which is a space embedded with the operations of pointwise addition and multiplication. We can perform the statistical analysis in this space and then project the output back into $\mathbb{D}^*(I)$ by the inverse map $\mathrm{clr}^{-1}(\cdot)$. This strategy via the centred log-ratio map has been proposed by Hron et al. (2016) for performing fPCA on univariate pdfs with compact support. Canale and Vantini (2016) developed a different isometric, bijective function which maps constrained functions into a pre-Hilbert space, then estimated a FAR

model on this space and transformed back the result into the original space. Though general, this framework is not explicitly designed for dealing with pdfs, but only bounded and monotonic functions, thus preventing from its use in the current setting. The same idea of transforming pdfs into $\mathcal{L}_2(I)$ via an invertible map has been followed by Petersen and Müller (2016), who defined two different transformations satisfying this property: the log hazard and the log quantile density transformations, respectively. Despite their strong theoretical properties, both maps have the shortcoming of not having an equivalent transformation applicable in the multivariate case, which makes them unsuited for the analysis of multivariate probability density functions.

The empirical finding by Guégan and Zhang (2010) represents the key stylized fact motivating our work. Given that a dynamic copula model may not be sufficiently flexible to describe the time varying dependence between financial variables, we contribute to this active field of research by proposing a different statistical framework for forecasting multivariate probability density functions. To address the issues related with modelling pdfs, we extend the procedure of Hron et al. (2016) who build on the previous work by van der Boogaart et al. (2010, 2014). The idea is to map the space of probability density functions to the space of integrable functions through an isometry, perform the analysis in this space (which has nicer properties), then use the inverse mapping to get the solution in the original space. Our contribution is also related to the studies of Liebl (2013) and Hays et al. (2012), who developed dynamic models for forecasting functional time series of electricity prices on the basis of fPCA. However, our focus is on the modelling of probability density functions, which call for the adoption of more complex tools than that of unrestricted functions. Finally, we contribute to the literature on dynamic dependence modelling in finance by providing a tool able to forecast the temporal evolution of the dependence pattern between the S&P500 and NASDAQ indices.

We propose a nonparametric framework for forecasting multivariate probability density functions by extending existing models for the analysis of cross sectional, univariate probability density functions with compact support. We focus on bivariate copula pdfs because of their great importance in finance, however the proposed methodology is flexible and general, thus permitting to deal with the prediction of general pdfs with bounded or, under some conditions, unbounded support. Thanks to the fact that a copula pdf encompasses all information on the dependence structure, we can interpret our approach as a general framework for modelling the (temporally evolving) dependence patterns between random variables.

The reminder of the paper is as follows. In Section 2 we introduce the notation as well as the fundamental concepts that will be used throughout the paper. Section 3 presents the details of the proposed baseline methodology, whereas Section 4 provides insights on potential issues and extensions. Section 6 provides an overview of the financial dataset used and presents the results of the empirical analysis. Finally, Section 7 concludes and describes some extensions of the current work and lines of future research.

2 Preliminaries

In this section we describe the proposed methodology after having introduced the main notation that will be used throughout the rest of the paper.

2.1 Notation

Throughout the paper, if not differently specified, greek letters denote unknown quantities to be estimated, whereas latin letters any other variable. We denote scalars with lower-case letters, vectors with boldface lower-case letters and matrices with boldface upper-case letters. We use the shorthand $f(\cdot)$ for denoting a function, regardless of the number of arguments it takes,

moreover we denote the composition of functions by $(g \circ f)(\cdot) = g(f(\cdot)) = g(f)(\cdot)$. The inner product between two functions $f(\cdot), g(\cdot)$ supported on $I \subseteq \mathbb{R}^n$ is defined in the standard way by $\langle f(\cdot), g(\cdot)\rangle = \int_I f(\mathbf{x})g(\mathbf{x})\,\mathrm{d}\mathbf{x}$. The integer part of the scalar x is denoted $\lfloor x \rfloor$.

We use the notation $\mathbf{A} = [\mathbf{A}_1, \ldots, \mathbf{A}_T]$ to denote a collection of T matrices $\mathbf{A}_t$ of equal size $N \times M$. The symbol $\mathbf{I}_k$ is used for the identity matrix of size $k \times k$, whereas $\mathbf{0}_k$ for the $k \times 1$ column vector of zeros. Moreover, empty spaces in the matrices stand for zero entries. Let $\mathbf{L}_N$ be the matrix representation of the first difference operator L, that is the $N \times (N+1)$ matrix which post-multiplied by a $(N+1)$-dimensional vector $\mathbf{a}$ yields a vector of size N, $\mathbf{La}$, whose entries are the first differences of the elements of $\mathbf{a}$. The Moore-Penrose pseudo-inverse of the $N \times M$ matrix $\mathbf{A}$ is denoted by $\mathbf{A}^\dagger$. If $\mathbf{A}$ is positive definite, we define by $\mathbf{A}^{1/2}$ its (unique) principal square root.

In Section 3 we will refer to the spaces of functions described as follows. We define $\mathbb{F}_+(I)$ to be the space of non-negative, integrable functions on $I \subseteq \mathbb{R}^n$, whose general element is the function $f : I \to \mathbb{R}_+$, and we let $\mathbb{F}_0(I)$ be the space of functions on I with zero integral. $\mathbb{D}(I)$ denotes the set of probability density functions with support I and we define $\mathbb{D}^*(I)$ to be the space of probability density functions with support I whose logarithm is square integrable. We denote by $\mu(\cdot)$ the Lebesgue measure on $\mathbb{R}^n$, for $n \geq 1$. In the case $n = 1$ we also use the shorthand notation $\mathrm{d}x = \mu(\mathrm{d}x)$, whereas for $n > 1$ we define $\mathrm{d}\mathbf{x} = \mu(\mathrm{d}\mathbf{x})$. Consequently, if $I = [a, b]$ then $\mu(I) = b - a$. All integrability definitions are made using the Lebesgue measure as reference measure, if not differently specified. Let $\mathcal{L}_p(I)$ be the space of p-integrable functions supported on I and let $\mathcal{L}_p^*(I)$ be the space of functions on I whose logarithm is p-integrable. The n-dimensional unit simplex is defined as $\mathcal{S}^n = \{\mathbf{x} \in \mathbb{R}^n : x_i > 0, i = 1, \ldots, n, \text{ and } \sum_{i=1}^n x_i = 1\}$, whereas $\mathcal{S}_0 = \{\mathbf{x} \in \mathbb{R}^n : \sum_{i=1}^n x_i = 0\}$ is the subspace of n-dimensional vectors whose elements have zero sum. We define $\{\mathbf{e}_1, \ldots, \mathbf{e}_N\}$ be the canonical basis of the space of $N \times N$ matrices. For two spaces X, Y we use the notation $X \cong_f Y$ to indicate that they are isomorphic through the isometric isomorphism $f : X \to Y$.

Let $\mathbf{x} = (x_1, \ldots, x_n)'$ be a vector of observations. In Section 3.1 we denote the empirical (marginal) cumulative distribution function of $\mathbf{x}$ by $F^n(\mathbf{x}) = (F^n(x_1), \ldots, F^n(x_n))'$. Moreover, define the rank transformation of $\mathbf{x}$ to be the function that maps each element x_i of $\mathbf{x}$ to:

$$R_i = \sum_{j=1}^{n} \mathbb{1}(x_j \leq x_i)\,. \tag{8}$$

Denote with $\mathbf{u} = (u_1, \ldots, u_n)'$ with $u_i \in [0, 1]$, $i = 1, \ldots, n$, the vector of pseudo-observations associated to the observations $\mathbf{x}$, used for the estimation of the empirical copula in Section 3.1. Each pseudo-observation is defined as:

$$u_i = \frac{1}{n} R_i\,. \tag{9}$$

In Section 3.2 and Appendix A, we will use the following notation in performing functional principal component analysis (fPCA). We define a sample of observed functions by $\mathbf{f}(\cdot) = (f_1(\cdot), \ldots, f_T(\cdot))'$, with $f_t : I \to \mathbb{R}$, for some domain $I \subseteq \mathbb{R}^n$. Moreover, we let $\breve{\mathbf{f}}(\cdot) = (\breve{f}_1(\cdot), \ldots, \breve{f}_T(\cdot))'$ denote the approximation of the observed functions obtained as an outcome of the fPCA. The principal component functions are denoted by $\boldsymbol{\xi}(\cdot) = (\xi_1(\cdot), \ldots, \xi_J(\cdot))'$ and the scores associated to the function $\breve{f}_t(\cdot)$ are $\boldsymbol{\beta}_t = (\beta_{t,1}, \ldots, \beta_{t,J})'$, moreover let $\mathbf{B} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_T)$. The corresponding estimated quantities are denoted by $\widehat{\boldsymbol{\xi}}(\cdot)$, $\widehat{\boldsymbol{\beta}}_t$ and $\widehat{\mathbf{B}}$.

In Section 3.2 we also use spline functions, for which we adopt the following notation. We denote by $\mathfrak{I}_n$ a n-dimensional index set whose elements are the n-tuple $(i_1, \ldots, i_n)$ with entries

$i_j \in [1, I_j]$, where I_j is a positive integer for each $j = 1, \ldots, n$. We denote with $\boldsymbol{\lambda}$ the vector with entries $\lambda_0 < \ldots < \lambda_{g+1}$ representing the points of the knot sequence used for the spline functions. For a m-order spline, define the extended knot sequence as the vector $\bar{\boldsymbol{\lambda}}$ of length $2m + g + 2$ whose entries satisfy the relations:

$$\underbrace{\lambda_{-m} = \ldots = \lambda_{-1}}_{m} = \underbrace{\lambda_0 < \ldots < \lambda_{g+1}}_{g+2} = \underbrace{\lambda_{g+1+1} = \ldots = \lambda_{g+m+1}}_{m} .$$

An extended knot sequence for a bivariate spline is defined by $\bar{\boldsymbol{\lambda}}^{x,y} = \bar{\boldsymbol{\lambda}}^{x} \otimes \bar{\boldsymbol{\lambda}}^{y}$, where $\bar{\boldsymbol{\lambda}}^{x}, \bar{\boldsymbol{\lambda}}^{y}$ are the extended knot sequences along each axis and the generic entry is the couple $\bar{\lambda}^{x,y}_{i,j} = (\bar{\lambda}^{x}_{i}, \bar{\lambda}^{y}_{j})$. $B^k_i(x)$ denotes the univariate (basis) B-spline function of degree $m-1$, with knot sequence indexed by i and $D^{\ell}_x[f](x)$, $\ell \leq m-1$, is the partial derivative operator of order ℓ, applied to the function f with respect to the variable x. For univariate splines of degree m, let $\mathbf{b} = (b_{-m}, \ldots, b_g)'$ be the $(g+m+1)$ coefficient vector, whereas for bivariate splines of the same degree we define the $(m+g+1) \times (m+g+1)$ coefficient matrix by $\bar{\mathbf{B}}$. Moreover, let $\mathbf{C}^{m+1}(\mathbf{x}^n)$ be the $n \times (g+m+1)$ collocation matrix of B-spline functions evaluated at the observation points $\mathbf{x}^n = (x_1, \ldots, x_n)'$:

$$\mathbf{C}^{m+1}(\mathbf{x}^n) = \begin{bmatrix} B^{m+1}_{-m}(x_1) & \ldots & B^{m+1}_{g}(x_1) \\ \vdots & \ddots & \vdots \\ B^{m+1}_{-m}(x_n) & \ldots & B^{m+1}_{g}(x_n) \end{bmatrix} \tag{10}$$

Following (De Boor, 2001, ch.10), a univariate spline function of degree k and the corresponding partial derivative of order ℓ are given by:

$$s_m(x) = \sum_{i=-m}^{g} b_i B^{m+1}_i(x) , \tag{11}$$

$$D^{\ell}_x[s_m](x) = s^{(\ell)}_m(x) = \sum_{i=-m}^{g} b^{\ell}_i B^{m}_i(x) , \tag{12}$$

Given an extended knot sequence $\bar{\boldsymbol{\lambda}}$ and evaluation points $\mathbf{x}^n = (x_1, \ldots, x_n)'$, are given by:

$$s_k(\mathbf{x}^n) = \sum_{i=-k}^{g} b_i B^{k+1}_i(\mathbf{x}^n) = \mathbf{C}^{k+1}(\mathbf{x}^n)\mathbf{b} , \tag{13}$$

$$D^{\ell}_x[s_m](\mathbf{x}^n) = s^{(\ell)}_k(\mathbf{x}^n) = \mathbf{C}^{m+1-\ell}(\mathbf{x}^n)\mathbf{b}^{(\ell)} = \mathbf{C}^{m+1-\ell}(\mathbf{x}^n)\mathbf{S}_{\ell}\mathbf{b} , \tag{14}$$

where $\mathbf{C}^{m+1}(\mathbf{x})$ is a matrix of B-splines evaluated at the points $\mathbf{x}$, $\mathbf{b}$ is the coefficient vector and $\mathbf{S}_{\ell}$ is a matrix transforming the coefficient vectors of splines of degrees m to those of their derivatives of degree ℓ. The direct link between the coefficients of a spline and its derivative are due to the property that the derivative of a spline is another spline of lower degree (see De Boor (2001)), that is:

$$s_{m+1}(x) = \int s_m(x) \, \mathrm{d}x . \tag{15}$$

Similarly, we define the d-dimensional tensor product spline function by the tensor product between univariate splines (see Schumaker (2007)):

$$s_m(x_1, \ldots, x_d) = \sum_{i_1} \cdots \sum_{i_d} b_{i_1,\ldots,i_d} B^m_{i_1}(x_1) \cdots B^m_{i_d}(x_d) = \sum_{i_1,\ldots,i_d} b_{i_1,\ldots,i_d} B^m_{i_1}(x_1) \cdots B^m_{i_d}(x_d) , \tag{16}$$

with $b_{i_1,\ldots,i_d} \in \mathbb{R}$ for $(i_1,\ldots,i_d) \in \mathfrak{I}_d$. Notice that the coefficients $b_{i_1,\ldots,i_d}$, with $(i_1,\ldots,i_d) \in \mathfrak{I}_d$, can be represented as a vector of length $\prod_{j=1}^{d} I_j$ or, equivalently, as a d-order array (or tensor) with dimensions $I_1 \times \ldots \times I_d$. The partial derivatives of the multivariate spline in eq. (16) are given by Schumaker (2007) (and can be easily computed via Algorithm 5.11 in Schumaker (2007)):

$$D_{x_1}^{\ell_1} \cdots D_{x_d}^{\ell_d}[s_k](x_1,\ldots,x_d) = \sum_{i_1,\ldots,i_d} b_{i_1,\ldots,i_d}^{\ell_1,\ldots,\ell_d} B_{i_1}^{m-\ell_1}(x_1) \cdots B_{i_d}^{m-\ell_d}(x_d)\,. \tag{17}$$

Finally, in Section 3.3 we use the notation $\widetilde{\mathbf{B}} = (\widetilde{\boldsymbol{\beta}}'_{T+1},\ldots,\widetilde{\boldsymbol{\beta}}_{T+H})'$, where $\widetilde{\boldsymbol{\beta}}_{T+h}$ is the forecast for the vector $\widehat{\boldsymbol{\beta}}_T$ at horizon $h = 1,\ldots,H$. The corresponding forecast for the fPCA approximate functions are denoted by $\widetilde{\mathbf{f}}_{T+H}(\cdot) = (\widetilde{f}_{T+1}(\cdot),\ldots,\widetilde{f}_{T+H}(\cdot))'$ whereas $\widetilde{\mathbf{c}}_{T+H}(\cdot) = (\widetilde{c}_{T+1}(\cdot),\ldots,\widetilde{c}_{T+H}(\cdot))'$ denotes the forecast of the copula probability density functions.

2.2 Related literature

Given an observed bivariate time series of relevant economic or financial variables $(\mathbf{x},\mathbf{y}) = \{x_{t,i}, y_{t,i})\}_{ti}$, with $i = 1,\ldots,N$ for each $t = 1,\ldots,T$, with unknown time varying copula probability density function $c_t(\cdot)$, the purpose of this methodology is to obtain a h-step ahead forecast the copula pdf $\widetilde{c}_{T+h}(\cdot)$, $h = 1,\ldots,H$. In order to achieve this result, we will borrow some concepts from various streams of literature. The core of the methodology is grounded on functional data analysis (FDA), in particular the technique of functional principal component analysis (fPCA), and on the centred log-ratio isometry between $\mathbb{D}^*(I)$ and $\mathcal{L}_2^*(I)$. Furthermore, we exploit several concepts from the literature on nonparametric estimation of copula functions and, finally, we use standard techniques for multivariate time series analysis.

The use of functional autoregressive processes (FAR) proposed by Bosq (2000) is prevented by the constraints holding on pdfs and the fact that $\mathbb{D}(I)$ is not closed under pointwise addition and multiplication. In situations like this, post-processing techniques are necessary for mapping the output of a given procedure into the desired space. However, this procedure is suboptimal as there are guarantees that all the information is preserved by this mapping. By contrast, an efficient forecasting model for probability density functions should yield a consistent output, that is the predicted function must be pdfs.

In second instance, as functions are infinite-dimensional objects, the original forecasting problem would require to work with infinite-dimensional spaces. Though natural, this brings in a significant degree of complexities that a similar problem in finite-dimensional spaces (i.e., Euclidean spaces). Clearly, a direct matching between an infinite-dimensional problem and a finite-dimensional one, does not exist. Moreover, naïve techniques for moving from an infinite-dimensional problem into one a finite one via discretization of the functions could be too rough and lose too much information. Nonetheless, under suitable assumptions it is possible to approximate the infinite-dimensional functional forecasting problem by a simpler one in which the parameters of interest are finite-dimensional vectors. Moreover, under certain conditions this approximation is optimal (according to a weighted least squares criterion).

In order to avoid post-processing and rough approximations, in Section 3.2 we exploit the centred log-ratio isometry between the spaces $\mathbb{D}^*(I)$, $\mathcal{L}_2^*(I)$ and define a factor model for approximating the clr-transformed densities (that is, $\mathrm{clr}(c_t)(\cdot) \approx \breve{f}_t(\cdot)$):

$$\breve{f}_t(\cdot) = \boldsymbol{\beta}_t'\boldsymbol{\xi}(\cdot) = \sum_{j=1}^{J} \beta_{t,j}\xi_j(\cdot)\,, \tag{18}$$

where $\xi_j(\cdot)$ are the principal component functions (or factors) and the coefficients $\beta_{t,j} \in \mathbb{R}$ are the principal component scores, both estimated by means of the fucntional principal component analysis. The factor model defines an approximation of the functions $\mathrm{clr}(c_t)(\cdot)$ by means of a finite linear combination of common, time-invariant factors with component specific time-varying scores. The optimality criterion given by the quadratic distance $||\mathrm{clr}(c_t)(\cdot) - \check{f}_t(\cdot)||_2$ is minimized by the choice of the principal component functions that maximize the explained variability of the series $\mathrm{clr}(c_t)(\cdot)$, $t = 1, \ldots, T$ (see Ramsay and Silverman (2005)).

Functional data analysis is a growing field of research and the existing results dealing with probability density functions are scarce. A possible interpretation of fPCA, in analogy with multivariate PCA, identifies the principal component functions with the eigenfunctions of their covariance operator of the observed functions. Following this interpretation, Hörmann et al. (2015) provided a remarkable extension of fPCA to time series functional data. They worked on the frequency domain using the techniques of Brillinger (2001) for estimating the dynamic principal component functions, which account for the temporal dependence among functional observations. Unfortunately, their results are not straightforwardly extendible to pdfs.

In fact, when dealing with pdfs, the estimation of the principal component functions $\boldsymbol{\xi}(\cdot)$ poses some issues which call for the development of specific procedures. Egozcue et al. (2006) has proved the analogy between probability density functions and compositional vectors, whcih are vectors belonging to the n-dimensional unit simplex $\mathcal{S}^n$ representing fractions or proportions and contitue the conrerstone of compositional data analysis (see Aitchison (1986)). Egozcue et al. (2006) interpreted pdfs as infinite-dimensional compositional vectors and translated into the functional domain the main results of compositional data analysis: this includes the definition of the operations of perturbation and powering (analogue of addition and scalar multiplication), $\oplus, \odot$, that make $(\mathbb{D}(I), \oplus, \odot)$ a space. van der Boogaart et al. (2010) and van der Boogaart et al. (2014) proved that $(\mathbb{D}(I), \oplus, \odot)$ is indeed a Hilbert space and showed that the centred log-ratio widely used in compositional data analysis is an isometry (i.e. an isometric isomorphism) between the spaces $\mathbb{D}^*(I), \mathcal{L}_2^*(I)$.

These results opened new possibilities to the functional analysis of pdfs. While it is possible to exploit the operations $\oplus, \odot$ for performing statistical analyses directly on $\mathbb{D}(I)$, the need for re-definition of standard techniques by $\oplus, \odot$ has lead the researchers to prefer the use of isometries. For the sake of working out fPCA of (transformed) univariate pdfs in $\mathcal{L}_2(I)$, Petersen and Müller (2016) proposed two isometries (the log-hazard and the log-quantile transforms) between $\mathbb{D}(K)$ and $\mathbb{D}(K)$, for K a compact subset of $\mathbb{R}$, whereas Hron et al. (2016) exploited the clr map. Other contributions in this area include Salazar et al. (2015), who proposed a forecasting model for univariate pdfs, and Menafoglio et al. (2014), who studied the problem of interpolation via the kriging method for probability density functions. The common strategy consists in three steps: the transformation of the pdfs into a suitable Hilbert subspace of $\mathcal{L}_2(I)$, where the statistical model is defined and the analysis is undertaken. Finally, the use of the inverse of the isometry for mapping the result back into $\mathbb{D}(I)$. Finally, the contribution of Machalovà et al. (2016) (see also Machalovà (2002a), Machalovà (2002b)) is based on the interpretation of fPCA as an eigenproblem, which the authors solved proposing a solution within the class of spline functions (see Appendix A for more details on fPCA and related solution methods). This allowed for the inclusion of the zero integral constraint as a constraint on the coefficients of the basis spline functions.

In this paper we follow van der Boogaart et al. (2014) and use the centred log-ratio transform to map pdfs into the Hilbert space $\mathcal{L}_2^*(I)$. Then, we extend to the multivariate framework the strategy developed by Machalovà et al. (2016) for dealing with the integral constraint for

univariate pdfs with compact support, thus obtaining a way to account for the constraint eq. (7) in the estimation of the principal component factors and scores.

The most appealing feature of the factor model in eq. (18) we specify is that all the information about the temporal dependence between the functions is carried by the scores, which form a vector-valued time series. Therefore, a forecast for the approximated function $\widetilde{f}_{T+h}(\cdot)$ at horizon $h \geq 1$, can be obtained by plugging-in a forecast for the scores, computed by well-known methods (e.g., VAR models). Then we get a forecast for the pdf, $\widetilde{c}_{T+h}(\cdot)$, by simply applying the inverse centred log-ratio map in eq. (6).

Our strategy shares some similarities with Liebl (2013) and Hron et al. (2016), but the methodologies differ in some key aspects. First and most important, we are interested in forecasting pdfs, which complicates the analysis with respect to the unrestricted case of Liebl (2013). Moreover, we extend the analysis of Hron et al. (2016) to the bivariate case (though the methodology generalizes easily to multidimensionality). Finally, we provide some remarks about how to deal with the case of densities with unbounded support.

3 Methodology

We propose a strategy for estimating the factor model in eq. (18), then forecasting the clr-transformed functions $\widetilde{f}_{T+h}(\cdot)$ and the corresponding pdfs $\widetilde{c}_{T+h}(\cdot)$, $h = 1, \ldots, H$. The methodology focuses on the forecast of bivariate copula probability density functions, however, the method is general and can be applied without structural changes to general multivariate pdfs with bounded support as well as to pdfs with unbounded support that satisfy an additional constraint (see Section 4). The modelling framework can be summarized as follows:

- in Section 3.1 we partition the raw dataset in sub-samples corresponding to different periods t, then for each of them we estimate the copula probability density function (or, in the generally case, the multivariate pdf).
  When dealing with copula pdfs, we use the nonparametric density estimator proposed by Chen (1999) for avoiding the boundary bias (for general multivariate pdfs we suggest standard product kernel estimators).

- next, in Section 3.2 we estimate the factor model in eq. (18) by a modified version of the functional principal component analysis. In this section we combine the centred log-ratio transform and spline functions for estimating the principal component functions and the scores ($\widehat{\boldsymbol{\xi}}(\cdot)$ and $\widehat{\mathbf{B}}$, respectively) such that the resulting functions $\breve{f}_t(\cdot)$ (approximating $\mathrm{clr}(c_t)(\cdot)$) satisfy the restrictions of probability density functions.
  We generalize the strategy proposed by Machalovà et al. (2016) and its application in Hron et al. (2016) to the multivariate (and potentially unbounded) case.

- finally, in Section 3.3 we estimate a VAR(p) process for the time series of scores previously estimated and forecast the scores h steps ahead, $h = 1, \ldots, H$. Then, we get the forecast of the approximated function $\breve{f}_{T+h}(\cdot)$ and by applying the inverse centred log-ratio transform we obtain a predicted copula probability density function (or the multivariate pdf) $\widetilde{c}_{T+h}(\cdot)$. The forecasting strategy extends Liebl (2013) from univariate unconstrained functions to multivariate pdfs.

Algorithm 1 synthetically represents the proposed strategy. Each block is described in detail in the following subsections.

Algorithm 1 Methodology

function COPULAESTIM($\mathbf{x}, \mathbf{y}$)
  a) split data $\{x_{t'}, y_{t'}\}_{t'=1}^{TN}$ into T sub-samples $\{x_{t,i}, y_{t,i}\}_{i=1}^{N}$ ▷ data for 2-steps
  for $t = 1, \ldots, T$ **do**
    b) compute pseudo-obs $\{u_{t,i}, v_{t,i}\}_{i=1}^{N}$ from $\{x_{t,i}, y_{t,i}\}_{i=1}^{N}$
    c) estimate copula $\widehat{c}_t(u, v)$ ▷ Beta kernel
    d) compute clr transform of copula values $\mathrm{clr}(\widehat{c}_t)(u_{t,i}, v_{t,i})$ ▷ clr
  end for
  return $(\mathbf{U}, \mathbf{V}, \mathbf{C}) = \{u_{t,i}, v_{t,i}, \mathrm{clr}(\widehat{c}_t)(u_{t,i}, v_{t,i})\}_{t,i}$
end function

function MOD_FPCA($\mathbf{U}, \mathbf{V}, \mathbf{C}, \bar{\boldsymbol{\lambda}}^{u,v}$)
  for $t = 1, \ldots, T$ **do**
    a) $(\mathbf{d}_t, \boldsymbol{\phi}) \leftarrow$ solve constrained optimal smoothing problem $(\mathbf{U}, \mathbf{V}, \mathbf{C}, \bar{\boldsymbol{\lambda}}^{x,y})$
  end for
  b) $(\widehat{\boldsymbol{\xi}}, \widehat{\mathbf{B}}) \leftarrow$ solve eigenproblem $(\mathbf{D}, \boldsymbol{\phi})$
  return $\widehat{\mathbf{B}} = (\widehat{\boldsymbol{\beta}}_1, \ldots, \widehat{\boldsymbol{\beta}}_T)$
end function

function PREDICTION($\widehat{\mathbf{B}}$)
  a) estimate VAR(p) for $\{\widehat{\boldsymbol{\beta}}_t\}_t$
  b) forecast scores $\widetilde{\mathbf{B}} = (\widetilde{\boldsymbol{\beta}}_{T+1}, \ldots, \widetilde{\boldsymbol{\beta}}_{T+H})$
  c) forecast transformed pdfs $\widetilde{\mathbf{f}}_{T+H}(\cdot) = (\widetilde{f}_{T+1}(\cdot), \ldots, \widetilde{f}_{T+H}(\cdot))'$
  d) forecast pdfs $\widetilde{\mathbf{c}}_{T+H}(\cdot) = (\widetilde{c}_{T+1}(\cdot), \ldots, \widetilde{c}_{T+H}(\cdot))'$ ▷ inverse clr
  return $\widetilde{\mathbf{c}}_{T+H}(\cdot)$
end function

3.1 Step 1 - Copula estimation

After the introduction of the empirical copula (Deheuvels (1978), Deheuvels (1979)), which is a nonparametric estimatior for the copula cumulative distribution function, several non-paramteric techniques for the estimation of a copula pdf and cdf have been proposed. We follow Chen (1999) and Charpentier et al. (2007) and estimate the copula pdfs from raw data via a product Beta kernel estimator. Among the main advantages of this approach we remark the greater flexibility with respect to parametric methods, the smoothness of the estimated function (as opposed to the empirical copula) and the absence of boundary bias.

Consider a sample of observations $\{x_{t'}, y_{t'}\}_{t'}$ of size T' (for instance, with daily frequency). First of all, we fix the reference period t (i.e., year, quarter) and split the raw sample accordingly into T sub-samples of size N (to be interpreted, for instance, as T years of N daily observations), $\{x_{t,i}, y_{t,i}\}_{i,t}$ of size N, for $t = 1, \ldots, T$. The reference period coincides with the frequency of the functional time series we want to analyse, whereas the intra-period observations are interpreted as noisy measurements of the discretized continuous function of interest $c_t(\cdot)$. Consequently, we are going to use the N data points in each period t to estimate the function $c_t(\cdot)$, then we use the resulting functional time series for performing forecasts of the probability density through a modified fPCA algorithm.

We exploit the intra-period information (i.e. N observations, for fixed period t) for estimating a copula pdf for each period, $\widehat{c}_t(\cdot)$. Recall that a copula probability density has uniformly distributed marginals representing the marginal cumulative distributions. As the latter are unknown, it is necessary to estimate them as first step. In practice, we compute the pseudo-

observations (see Nelsen (2013), Cherubini et al. (2004)) defined as follows:

$$(u_{t,i}, v_{t,i}) = \left(F_x^N(x_{t,i}), F_y^N(y_{t,i})\right) . \tag{19}$$

The marginals $F_x^N(x_{t,i}), F_y^N(y_{t,i})$ can be estimated via the empirical cumulative distribution function or through the rank transformed data.

We choose the second method, as it is computationally faster and provides distributions closer to the uniform. The rank transformation generates pseudo-observations according to (similarly for y):

$$R_{t,i}^x = \sum_{j=1}^{N} \mathbb{1}(x_{t,j} \leq x_{t,i}) , \qquad u_{t,i} = \frac{1}{N} R_{t,i}^x . \tag{20}$$

Given the pseudo observations, we estimate the copula probability density function by using a nonparametric kernel density estimator obtained as the product of univariate Beta kernels. Given a sample $\{x_t\}_{t=1}^T$, the Beta kernel density estimator (Chen (1999), Charpentier et al. (2007)) is defined as:

$$\widehat{f}_h(x) = \frac{1}{T} \sum_{t=1}^{T} \mathcal{K}_\beta \left(x_t; 1 + \frac{x}{h}, 1 + \frac{1-x}{h} \right) , \tag{21}$$

where $\mathcal{K}_\beta(\cdot; a, b)$ is the pdf of a Beta distribution with parameters (a, b) and h is the bandwidth. Alternative nonparametric methods for the estimation of a probability density function, such as the kernel estimator of Fermanian and Scaillet (2003), do not fit well the current framework because of the inadequateness of the methods to deal with the compact support. This causes a boundary bias problem if an unbounded kernel (such as the Gaussian) is used or a lack of smoothness, if the derivatives of the empirical copula distribution are chosen. Both shortcomings are instead solved by the product Beta kernel estimator. The price to pay is the lack of adequate rules of thumb for the specification of the bandwidth, which must be tuned case-by-case. The estimated smooth functions $(\widehat{c}_1(\cdot), \ldots, \widehat{c}_T(\cdot))$ are used to compute the values of the copula function at specific couples of pseudo-observations, that is $\widehat{c}_t(u_{t,i}, v_{t,j})$, for $i, j = 1, \ldots, N$, $t = 1, \ldots, T$. Finally, we apply the clr transform in eq. (5) to obtain $\mathbf{C}_t = (\mathrm{clr}(\widehat{c}_t)(u_{t,i}, v_{t,j}))_{ij}$, for $i, j = 1, \ldots, N$ and $t = 1, \ldots, T$. In compact notation, denote the matrices of pseudo-observations $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_T)$, $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_T)$ and the associated collection of matrices of clr-transformed copula pdf values $\mathbf{C} = [\mathbf{C}_1, \ldots, \mathbf{C}_T]$. The series of matrices $\mathbf{C}$ is required for estimating the constrained spline functions in Section 3.2.

In the general case, when the interest lies on multivariate pdfs with unbounded support, we propose to estimate the density via product kernel estimators, with standard choices of the univariate kernels such as Gaussian or Epanechnikov.

Concerning the interpretation of the method, we make the following remarks:

- functions are infinite-dimensional objects, thus from a computational perspective it is impossible to deal with them directly, but a discretization step is in order. Functional data in a strict sense do no exist, instead available data can be defined as noisy observations of discretized functions. Each discretized functional data point, broadly speaking, consists of a pair of location and value of the function at location (where location has no particular meaning). For instance, for univariate functions the location is the point on the x-axis, whereas the value at the location is the corresponding value of the function (on the y-axis). Therefore, in the current framework, we may think of the N rank transformed observations $\{u_{t,i}, v_{t,i}\}_i$, for a given t, as a set of location points whereas $\widehat{c}_t(u_{t,i}, v_{t,i})$ represents the value

of the copula function at those locations (i.e., a discretized version of the underlying smooth function). We remark that this is a standard interpretation in functional data analysis (e.g., see Ramsay and Silverman (2005)) and is unrelated to the procedure developed here.

- From a financial point of view, the copula pdfs are a flexible instrument providing all the information about the dependence between the marginal series $\mathbf{x}_t = \{x_{t,i}\}_i$ and $\mathbf{y}_t = \{y_{t,i}\}_i$ at time t. They are remarkably richer than a single scalar parameter: in addition to the extreme cases of independence (corresponding to a product copula) and perfect dependence (diagonal copula), they permit to study several particular forms of dependence, such as tail dependence (i.e. the probability of comovements of the variables in the upper/lower tail of the distribution, see Joe (1997), Nelsen (2013)).
  From this perspective, the availability of a (estimated) time series of copula pdfs permits to have information on different forms of dependence across several periods. Instead of limiting to a descriptive analysis on the variation of (finite-dimensional) synthetic statistics built from each function $c_t(\cdot)$, we aim at characterizing how the whole dependence pattern evolves over time.

3.2 Step 2 - Modified fPCA

Starting from $\mathbf{C}$, the time series of clr-transformed pdfs values estimated in Section 3.1, our goal in this section is to estimate the factor model in eq. (18) using the tools from functional principal component analysis (fPCA). In words, we estimate the function $\breve{f}_t(\cdot)$ that approximates the centred log-ratio transform of the pdf $c_t(\cdot)$, for $t = 1, \ldots, T$. In this section we are considering bivariate copula pdf, whose support is compact $[0, 1]^d$, with $d = 2$. See Section 4 for a discussion about the general frameworks when the pdfs have unbounded support or are multivariate with $d > 2$. The strategy does not impose any assumption except that the decay of the pdf at infinity must be such that its logarithm is square integrable. Moreover, given that probability density functions represent a special case of constrained functions, the proposed methodology can be applied as well for forecasting multivariate square integrable functions.

The outcome of this step is a vector of (time invariant) estimated factors $\widehat{\boldsymbol{\xi}}(\cdot)$ and a vector-valued time series of scores $\widehat{\mathbf{B}} = (\widehat{\boldsymbol{\beta}}_1, \ldots, \widehat{\boldsymbol{\beta}}_T)$ which will be used in Section 3.3 for building a forecast of the pdf $\widetilde{c}_{T+h}(\cdot)$, with $h = 1, \ldots, H$. Appendix A provides a summary of the results from functional data analysis used in this paper, we refer to Ramsay and Silverman (2005), Ferraty and Vieu (2006) for a more detailed presentation.

We present the outline of the strategy and the results, referring to Appendix B for detailed computations. Ordinary fCPA is designed for the analysis of unconstrained functions, however in our framework the object of interest are pdfs, that is functions constrained to be positive on their support and to have unit integral. This calls for a modification of standard fPCA in order to account for the constraints without the need to post-process the output. We propose a strategy for addressing this issue consisting in the exploitation of the centred log-ratio transform and spline functions. The clr transform allows the analysis to be carried out in the space $\mathcal{L}_2^*(I)$, which is preferred over $\mathbb{D}^*(I)$ due to its nicer properties that make easier ordinary calculus. Then, we are left with the estimation of the factor model in eq. (18), which we interpret as an eigenproblem. A first approach consists in the discretization of the functions involved and the solution of the resulting multivariate problem: despite being intuitive, this approach easily breaks down as the dimension increases because of the number of points necessary for providing a good discrete grid. Instead, we choose to express both the target function to be approximated by the factors and the factors themselves by a finite linear combination of pre-specified basis functions. This implicitly reduces the infinite-dimensional problem to an eigenproblem for the

vecor coefficients of the basis expansion. Following Machalovà et al. (2016), we choose a B-spline basis as it allows to analytically solve the resulting eigenproblem taking into account the integral constraint in eq. (7).

More formally, we propose to estimate the factor model in eq. (18) by interpreting the functions $\boldsymbol{\xi}(\cdot)$ as the eigenfunctions of the covariance operator of the functions $\breve{\mathbf{f}}(\cdot)$, denoted G. For each period $t = 1, \ldots, T$ and $k = 1, 2, \ldots$, this yields the eigenproblem:

$$\begin{cases} \displaystyle\int G(\mathbf{x}, \mathbf{y})\xi_k(\mathbf{x})\, \mathrm{d}\mathbf{x} = \beta_{t,k}\xi_k(\mathbf{y}) & \text{(22a)} \\ \displaystyle\int \xi_k(\mathbf{x})\xi_k(\mathbf{x})\, \mathrm{d}\mathbf{x} = 1 & \text{(22b)} \end{cases}$$

subject to the additional constraints $\langle \xi_k, \xi_j \rangle = 0$, for $k \neq j$ and $\int \xi_j\, \mu(\mathrm{d}\mathbf{x}) = 0$ for $j = 1, 2, \ldots$. Then, we look for a solution within the class of tensor product, bivariate spline functions (see (Ramsay and Silverman, 2005, ch.8) for a review of alternative solution methods), which allows to include the zero integral constraint as a linear constraint on the coefficients of the basis spline functions, thanks to the relation between splines with their derivatives.

Since a spline function can be expressed as a linear combination of known basis B-splines (see Section 2.1 for the notation), we need to solve a finite dimensional optimization problem for the coefficient vector of the spline. The constrained optimal smoothing problem, for each period $t = 1, \ldots, T$, is:

$$\begin{cases} \displaystyle\min_{s_m} \int_{a_1}^{b_1} \int_{a_2}^{b_2} \left[s_m^{(\ell_1,\ell_2)}(u, v) \right]^2 \mathrm{d}v\, \mathrm{d}u + \alpha \left[\sum_{i=1}^{N} \sum_{j=1}^{N} w_{i,j} \left(\mathrm{clr}(\widehat{c}_t)(u_{t,i}, v_{t,j}) - s_m(u_{t,i}, v_{t,j}) \right)^2 \right] \\ \text{s.t.} \displaystyle\int_{a_1}^{b_1} \int_{a_2}^{b_2} s_m(u, v)\, \mathrm{d}v\, \mathrm{d}u = 0 \end{cases} \tag{23}$$

where $s_m(\cdot, \cdot)$ is a spline of degree m, ℓ_1, ℓ_2 are the degree of the partial derivatives with respect to u, v, respectively, $\{u_{t,i}, v_{t,j}\}_{ij}$ with $i, j = 1, \ldots, N$, are the evaluation points and $\{\mathrm{clr}(\widehat{c}_t)(u_{t,i}, v_{t,j})\}_{ij}$ is the corresponding value of the clr-transformed pdf. Notice that N is number of observations allocated to each period $t = 1, \ldots, T$. $\{w_{i,j}\}_{ij}$ is a sequence of point-specific weights, whereas α is the global weight of the least squares component in the smoothing problem. Finally, the interval $(a_1, b_1) \times (a_2, b_2)$ is support of the original function and of the spline. In the following we assume: $a_1 = a_2 = 0$, $b_1 = b_2 = 1$, $\ell_1 = \ell_2 = 2$, meaning that we look for a solution in the class of cubic splines on the interval $[0, 1]^2$. Moreover, we consider an extended knot sequence given by the regular grid $\bar{\boldsymbol{\lambda}}^{u,v} = \bar{\boldsymbol{\lambda}}^u \otimes \bar{\boldsymbol{\lambda}}^v$, with:

$$\bar{\boldsymbol{\lambda}}^u = (\lambda^u_{-m}, \lambda^u_{-m+1}, \ldots, \lambda^u_{g+m+1})', \tag{24a}$$
$$\bar{\boldsymbol{\lambda}}^v = (\lambda^v_{-m}, \lambda^v_{-m+1}, \ldots, \lambda^v_{g+m+1})' \tag{24b}$$

with:

$$\lambda^u_{-m} = \ldots = \lambda^u_0 < \ldots < \lambda^u_{g+1} = \ldots = \lambda^u_{g+m+1} \tag{25a}$$
$$\lambda^v_{-m} = \ldots = \lambda^v_0 < \ldots < \lambda^v_{g+1} = \ldots = \lambda^v_{g+m+1}\,. \tag{25b}$$

This is a square grid with the same knots along both directions (that is, the x-axis and the y-axis, respectively), however we may choose a different number of interpolation knots for each dimension. We have decided to use the same number of knots and the same location because we are interpolating a copula probability density function with support $[0, 1]^2$.

Lemma 3.1
Define $\psi_k^{m+1}(\cdot)$, $k = 1, \ldots, K$ the B-spline basis functions of order m. The optimal spline function solving the problem in eq. (23) is given by:

$$s_m(u, v) = \mathbf{C}^{m+1}(u, v)\mathbf{d} = \sum_{k=1}^{K} d_k \psi_k^{m+1}(u, v)\,. \tag{26}$$

See Appendix B.1 for the detailed computations.

The spline functions in eq. (26) represent an interpolated multivariate probability density function, with evaluation points $(u_{t,i}, v_{t,i})_i$ and values $\mathrm{clr}(\widehat{c}_t)(u_{t,i}, v_{t,i})$, for $i = 1, \ldots, N$. By repeating this procedure for each sub-sample $(\mathbf{u}_t, \mathbf{v}_t)_t$, with $t = 1, \ldots, T$, we end up with a series of T multivariate spline functions satisfying the zero integral constraint. With a slight abuse of notation, define $\breve{f}_t(\cdot) = s_m(\cdot)$ the spline in eq. (26) estimated using the sub-sample $(\mathbf{u}_t, \mathbf{v}_t)_t$, for each period $t = 1, \ldots, T$. Therefore, we can write in compact notation:

$$\breve{f}_t(\cdot) = \sum_{k=1}^{K} d_{t,k} \psi_k^{m+1}(\cdot) = \mathbf{d}_t' \boldsymbol{\psi}(\cdot)\,, \tag{27}$$

where $\mathbf{d}_t = (d_{t,1}, \ldots, d_{t,K})'$ and $\boldsymbol{\psi}(\cdot) = (\psi_1^{m+1}(\cdot), \ldots, \psi_K^{m+1}(\cdot))'$. It is now possible to solve the eigenproblem in eq. (22a) using the same B-spline functions $\boldsymbol{\psi}(\cdot)$ as a basis for the principal component functions $\xi_j(\cdot)$, $j = 1, 2, \ldots$:

$$\xi_j(\cdot) = \sum_{k=1}^{K} a_{j,k} \psi_k^{m+1}(\cdot) = \mathbf{a}_j' \boldsymbol{\psi}(\cdot)\,, \tag{28}$$

where $\mathbf{a}_j = (a_{j,1}, \ldots, a_{j,K})'$. From this basis expansion, the infinite-dimensional eigenproblem in eq. (22a) reduces to a finite-dimensional optimization problem for the coefficient vectors $\mathbf{a}_j$, for $j = 1, \ldots, J$. For selecting the number of principal components J, we sort the estimated eigenvalues in decreasing order and compute the proportion of total variability explained by $v_j = \rho_j / \sum_k \rho_k$, for $j = 1, 2, \ldots$. Then, we retain the first J factors accounting for a given share $\bar{d}$ of the total variability, that is $J = \arg\min_j \{\sum_j v_j \geq \bar{d}\}$. The solution of this multivariate eigenproblem is obtained by first finding the optimal $\mathbf{u}_j$ satisfying (see Appendix B.2 for detailed computations):

$$T^{-1}\mathbf{M}^{1/2}\mathbf{D}'\mathbf{D}\mathbf{M}^{1/2}\mathbf{u}_j = \rho_j \mathbf{u}_j\,, \tag{29}$$

then transforming $\widehat{\mathbf{a}}_j = \mathbf{M}^{1/2}\widehat{\mathbf{u}}_j$, for $j = 1, 2, \ldots$. The solution of eq. (29) yields an estimate of the principal component functions by plugging $\widehat{\mathbf{a}}_j$ in eq. (28):

$$\widehat{\xi}_j(\cdot) = \widehat{\mathbf{a}}_j' \boldsymbol{\psi}(\cdot)\,. \tag{30}$$

Since the eigenvectors are not uniquely identified, we follow Liebl (2013) and transform them by applying the VARIMAX orthonormal rotation (see Kaiser (1958), Abdi (2003)). The eigenvalues provide an estimate for the scores $\widehat{\boldsymbol{\beta}}_t = (\widehat{\beta}_{t,1}, \ldots, \widehat{\beta}_{t,J})'$, for each period $t = 1, \ldots, T$. This coincide with (see Ramsay and Silverman (2005)):

$$\widehat{\boldsymbol{\beta}}_t = \begin{bmatrix} \langle \widehat{\xi}_1, \widehat{\xi}_1 \rangle & \ldots & \langle \widehat{\xi}_1, \widehat{\xi}_J \rangle \\ \vdots & \ddots & \vdots \\ \langle \widehat{\xi}_J, \widehat{\xi}_1 \rangle & \ldots & \langle \widehat{\xi}_J, \widehat{\xi}_J \rangle \end{bmatrix}^{-1} \begin{bmatrix} \langle \breve{f}_t, \widehat{\xi}_1 \rangle \\ \vdots \\ \langle \breve{f}_t, \widehat{\xi}_J \rangle \end{bmatrix}. \tag{31}$$

As final output of this step we obtain the estimated time series of scores $\widehat{\mathbf{B}} = (\widehat{\boldsymbol{\beta}}_1, \ldots, \widehat{\boldsymbol{\beta}}_T)$.

Each estimated eigenfunction can be seen as a continuous function of the clr-transformed functions, that is $\widehat{\xi}_j(\cdot) = g(\breve{f}_1(\cdot), \ldots, \breve{f}_T(\cdot))$. Hence, by the continuous mapping theorem, the estimator of the eigenfunction is consistent provided that the estimators for the clr-transformed functions are consistent too. Recall that each $\breve{f}_t$ corresponds to the centred log-ratio (continuous and smooth) transformation of a copula pdf, and it is estimated via a spline. It is known (see Schumaker (2007)) that splines approximate arbitrarily well continuous smooth functions on a bounded interval.

Consequently, from the consistency of splines in approximating a smooth function (as is $\breve{f}_t(\cdot)$, $t = 1, \ldots, T$ in our case) it descends the consistency of the estimator for each eigenfunction $\widehat{\xi}_j(\cdot)$ and, by another application of the continuous mapping theorem, the consistency of the estimator of the associated scores $\widehat{\boldsymbol{\beta}}_t$.

3.3 Step 3 - Prediction

In this last step, we aim at obtaining a H steps ahead forecast $\widetilde{c}_{T+H}(\cdot)$ of the pdf $c_t(\cdot)$. The task is accomplished in three steps: first, we estimate a VAR(p) process on the time series of estimated principal component scores from Section 3.2, $\{\widehat{\boldsymbol{\beta}}_t\}_{t=1}^T$, then we use the fitted values for obtaining a forecast of the scores $\widetilde{\boldsymbol{\beta}}_{T+h}$, $h = 1, \ldots, H$. Next, for $h = 1, \ldots, H$ we derive a forecast for the approximated function $\widetilde{f}_{T+h}(\cdot)$ by plugging-in eq. (18) and finally we get the forecast of the pdf $\widetilde{c}_{T+h}(\cdot)$ by applying the inverse clr transform to $\widetilde{f}_{T+h}(\cdot)$.

The estimated scores from the Section 3.2 for a vector-valued time series, where each vector has length J. We propose to model the time series through a VAR(p), as follows:

$$\widehat{\boldsymbol{\beta}}_t = \boldsymbol{\phi}_{const} + \boldsymbol{\phi}_{trend} t + \sum_{l=1}^{p} \boldsymbol{\Phi}_l \widehat{\boldsymbol{\beta}}_{t-l} + \boldsymbol{\epsilon}_t \,, \qquad \boldsymbol{\epsilon}_t \overset{iid}{\sim} \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_J) \,. \tag{32}$$

Denoting the estimated coefficients by $(\widehat{\boldsymbol{\phi}}_{const}, \widehat{\boldsymbol{\phi}}_{trend}, \widehat{\boldsymbol{\Phi}}_1, \ldots, \widehat{\boldsymbol{\Phi}}_p)$, we perform forecasts for each $h = 1, \ldots, H$ steps ahead in the usual way:

$$\widetilde{\boldsymbol{\beta}}_{T+h} = \widehat{\boldsymbol{\phi}}_{const} + \widehat{\boldsymbol{\phi}}_{trend}(T+h) + \sum_{l=1}^{p} \widehat{\boldsymbol{\Phi}}_l \widehat{\boldsymbol{\beta}}_{T-l} \,. \tag{33}$$

Then, we obtain the predicted clr-transformed function $\widetilde{f}_{T+h}(\cdot) \in \mathcal{L}_2^*(I)$ by substituting $\widetilde{\boldsymbol{\beta}}_{T+h}$ and the estimated principal components $\widehat{\boldsymbol{\xi}}(\cdot)$ into eq. (18), thus obtaining for $h = 1, \ldots, H$:

$$\widetilde{f}_{T+h}(\cdot) = \widetilde{\boldsymbol{\beta}}'_{T+h} \widehat{\boldsymbol{\xi}}(\cdot) = \sum_{j=1}^{J} \widetilde{\beta}_{T+h,j} \widehat{\xi}_j(\cdot) \,. \tag{34}$$

Finally, in order to compute the predicted probability density function $\widetilde{c}_{T+h}(\cdot) \in \mathbb{D}^*(I)$ we apply the inverse centred log-ratio transformation, for $h = 1, \ldots, H$:

$$\widetilde{c}_{T+h}(\cdot) = \mathrm{clr}^{-1}(\widetilde{f}_{T+h})(\cdot) = \frac{\exp\left\{\widetilde{f}_{T+h}(\cdot)\right\}}{\displaystyle\int \exp\left\{\widetilde{f}_{T+1}(\cdot)\right\}} \,. \tag{35}$$

The final outcome of the whole procedure is the set of forecasts of the multivariate pdf $\widetilde{\mathbf{c}}_{T+H}(\cdot) = (\widetilde{c}_{T+1}(\cdot), \ldots, \widetilde{c}_{T+H}(\cdot))'$.

The size of the VAR process in eq. (32) corresponds to the number of principal components selected in the fPCA, J and is generally small. Therefore, the dimensionality of the VAR does

not hamper the estimation procedure even though the length T of the time series is not really long. This is a consequence of the dimensionality reduction brought by the fPCA, interpreted as a factor model here.

Nonetheless, in higher-dimensional settings it may be still possible to estimate the coefficient matrix in eq. (32) by adding a regularization term. The recent contributions, Nicholson et al. (2016) and Nicholson et al. (2017) designed and implemented[1] several types of penalized regression for large VARX models (including the LASSO case) allowing up to $d = 130$ marginal series.

As regards the numerical implementation of the procedure, the core of the proposed methodology relies on standard linear algebra operations, for which computationally efficient algorithms are available. Moreover, the dimensionality reduction brought by the fPCA has the additional advantage of reducing the size of the coefficient matrix of the VAR process do be estimated. Overall, the entire procedure represented in Algorithm 1 is quite fast (see the details for the application in Section 6).

4 Extensions

Here we briefly discuss some possible extensions of the methodology discussed in Section 3.

4.1 Unbounded support

The results in van der Boogaart et al. (2010), van der Boogaart et al. (2014) hold also for pdfs with unbounded support, provided that they are absolutely continuous with respect to a measure with finite total mass. This requirement is a direct consequence of the formula for the centred log-ratio in eq. (5), which involves at the denominator the total mass of the support. In fact, the problem when dealing with pdfs defined on an unbounded region is that the Lebesgue measure of the whole domain is not finite, hence it would be necessary to choose a different, finite the reference measure of the spaces $\mathbb{D}(I), \mathbb{D}^*(I), \mathcal{L}_2(I), \mathcal{L}_2^*(I)$. If the new reference measure ν is absolutely continuous with respect to the Lebesgue measure, i.e. $\mathrm{d}\nu = g(\cdot)\,\mathrm{d}\mu$, then for $h(\cdot) \in \mathbb{D}_\nu(I)$ it holds $f(\cdot) = h(\cdot)g(\cdot) \in \mathbb{D}_\mu(I)$. Therefore in the particular case $\nu \ll \mu$ performing the analysis of the original pdf series $h_t(\cdot)$ under the reference measure ν is equivalent to perform the analysis of the modified series $h_t(\cdot)g(\cdot)$ under the Lebesgue measure.

Example 4.1 (Alternative reference measure)
Let $I = \mathbb{R}^n$ and let $g = \mathrm{d}\mathbb{P}_\mathcal{N}/\mathrm{d}\mu$ to be the Radon-Nikodym derivative of the finite standard Gaussian measure $\mathbb{P}_\mathcal{N}$ with respect to the n-dimensional Lebesgue measure on $\mathbb{R}^n$ (thus, g is the pdf of a standard normal distribution), the change of measure yields:

$$\int_{\mathbb{R}^n} f(\mathbf{x})\,\mathrm{d}\mu = \int_{\mathbb{R}^n} f(\mathbf{x})\,\frac{\mathrm{d}\mu}{\mathrm{d}\mathbb{P}_\mathcal{N}}\mathrm{d}\mathbb{P}_\mathcal{N} = \int_{\mathbb{R}^n} \frac{f(\mathbf{x})}{g(\mathbf{x})}\,\mathrm{d}\mathbb{P}_\mathcal{N} = \int_{\mathbb{R}^n} h(\mathbf{x})\,\mathrm{d}\mathbb{P}_\mathcal{N}\,.$$

If $\log(h)(\mathbf{x})$ is square integrable, then all the previous results can be applied, since $\mathbb{P}_\mathcal{N}(\mathbb{R}^n) = 1$. If instead $I = \mathbb{R}_+$ one may use the measure ω induced by a Gamma distribution, since $\omega(\mathbb{R}_+) = 1$.

Example 4.2 (Clr with unbounded support)
Let $p_0 = \mathrm{d}\mathbb{P}_\mathcal{N}/\mathrm{d}\mu$ be the density of the standard Gaussian measure with respect to the Lebesgue measure on $\mathbb{R}$. Let ν be a measure and $p_\nu = \mathrm{d}\nu/\mathrm{d}\mu$ be its density with respect to the Lebesgue measure. Let $g = \mathrm{d}\nu/\mathrm{d}\mathbb{P}_\mathcal{N}$ be the density of ν with respect to the Gaussian measure. Since

[1] Estimation can be carried out using the `R` (https://cran.r-project.org) package "`BigVAR`" (https://cran.r-project.org/web/packages/BigVAR/index.html), see Nicholson et al. (2017).

$\mu(\mathbb{R}) = \infty$, the centred log-ratio for g is not defined. However, by changing measure from μ to $\mathbb{P}_{\mathcal{N}}$ we obtain:

$$\begin{aligned}
\mathrm{clr}(g)(\cdot) &= \log\left(\frac{\mathrm{d}\nu}{\mathrm{d}\mathbb{P}_{\mathcal{N}}}\right)(\cdot) - \frac{1}{\mathbb{P}_{\mathcal{N}}(\mathbb{R})}\int_{\mathbb{R}} \log\left(\frac{\mathrm{d}\nu}{\mathrm{d}\mathbb{P}_{\mathcal{N}}}\right)(u)\,\mathrm{d}\mathbb{P}_{\mathcal{N}}(u) \\
&= \log\left(\frac{\mathrm{d}\nu}{\mathrm{d}\mu}\frac{\mathrm{d}\mu}{\mathrm{d}\mathbb{P}_{\mathcal{N}}}\right)(\cdot) - \int_I \log\left(\frac{\mathrm{d}\nu}{\mathrm{d}\mu}\frac{\mathrm{d}\mu}{\mathrm{d}\mathbb{P}_{\mathcal{N}}}\right)(u)\,\frac{\mathrm{d}\mathbb{P}_{\mathcal{N}}}{\mathrm{d}\mu}\,\mathrm{d}\mu(u) \\
&= \log\left(\frac{p_\nu}{p_0}\right)(\cdot) - \int_I \log\left(\frac{p_\nu}{p_0}\right)(u)\,p_0(u)\,\mathrm{d}\mu(u)\,. \qquad (36)
\end{aligned}$$

Notice that the integral on the last line is an expectation with respect to the probability measure $\mathbb{P}$, also Monte Carlo methods for numerical integration can be applied if the density p_0 can be easily sampled from, as is, for example, when p_0 is the pdf of a normal distribution.

Once a new reference measure has been chosen and the clr transform has been applied accordingly, the unbounded support is no more of concern for the methodology. In fact, the B-spline basis functions are defined also on unbounded regions and are computed for a given, finite knot sequence. The location of the knots would depend on the fatness of the tails of the densities, since fatter tails would require the knot sequence to be more scattered for having the resulting spline interpolating well the pdf. For example, a standard normal random variable has unbounded support, but almost the 95% of the mass in the interval $[-2, 2]$.

Consequently, the unboundedness of the support of the pdfs affect the spaces $\mathbb{D}^*_\nu(I), \mathcal{L}^*_{2,\nu}(I)$ to which the functions belong, but does not require a modification of the other parts of the procedure, since the basic constructions behind the result in eq. (26) are left unchanged.

4.2 Multivariate case: $d > 2$

The proposed methodology can be easily extended to deal with d-dimensional ($d > 2$) probability density functions. The change would be involve the size of the sparse block diagonal matrices described in Appendix B.

The only concern that arises when $d > 2$ is the curse of dimensionality, as is typical in nonparametric statistics. In the proposed model this occurs through the need for an increasingly high number of observations $\{x_{1,t,n}, \ldots, x_{d,t,n}\}_n$ for each period t in order to provide a good kernel estimation of the copula probability density function. In addition, if the high dimension is associated to a high degree of complexity of the dependence structure, it may be necessary also to increase the number of principal components to keep, J. This in turn results in a higher dimensionality of the VAR model for the scores in Section 3.3. However, we do not expect this to be a significant obstacle, as compared to the previous issue which represents the true bottleneck to high-dimensional applications.

5 Simulation example

To be included.

6 Application

The dataset is composed by daily observations of S&P500 and NASDAQ indices from 1st January 1980 to 31st December 2017, for a total of 10,032 observations over 38 years. We make the following assumptions. We start by taking first differences of the two series in order to remove non-stationarity, then for each period $t = 1, \ldots, T$ we assume to observe a sample $\{x_{t,i}, y_{t,i}\}_{i=1}^N$,

Lags	Model A	Model B	Model C	Model D
1	1823.5	1799.6	1821.5	1797.8
2	1831.7	1804.0	1827.0	1799.3
3	1834.2	1813.6	1826.3	1804.4
4	1820.7	1804.2	1801.2	1769.1

Table 2: BIC for different VAR specifications of the VAR(p) model in eq. (32). Model A: no constant, no trend; model B: constant, no trend; model C: trend, no constant; model D: constant and trend. The best model according to BIC is in bold.

with $N = 247$, of intra-period observations $(x_{t,i}, y_{t,i}) \in I = [0,1]^2$. We compute the copula pseudo-observations $(u_{t,i}, v_{t,i}) = (F_x^N(x_{t,i}), F_y^N(y_{t,i}))$, $i = 1, \ldots, N$, for each $t = 1, \ldots, T$ via the rank of the observations. Then, the empirical copula probability density function is estimated non-parametrically with the Beta kernel density estimator (Charpentier et al. (2007), Chen (1999)), using a diagonal bandwidth. The choice of the bandwidth for Beta kernel estimators is tricky since no rules of thumb are available for its optimal choice. As a consequence, we performed several experiments with varying h and found that the value $h = 0.05$ provides a good balance between bias and variance. Smaller or greater values resulted in too wiggly or too wide contours, respectively, which we interpreted as indicators of a bad value of the bandwidth.

The motivation for the choice of this splitting of the sample into $T = 38$ years is twofold. First, this allows us to estimate the function $c_t(\cdot)$, for each t, using up to $N = 248$ data points, while keeping a time series of estimated functions of length $T = 38$, thus providing a good balance of the data between the intra-period and the temporal dimensions. Second, a intra-period window corresponding to one year permits to embed the seasonal effects (with period up to one year) in the estimated function, thus removing the need to account for them when analysing the time series $c_t(\cdot)$, $t = 1, \ldots, T$. We take first differences of the raw data (see Appendix C for additional plots), thus reducing the size of each sub-sample to $N = 247$, in order to account for non-stationarity of the original series.

We choose the following values of the parameters:

$$\begin{array}{ccccc} T = 38 & N = 247 & H = 10 & h = 0.05 & \bar{d} = 0.92 \\ g = 4 & m = 3 & \ell = 2 & \alpha = 0.8 & \mathbf{W} = \mathbf{I}_{n^2} \end{array} \tag{37}$$

After having estimated the copula pdfs $\widehat{c}_1(\cdot), \ldots, \widehat{c}_{38}(\cdot)$, we de-meaned them using the perturbation and powering operations defined in Section 2, obtaining $\widehat{\widehat{c}}_t(\cdot) = \widehat{c}_t(\cdot) \ominus \bar{c}$, where $\bar{c} = 1/T \odot \bigoplus_{t=1}^T \widehat{c}_t(\cdot)$, which has been used as input for the step 2 of Algorithm 1.

The number of eigenfunctions to take has been estimated as described in Section 3.2, by $J = \arg\min_j \{\sum_j \widehat{\rho}_j \geq \bar{d}\}$, yielding $J = 4$. Values of α lower (greater) than unity imply higher (lower) relative weight of the smoothing component with respect to the least squares in the constrained optimal smoothing problem in eq. (23). We found that $\alpha = 0.8$ provides a good balance between the two. As robustness check, we performed the analysis with different values of $\bar{d}$ (thus implying different number of eigenfunctions J) without significant changes. We run Algorithm 1 on an Apple MacBookPro with a 3.1GHz Intel Core i7 processor, RAM 16GB, using MATLAB r2017b without exploiting parallel calculus. This required around ten minutes of computation, with step 1 being the most computational intensive part.

The value of the BIC for several specifications of the VAR(p) model in eq. (32) for the time series of scores are reported in Table 2 and suggest to choose a VAR(5) model including a constant and a time trend. All the estimated VAR models are stable.

For comparing the results, the estimated copula pdfs $\widehat{c}_t(\cdot)$, $t = 1, \ldots, T$ (respectively, the forecasted copula pdfs $\widetilde{c}_{T+h}(\cdot)$, $h = 1, \ldots, H$) have been computed by applying the inverse clr map to the functions $\breve{f}_t(\cdot)$ ($\widetilde{f}_{T+h}(\cdot)$) estimated (forecasted) from the factor model in eq. (18), using J eigenfunctions. Fig. 1 shows the contour plot of the time series of the estimated bivariate copula pdfs $\widehat{c}_t(\cdot)$, $t = 1, \ldots, T$, whereas Fig. 2 reports the contour and 3D density plots of the forecasted pdfs $\widetilde{c}_{T+h}(\cdot)$, $h = 1, \ldots, H$. We found that:

- there is evidence of significant temporal changes of the estimated pdfs $\widehat{c}_t(\cdot)$. Periods (i.e. years) where the joint probability is concentrated around the top-right corner, meaning strong upper tail dependence, alternate with periods where the opposite happens, that is, periods with strong lower tail dependence. There are two main implications of this stylized fact:
  - it signals that none of the copula families considered in Table 1, which are the most commonly used in econometrics, is able to account for the varying dependence over the whole time span of the sample, not even by letting the copula parameter vary over time. The reason is that all of them have either only one type of tail dependence (upper or lower), or both but in symmetric way. Moreover, the same conclusion holds even if a dynamic copula model is specified by allowing the copula parameter to change over time.
  - it is consistent with the results of Guégan and Zhang (2010) discussed in Section 1, who found that a dynamic copula model for the whole sample is not satisfactory and that different parametric copula families should be used for modelling different temporal windows.
- Fig. 7 in Appendix C shows the time series of the fPCA scores along with their forecasts (with 95% confidence intervals). For all series we do not reject the null hypothesis of stationarity (using the ADF test). Moreover, by comparing Fig. 1 with Fig. 7 we find that smooth evolutions of the fPCA scores of the clr-transformed pdfs are able to generate significant changes of the pdf.
- the forecasts of the bivariate copula pdf in Fig. 2 are smoothly varying over the forecasting horizon. The heterogeneity of the tail dependence observed in the sample seems to be replicated in the forecasts: for example, the upper tail dependence forecasted at horizon 2-3 is significantly reduced between horizons 4-7 and shows again at horizons 9-10. Consequently, we find that the proposed methodology is able to provide non-flat forecasts which can capture and describe the temporal evolution of the bivariate time series.

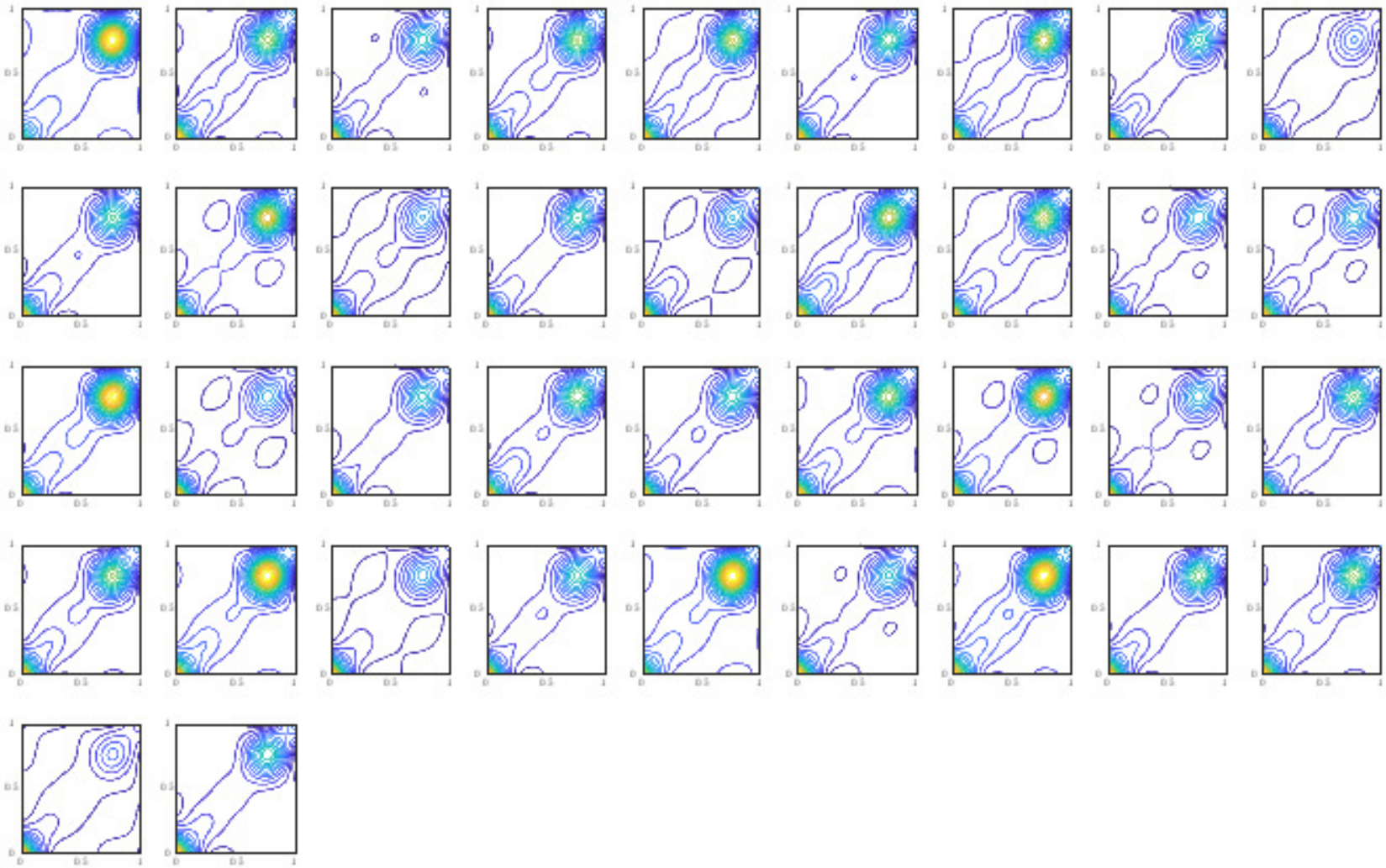

Figure 1: Contour plot of time series of bivariate copula pdfs, approximated via fPCA, for each year $t = 1, \ldots, T$, starting from $t = 1$ in the top-left panel.

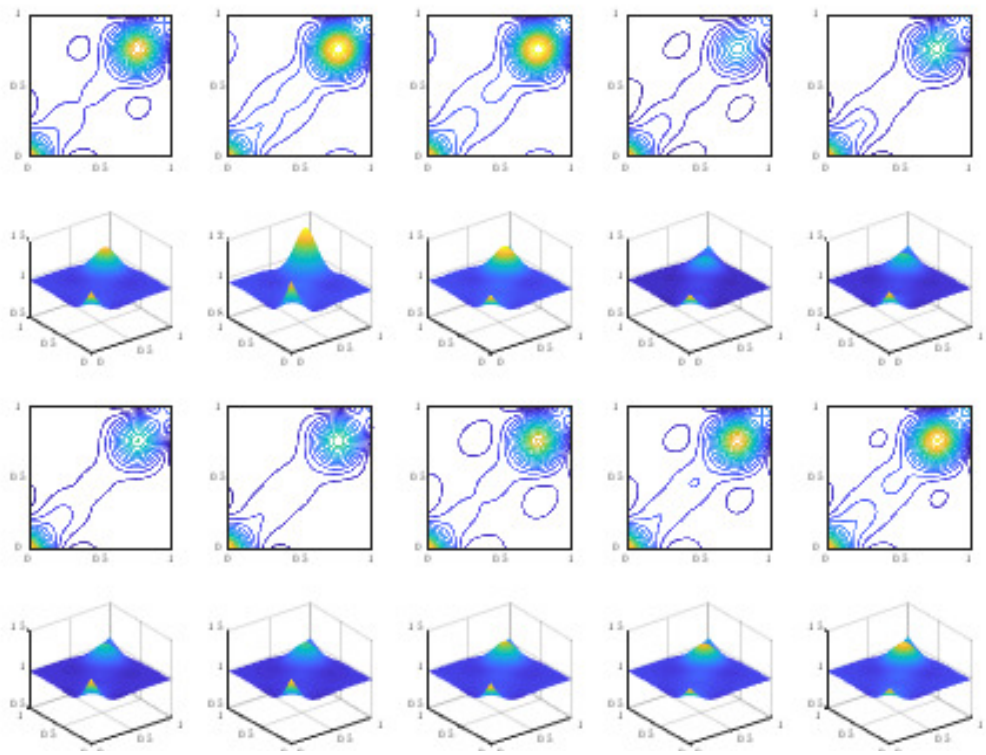

Figure 2: Contour plots (*first* and *third* row) and the corresponding 3D density plot (*second* and *fourth* row) of the forecasted bivariate copula pdfs, approximated via fPCA, for each horizon $h = 1, \ldots, 5$ (first and second rows) and $h = 6, \ldots, 10$ (third and fourth rows), starting from the top-left panel.

Several parametric and nonparametric estimators for the TDC λ_U, λ_L defined in eq. (1)-(3) have been proposed in the literature. Here we use the non-parametric estimator obtained from eq. (3). Let $u \in [0, 1]$ be an arbitrarily small threshold and let $\hat{C}_N(\cdot)$ be the empirical copula cumulative probability function, then the estimator is defined by (see Frahm et al. (2005)):

$$\widehat{\lambda}_U = 2 - \frac{\log(\hat{C}_N(1-u, 1-u))}{\log(1-u)}, \qquad \widehat{\lambda}_L = 2 - \frac{\log(1 - 2u + \hat{C}_N(u, u))}{\log(1-u)}. \tag{38}$$

Fig. 3 shows the estimated tail dependence coefficients for the sample observations, for each period $t = 1, \ldots, 38$, using a grid of 20 equally spaced threshold values between 0.01 and 0.20. Instead, Fig. 4 plots only the case for the median value of the threshold values, i.e. $u = 0.10$. We find significant variation of both the upper and lower tail dependence coefficients over time, which are always different from zero. In addition, the values of the upper TDC differ from those of the lower TDC, thus highlighting an asymmetric tail dependence. The threshold parameter seems to exert a minor role, as almost all the trajectories of both $\widehat{\lambda}_U$, $\widehat{\lambda}_L$ remain quite close to each other, except for few values of u, thus indicating robustness of the results (with respect to u). Moreover, the range of variation of the lower TDC is slightly higher than that of the upper TDC, in line with the previous findings in the financial econometrics literature (see Cherubini et al. (2004)).

Fig. 5 shows the estimated tail coefficients for the forecasted copula pdfs, for each horizon $h = 1, \ldots, 10$, using the threshold value $u = 0.10$. The results are in line with the findings of the sample data: both estimated coefficients are different from zero and asymmetric between the upper and lower case, furthermore they change over time. Considered together, the findings

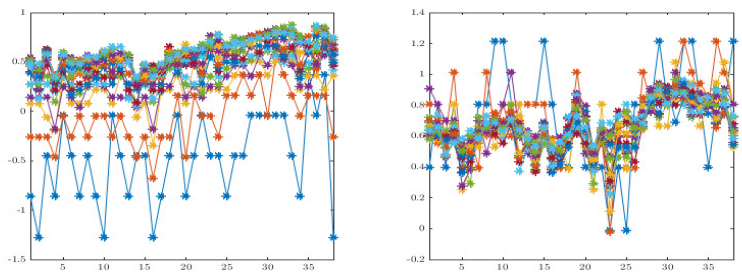

Figure 3: Upper (*left*) and lower (*right*) tail dependence coefficients of the bivariate time series $(\mathbf{x}_t, \mathbf{y}_t)$, for $t = 1, \ldots, 38$ (*x-axis*). Each curve corresponds to a different threshold $u = 0.01, 0.02, \ldots, 0.20$.

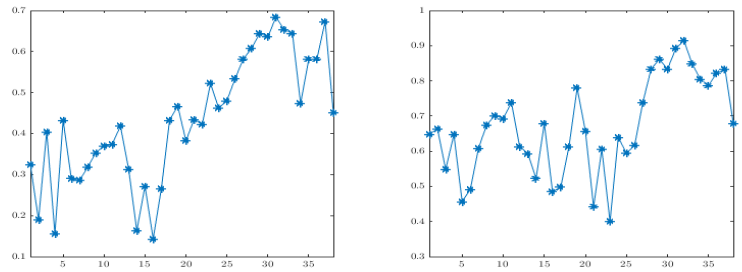

Figure 4: Upper (*left*) and lower (*right*) tail dependence coefficients of the bivariate time series $(\mathbf{x}_t, \mathbf{y}_t)$, for $t = 1, \ldots, 38$ (*x-axis*), threshold $u = 0.10$.

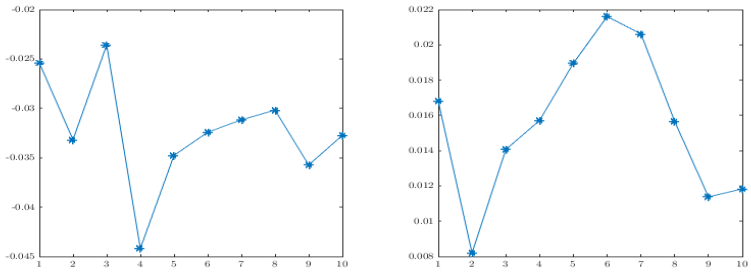

Figure 5: Upper (*left*) and lower (*right*) tail dependence coefficients of the forecasted bivariate copula pdf $c_{T+h}(\cdot)$, for $h = 1, \ldots, 10$ (*x-axis*), threshold $u = 0.10$.

in and out of sample estimated TDC points towards the rejection the use of copula families has either only one type of tail dependence or symmetric tail dependence, as show in Table 1.

Remark 6.1 (Interpretation)
The proposed methodology, as opposed to standard (semi)parametric dynamic copula models allows to visualize and quantify the temporal evolution of both the upper and lower tail dependence between bivariate time series, as well as to estimate the associated TDC. These findings suggest that the use of this methodology can improve the state-of-the-art on risk modelling due to its flexibility in modelling the dynamics of the dependence between random variables, which is the cornerstone for definition of adequate risk measures.

7 Conclusions

The time varying nature of the dependence pattern between financial variables is a challenging issue in statistics and econometrics. Common methods based on the specification of a dynamic copula model are not enough flexible to describe the temporal change, because each copula family has only a specific kind of tail dependence.

We contribute to this literature by proposing a nonparametric model for forecasting multivariate probability density functions with bounded or unbounded support. The methodology is used for studying the temporal evolution of the copula probability density function encrypting the dependence structure between the S&P500 and the NASDAQ indices. We found evidence of time varying tail dependence which cannot be captured by commonly used econometric models based on dynamic copulas, whereas the model we propose is able to account for these changes. The forecasts highlight smooth but significant variation of the bivariate copula pdf.

The proposed methodology is quite general and can be applied also to other domains. An appealing framework deserving further research concerns the definition of time varying graphical models through dynamic vine copulas (Bedford and Cooke (2002), Joe and Kurowicka (2011)), which combine a tree-like graphical structure, for representing the conditional independence relationships among a set of variables, with bivariate copulas, which describe the pairwise conditional dependence. Here, the method can be used for (separately) modelling the temporal evolution of each bivariate copula characterizing the edges of the network. Our methodology can be parallelized over the edges, for coupling with the issue of dimensionality.

Another stream of research worth further investigation regards the empirical analysis of multivariate (with dimension $d > 2$) pdfs with unbounded support, such as multivariate normal distributions, which are the building block of many well-known econometric models.

References

Abdi, H. (2003): "Factor rotations in factor analyses," *Encyclopedia for Research Methods for the Social Sciences. Sage: Thousand Oaks, CA*, 792–795.

Aitchison, J. (1986): *The statistical analysis of compositional data*, Chapman & Hall.

Almeida, C., C. Czado, and H. Manner (2016): "Modeling high-dimensional time-varying dependence using dynamic D-vine models," *Applied Stochastic Models in Business and Industry*, 32, 621–638.

Atkinson, K. (2009): *The numerical solution of integral equations of the second kind*, Cambridge Monographs on Applied and Computational Mathematics 4, Cambridge University Press.

Atkinson, K. and W. Han (2005): *Theoretical numerical analysis*, Springer.

Aue, A., D. D. Norinho, and S. Hörmann (2015): "On the prediction of stationary functional time series," *Journal of the American Statistical Association*, 110, 378–392.

Bartram, S. M., S. J. Taylor, and Y.-H. Wang (2007): "The Euro and European financial market dependence," *Journal of Banking & Finance*, 31, 1461–1481.

Bedford, T. and R. M. Cooke (2002): "Vines: a new graphical model for dependent random variables," *The Annals of Statistics*, 30, 1031–1068.

Bosq, D. (2000): *Linear processes in function spaces: theory and applications*, vol. 149, Springer Science & Business Media.

Brillinger, D. R. (2001): *Time series: data analysis and theory*, vol. 36, SIAM.

Canale, A. and M. Ruggiero (2016): "Bayesian nonparametric forecasting of monotonic functional time series," *arXiv preprint arXiv:1608.08056*.

Canale, A. and S. Vantini (2016): "Constrained functional time series: applications to the Italian gas market," *International Journal of Forecasting*, 32, 1340–1351.

Charpentier, A., J.-D. Fermanian, and O. Scaillet (2007): *Copulas: from theory to application in finance*, Risk Books, chap. The estimation of copulas: Theory and practice.

Chen, S. X. (1999): "Beta kernel estimators for density functions," *Computational Statistics & Data Analysis*, 31, 131–145.

Cherubini, U., E. Luciano, and W. Vecchiato (2004): *Copula methods in finance*, John Wiley & Sons.

Cherubini, U., S. Mulinacci, F. Gobbi, and S. Romagnoli (2011): *Dynamic copula methods in finance*, John Wiley & Sons.

De Boor, C. (2001): *A practical guide to splines*, Springer-Verlag New York.

Deheuvels, P. (1978): "Caractérisation complète des lois extrêmes multivariées et de la convergence des types extrêmes," *Publ. Inst. Statist. Univ. Paris*, 23, 1–36.

——— (1979): "La fonction de dépendance empirique et ses Proprétés. Un test non paramétrique d'indépendance," *Académie Royale de Belgique. Bulletin de la Classe des Sciences (5)*, 65, 274–292.

Dias, A. and P. Embrechts (2004): "Dynamic copula models for multivariate high-frequency data in finance," *Manuscript, ETH Zurich*, 81.

Durante, F. and C. Sempi (2015): *Principles of copula theory*, CRC Press.

Egozcue, J., J. Díaz-Barrero, and V. Pawlowsky-Glahn (2006): "Hilbert space of probability density function based on Aitchison geometry," *Acta Mathematica Sinica*, 22, 1175–1182.

Egozcue, J. and V. Pawlowsky-Glahn (2015): "Changing the reference measure in the simplex and its weighting effects," in *Welcome to CoDawork*.

Fermanian, J.-D. and O. Scaillet (2003): "Nonparametric estimation of copulas for time series," *Journal of Risk*, 5, 25–54.

——— (2004): "Some statistical pitfalls in copula modeling for financial applications," Tech. rep., International Center for Financial Asset Management and Engineering.

Fermanian, J.-D. and M. H. Wegkamp (2012): "Time-dependent copulas," *Journal of Multivariate Analysis*, 110, 19–29.

Ferraty, F. and P. Vieu (2006): *Nonparametric functional data analysis: theory and practice*, Springer Science & Business Media.

Frahm, G., M. Junker, and R. Schmidt (2005): "Estimating the tail-dependence coefficient: properties and pitfalls," *Insurance: mathematics and economics*, 37, 80–100.

Granger, C. W. (1988): "Some recent development in a concept of causality," *Journal of econometrics*, 39, 199–211.

Guégan, D. and J. Zhang (2010): "Change analysis of a dynamic copula for measuring dependence in multivariate financial data," *Quantitative Finance*, 10, 421–430.

Guidolin, M. and A. Timmermann (2006): "An econometric model of nonlinear dynamics in the joint distribution of stock and bond returns," *Journal of Applied Econometrics*, 21, 1–22.

Hafner, C. M. and H. Manner (2012): "Dynamic stochastic copula models: estimation, inference and applications," *Journal of Applied Econometrics*, 27, 269–295.

Hafner, C. M. and O. Reznikova (2010): "Efficient estimation of a semiparametric dynamic copula model," *Computational Statistics & Data Analysis*, 54, 2609–2627.

Hall, P., H.-G. Müller, and J.-L. Wang (2006): "Properties of principal component methods for functional and longitudinal data analysis," *The Annals of Statistics*, 34, 1493–1517.

Hays, S., H. Shen, and J. Z. Huang (2012): "Functional dynamic factor models with application to yield curve forecasting," *The Annals of Applied Statistics*, 6, 870–894.

Hörmann, S., L. Kidziński, and M. Hallin (2015): "Dynamic functional principal components," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77, 319–348.

Hörmann, S. and P. Kokoszka (2012): *Functional time series*, Elsevier, vol. 30.

Horváth, L., M. Hušková, and P. Kokoszka (2010): "Testing the stability of the functional autoregressive process," *Journal of Multivariate Analysis*, 101, 352–367.

Horváth, L., P. Kokoszka, and G. Rice (2014): "Testing stationarity of functional time series," *Journal of Econometrics*, 179, 66–82.

Hron, K., A. Menafoglio, M. Templ, K. Hruzová, and P. Filzmoser (2016): "Simplicial principal component analysis for density functions in Bayes spaces," *Computational Statistics & Data Analysis*, 94, 330–350.

Hu, J. (2010): "Dependence structures in Chinese and US financial markets: a time-varying conditional copula approach," *Applied Financial Economics*, 20, 561–583.

Joe, H. (1997): *Multivariate models and multivariate dependence concepts*, CRC Press.

Joe, H. and D. Kurowicka (2011): *Dependence modeling: vine copula handbook*, World Scientific.

Jondeau, E. and M. Rockinger (2006): "The copula-GARCH model of conditional dependencies: an international stock market application," *Journal of International Money and Finance*, 25, 827–853.

Kaiser, H. F. (1958): "The varimax criterion for analytic rotation in factor analysis," *Psychometrika*, 23, 187–200.

Kargin, V. and A. Onatski (2008): "Curve forecasting by functional autoregression," *Journal of Multivariate Analysis*, 99, 2508–2526.

Karhunen, K. (1947): "Über lineare methoden in der wahrscheinlichkeitsrechnung," *Annales Academiae Scientiarum Fennicae*, 1–79.

Kidziński, L. (2015): "Functional time series," *arXiv preprint arXiv:1502.07113*.

Kidziński, L., P. Kokoszka, and N. M. Jouzdani (2016): "Principal component analysis of periodically correlated functional time series," *arXiv preprint arXiv:1612.00040*.

Klepsch, J., C. Klüppelberg, and T. Wei (2017): "Prediction of functional ARMA processes with an application to traffic data," *Econometrics and Statistics*, 1, 128–149.

Kokoszka, P. (2012): "Dependent functional data," *ISRN Probability and Statistics*.

Li, Y. and T. Hsing (2010): "Uniform convergence rates for nonparametric regression and principal component analysis in functional/longitudinal data," *The Annals of Statistics*, 38, 3321–3351.

Liebl, D. (2010): "Modeling hourly electricity spot market prices as non stationary functional times series," Tech. rep., University Library of Munich, Germany.

——— (2013): "Modeling and forecasting electricity spot prices: a functional data perspective," *The Annals of Applied Statistics*, 7, 1562–1592.

Loève, M. (1945): "Fonctions aléatoires de second ordre," Tech. Rep. 220.

Lyche, T. and K. Morken (2008): *Spline methods draft*, Department of Informatics, Center of Mathematics for Applications, University of Oslo.

Machalovà, J. (2002a): "Optimal interpolating and optimal smoothing spline," *Journal of Electrical Engineering*, 53, 79–82.

——— (2002b): "Optimal interpolatory splines using B-spline representation," *Acta Universitatis Palackianae Olomucensis. Facultas Rerum Naturalium. Mathematica*, 41, 105–118.

Machalovà, J., K. Hron, and G. S. Monti (2016): "Preprocessing of centred logratio transformed density functions using smoothing splines," *Journal of Applied Statistics*, 43, 1419–1435.

Manner, H. and O. Reznikova (2012): "A survey on time-varying copulas: specification, simulations, and application," *Econometric Reviews*, 31, 654–687.

Menafoglio, A., A. Guadagnini, and P. Secchi (2014): "A kriging approach based on Aitchison geometry for the characterization of particle-size curves in heterogeneous aquifers," *Stochastic Environmental Research and Risk Assessment*, 28, 1835–1851.

Nelsen, R. B. (2013): *An introduction to copulas*, vol. 139, Springer Science & Business Media.

Nicholson, W., J. Bien, and D. Matteson (2016): "High dimensional forecasting via interpretable vector autoregression," *ArXiv e-prints*.

Nicholson, W. B., D. S. Matteson, and J. Bien (2017): "VARX-L: structured regularization for large vector autoregressions with exogenous variables," *International Journal of Forecasting*, 33, 627–651.

Oh, D. H. and A. J. Patton (2017): "Time-varying systemic risk: evidence from a dynamic copula model of CDS spreads," *Journal of Business & Economic Statistics*, 1–15.

Patton, A. J. (2006a): "Estimation of multivariate models for time series of possibly different lengths," *Journal of Applied Econometrics*, 21, 147–173.

——— (2006b): "Modelling asymmetric exchange rate dependence," *International Economic Review*, 47, 527–556.

——— (2012): "A review of copula models for economic time series," *Journal of Multivariate Analysis*, 110, 4–18.

Petersen, A. and H.-G. Müller (2016): "Functional data analysis for density functions by transformation to a Hilbert Space," *The Annals of Statistics*, 44, 183–218.

Petris, G. (2013): "A Bayesian framework for functional time series analysis," *arXiv preprint arXiv:1311.0098*.

Ramsay, J. O. and B. W. Silverman (2005): *Functional data analysis*, Springer Series in Statistics, Springer, second edition ed.

Salazar, E., R. Giraldo, and E. Porcu (2015): "Spatial prediction for infinite-dimensional compositional data," *Stochastic Environmental Research and Risk Assessment*, 29, 1737–1749.

Schumaker, L. (2007): *Spline functions: basic theory*, Cambridge University Press.

Sen, R. and C. Klüppelberg (2015): "Time series of functional data," *Reports of the Indian Statistical Institute*.

Sen, R. and C. Ma (2015): "Forecasting density function: application in finance," *Journal of Mathematical Finance*, 5, 433–447.

Sklar, A. (1959): *Fonctions de répartition à n dimensions et leurs marges*, Université Paris 8.

So, M. K. and C. Y. Yeung (2014): "Vine-copula GARCH model with dynamic conditional dependence," *Computational Statistics & Data Analysis*, 76, 655–671.

Staniswalis, J. G. and J. J. Lee (1998): "Nonparametric regression analysis of longitudinal data," *Journal of the American Statistical Association*, 93, 1403–1418.

Van Den Goorbergh, R. W., C. Genest, and B. J. Werker (2005): "Bivariate option pricing using dynamic copula models," *Insurance: Mathematics and Economics*, 37, 101–114.

van der Boogaart, K. G., J. J. Egozcue, and V. Pawlowsky-Glahn (2010): "Bayes linear spaces," *SORT: Statistics and Operations Research Transactions*, 34, 201–222.

——— (2014): "Bayes Hilbert spaces," *Australian & New Zealand Journal of Statistics*, 56, 171–194.

Weiss, G. N. and H. Supper (2013): "Forecasting liquidity-adjusted intraday value-at-risk with vine copulas," *Journal of Banking & Finance*, 37, 3334–3350.

White, H. and X. Lu (2010): "Granger causality and dynamic structural systems," *Journal of Financial Econometrics*, 8, 193–243.

Yao, F., H.-G. Müller, and J.-L. Wang (2005): "Functional data analysis for sparse longitudinal data," *Journal of the American Statistical Association*, 100, 577–590.

A Functional PCA

In this section, denote $\mathbf{f}(\cdot) = (f_1(\cdot), \ldots, f_T(\cdot))'$ a sequence of T random functions $f_t : \mathbb{R}^n \to \mathbb{R}$ and let $V : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be the covariance operator defined as:

$$V(f)(\cdot) = \int_{\mathbb{R}^n} v(\cdot, \mathbf{y}) f(\mathbf{y}) \, \mathrm{d}\mathbf{y} \,, \tag{39}$$

where the kernel $v : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, expressed as $v(\mathbf{x}, \mathbf{y})$, is the covariance function.

Functional principal component analysis (fPCA) is the infinite-dimensional analogue of multivariate principal component analysis (PCA), from which it borrows the terminology and interpretation (see (Ramsay and Silverman, 2005, ch.8) and Ferraty and Vieu (2006)). It is possible to interpret fPCA as a truncated the Karhunen-Loéve decomposition (Karhunen (1947), Loève (1945)). The latter is used to represent a function $f : \mathbb{R}^n \to \mathbb{R}$ via an infinite linear combination of basis functions $\xi_j(\cdot)$ with coefficients β_j given by:

$$f(\mathbf{x}) = \sum_{j=1}^{\infty} \beta_j \xi_j(\mathbf{x}) \,. \tag{40}$$

In fPCA, the infinite sum is truncated by keeping only J components, thus reducing the infinite-dimensional problem into a finite-dimensional one, given by $(\xi_j(\cdot), \beta_j)$, $j = 1, \ldots, J$. In fact, the purpose of fPCA is to find out the linear combination of principal component functions (or factors) $\boldsymbol{\xi}(\cdot) = (\xi_1(\cdot), \ldots, \xi_J(\cdot))'$ and principal component scores (or loadings) $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_J)'$, which best approximates a given function (or series of functions). The factors represent the main modes of variability and the scores specify the weight of each principal component function in the approximation of the observed function. Let $\breve{\mathbf{f}} = (\breve{f}_1(\cdot), \ldots, \breve{f}_T(\cdot))'$ the set of functions approximating the series $\mathbf{f} = (f_1(\cdot), \ldots, f_T(\cdot))'$. Then each $\breve{f}_t(\cdot)$ is obtained as:

$$\breve{f}_t(\cdot) = \boldsymbol{\beta}_t' \boldsymbol{\xi}(\cdot) = \sum_{j=1}^{J} \beta_{t,j} \xi_j(\cdot) \,. \tag{41}$$

For identification purposes, the principal component functions are often constrained to be orthonormal, that is $||\xi_j(\cdot)||_2 = 1$, $j = 1, \ldots, J$ and $\langle \xi_k(\cdot), \xi_j(\cdot) \rangle = 0$, for $k \neq j$.

Different criteria are available for the choice of the number J of principal components to take in the approximation[2] of eq. (45). We interpret the estimation of the factors as an eigenproblem (see next paragraph) and, after having sorted the estimated eigenvalues in decreasing order, we keep the first J eigenfunctions (corresponding to the factors) such that the proportion of variability explained is above a threshold $\bar{d}$. In the empirical analysis, regardless of the criterion used, the value of J is generally very small, thus allowing to interpret and use fPCA as a dimensionality reduction technique for the original series $\mathbf{f}(\cdot)$.

There are several ways to estimate the principal component functions, according to the interpretation of the problem (see Ramsay and Silverman (2005) and Ferraty and Vieu (2006) for a review). By interpreting them as the eigenfunctions of the covariance operator of the functions $(f_1(\cdot), \ldots, f_T(\cdot))$, we can estimate each pair (ξ_j, ρ_j) of principal component function and score by solving the eiganproblem:

$$\begin{cases} \displaystyle\int_{\mathbb{R}^n} V(\mathbf{x}, \mathbf{y}) \xi_j(\mathbf{x}) \, \mathrm{d}\mathbf{x} = \rho_j \xi_j(\mathbf{y}) & \text{(42a)} \\ \langle \xi_j(\cdot), \xi_j(\cdot) \rangle = 1 & \text{(42b)} \end{cases}$$

[2]Notice that the number and shape of the factors necessary to approximate a function provide information about its complexity.

subject to the additional constraint $\langle \xi_k(\cdot), \xi_j(\cdot) \rangle = 0$, for $k \neq j$. For $t = 1, \ldots, T$, $j = 1, \ldots, J$, the principal component scores, in the case of orthonormal eigenfunctions, satisfy:

$$\beta_{j,t} = \int_{\mathbb{R}^n} f_t(\mathbf{x}) \xi_j(\mathbf{x}) \, \mathrm{d}\mathbf{x} = \langle f_t(\cdot), \xi_j(\cdot) \rangle \,. \tag{43}$$

Following an alternative approach, each function $\xi_k(\cdot)$ is obtained by solving the optimization problem:

$$\begin{cases} \max\limits_{\xi_k} \; \dfrac{1}{T} \displaystyle\sum_{t=1}^{T} \left(\int_{\mathbb{R}^n} f_t(\mathbf{x}) \xi_k(\mathbf{x}) \, \mathrm{d}\mathbf{x} \right)^2 & \text{(44a)} \\ \text{s.t.} \;\; \left\| \xi_k(\mathbf{x}) \right\|_2 = 1 & \text{(44b)} \end{cases}$$

with the additional constraint that $\langle \xi_k, \xi_j \rangle = 0$, for $k \neq j$. For $t = 1, \ldots, T$, $j = 1, \ldots, J$, the scores are obtained again from eq. (43). In both cases, the output is a sequence of estimated factors $\widehat{\boldsymbol{\xi}}(\cdot) = (\widehat{\xi}_1(\cdot), \ldots, \widehat{\xi}_J(\cdot))'$ and scores $\widehat{\mathbf{B}} = (\widehat{\boldsymbol{\beta}}_1, \ldots, \widehat{\boldsymbol{\beta}}_T)$, with $\widehat{\boldsymbol{\beta}}_t = (\widehat{\beta}_{t,1}, \ldots, \widehat{\beta}_{t,J})'$ for $t = 1, \ldots, T$. Then, we obtain:

$$\mathbf{f}(\cdot) \approx \breve{\mathbf{f}}(\cdot) = \widehat{\mathbf{B}}' \widehat{\boldsymbol{\xi}}(\cdot) \,, \qquad \breve{f}_t(\cdot) = \widehat{\boldsymbol{\beta}}_t' \widehat{\boldsymbol{\xi}}(\cdot) \,. \tag{45}$$

In the paper we follow the first interpretation and estimate the principal component functions and scores by solving an eigenproblem. This poses the preliminary problem of estimating the covariance of the observed sample of functions $\mathbf{f}(\cdot)$. The standard sample covariance function estimator is given by:

$$\widehat{V}(\mathbf{x}, \mathbf{y}) = \frac{1}{T} \sum_{t=1}^{T} f_t(\mathbf{x}) f_t(\mathbf{y}) = T^{-1} \langle f_t(\cdot), f_t(\cdot) \rangle \,. \tag{46}$$

Alternative non-parametric estimators have been developed in the earlier contributions of Hall et al. (2006), Li and Hsing (2010), Yao et al. (2005) and Staniswalis and Lee (1998). In matrix notation, eq. (46) is written as $v(\mathbf{x}, \mathbf{y}) = T^{-1} \mathbf{f}'(\cdot) \mathbf{f}(\cdot)$. By exploiting eqs. (46) and (45) we get:

$$T^{-1} \mathbf{f}'(\cdot) \mathbf{f}(\cdot) = T^{-1} \boldsymbol{\xi}(\cdot) \mathbf{B}' \mathbf{B} \boldsymbol{\xi}(\cdot) \,. \tag{47}$$

Therefore, the k-th principal component function $\widehat{\xi}_k(\cdot)$ and the score $\widehat{\boldsymbol{\beta}}_t = (\widehat{\beta}_{t,1}, \ldots, \widehat{\beta}_{t,J})' = (\langle f_t(\cdot), \xi_1(\cdot) \rangle, \ldots, \langle f_t(\cdot), \xi_J(\cdot) \rangle)'$, for $t = 1, \ldots, T$, are obtained by solving the eigenproblem[3]:

$$V \xi_k(\cdot) = \rho_k \xi_k(\cdot) \,, \tag{48}$$

under the constraints $\langle \xi_k(\cdot), \xi_j(\cdot) \rangle = 0$ for $k \neq j$ and $||\xi_k(\cdot)||_2 = 1$. One way of solving the eigenproblem requires to discretize the functions on a specified grid of points $\{\mathbf{x}_i\}_{i=1}^N$, which permits to re-state eq. (48) as a finite-dimensional eigenproblem in matrix form. Then, standard methods used in multivariate PCA are applied for obtaining the solution.

Alternatively, one may assume that both the original functions $f_t(\cdot)$ and the eigenfunctions $\xi_k(\cdot)$ can be expressed as a finite linear combination of some chosen basis functions $\boldsymbol{\psi}(\cdot) = (\psi_1(\cdot), \ldots, \psi_K(\cdot))'$, with different coefficients:

$$f_t(\cdot) = \sum_{k=1}^{K} d_{t,k} \psi_k(\cdot) = \mathbf{d}_t' \boldsymbol{\psi}(\cdot) \,, \qquad \xi_j(\cdot) = \sum_{k=1}^{K} a_{j,k} \psi_k(\cdot) = \mathbf{a}_j' \boldsymbol{\psi}(\cdot) \,, \tag{49}$$

[3]This can also be interpreted as a n-dimensional Fredhölm integral equation of the second type, see Atkinson (2009), Atkinson and Han (2005)

for $t = 1, \dots, T$ and $j = 1, \dots, J$. Given the choice of the basis functions, this reduces the infinite-dimensional problem for $\xi_j(\cdot)$ to a finite-dimensional one for the vector $\mathbf{a}_j = (a_{j,1}, \dots, a_{j,K})'$. From eqs. (39), (46) and (48) we obtain:

$$T^{-1}\boldsymbol{\psi}(\cdot)'\mathbf{D}'\mathbf{D}\mathbf{M}\mathbf{a}_j = \rho_j \boldsymbol{\psi}(\cdot)'\mathbf{a}_j \tag{50}$$

$$T^{-1}\mathbf{D}'\mathbf{D}\mathbf{M}\mathbf{a}_j = \rho_j \mathbf{a}_j \,, \tag{51}$$

with $\mathbf{D} = (\mathbf{d}_1, \dots, \mathbf{d}_T)$ and $\mathbf{M} = (\langle \psi_k(\cdot), \psi_j(\cdot) \rangle)_{k,j}$, which is the identity matrix if the basis functions form an orthonormal system.

As common practice in multivariate PCA, the estimated eigenvalues $\widehat{\rho}_1, \widehat{\rho}_2, \dots$ are then sorted in decreasing order and the number J of principal component functions to take is decided on the basis of the proportion of total variation explained $J = \arg\min_j \{\sum_j \widehat{\rho}_j > \bar{d}\}$.

B Computations

In this section we provide the details of the computations needed in Section 3. We start by recalling a result from Lyche and Morken (2008) stating some useful properties of B-spline functions. A comprehensive discussion of spline functions and their properties can be found in De Boor (2001) and Schumaker (2007).

B.1 Proof of Lemma 3.1

In the following we show the procedure for solving the constrained optimal smoothing problem in eq. (23). Let $\bar{\boldsymbol{\lambda}}^{x,y} = \bar{\boldsymbol{\lambda}}^{x} \otimes \bar{\boldsymbol{\lambda}}^{y}$ denote an extended knot sequence (see Section 2.1 for the notation). We define the difference $\bar{\lambda}^{x,y}_{i,k} - \bar{\lambda}^{x,y}_{j,k}$ as the difference between the first coordinate, that is $\bar{\lambda}^{x,y}_{i,k} - \bar{\lambda}^{x,y}_{j,k} = \bar{\lambda}^{x}_{i} - \bar{\lambda}^{x}_{j}$ and $\bar{\lambda}^{x,y}_{k,i} - \bar{\lambda}^{x,y}_{k,j} = \bar{\lambda}^{y}_{i} - \bar{\lambda}^{y}_{j}$. In this section, for ease of notation we omit the bar and the superscripts and we implicitly refer to augmented knot sequences, that is we use $\lambda_{i,j}$ instead of $\bar{\lambda}^{x,y}_{i,j}$.

The integral constraint in eq. (23) yields:

$$\begin{aligned}
\int_{a_1}^{b_1}\int_{a_2}^{b_2} s_m(u,v)\,\mathrm{d}v\,\mathrm{d}u &= \int_{a_1}^{b_1} \tilde{s}_m(u,b_2) - \tilde{s}_m(u,a_2)\,\mathrm{d}u \\
&= s_{m+1}(b_1,b_2) - s_{m+1}(b_1,a_2) - s_{m+1}(a_1,b_2) + s_{m+1}(a_1,a_2) \\
&= s_{m+1}(\lambda_{g+1,g+1}) - s_{m+1}(\lambda_{g+1,0}) - s_{m+1}(\lambda_{0,g+1}) + s_{m+1}(\lambda_{0,0}) = 0
\end{aligned} \tag{52}$$

Starting from this result, we look for an equation allowing us to express the coefficient of a bivariate spline of order k with those of a spline obtained after differentiating it with respect to both arguments (that is, we look for an analogue of eq. (14)).

Now, we should derive the implication that the previous solution has on the coefficients $c_{i,j}$ of the spline $s_{m+1}(u,v)$ (e.g.: in the univariate case we end up with $0 = s_{m+1}(\lambda_{g+1}) - s_{m+1}(\lambda_0) = c_g - c_{-m-1}$ thus implying $c_{-m-1} = c_g$)[4].

By exploiting known propertis of splines (see Lyche and Morken (2008)) we obtain:

$$0 = s_{m+1}(\lambda_{g+1,g+1}) - s_{k+1}(\lambda_{g+1,0}) - s_{m+1}(\lambda_{0,g+1}) + s_{m+1}(\lambda_{0,0})\,, \tag{53}$$

$$0 = \sum_i \sum_j c_{ij}\left[B_i(\lambda_{g+1})B_j(\lambda_{g+1}) - B_i(\lambda_{g+1})B_j(\lambda_0) - B_i(\lambda_0)B_j(\lambda_{g+1}) + B_i(\lambda_0)B_j(\lambda_0)\right]\,. \tag{54}$$

By property (i) and (iii):

[4] They obtain the result by using the properties of B-splines in Lyche and Morken (2008). Some bases are exactly 1, others 0, reducing the spline function to the coefficient of the unique basis equal to 1.

- for $i = j = g$ it holds:

$$B_g(\lambda_{g+1})B_g(\lambda_{g+1}) - B_g(\lambda_{g+1})B_g(\lambda_0) - B_g(\lambda_0)B_g(\lambda_{g+1}) + B_g(\lambda_0)B_g(\lambda_0) = 1\,, \tag{55}$$

- for $i = j = -m-1$ it holds:

$$\begin{aligned} &B_{-m-1}(\lambda_{g+1})B_{-m-1}(\lambda_{g+1}) - B_{-m-1}(\lambda_{g+1})B_{-m-1}(\lambda_0) \\ &- B_{-m-1}(\lambda_0)B_{-m-1}(\lambda_{g+1}) + B_{-m-1}(\lambda_0)B_{-m-1}(\lambda_0) = 1\,, \end{aligned} \tag{56}$$

- for $i, j \notin \{-m-1, g\}$ the previous equation is always 0 since at least one of the terms of each product is 0.

Therefore we obtain:

$$s_{m+1}(\lambda_{g+1,g+1}) - s_{m+1}(\lambda_{g+1,0}) - s_{m+1}(\lambda_{0,g+1}) + s_{m+1}(\lambda_{0,0}) = 0\,, \tag{57}$$

which implies:

$$c_{g,g} + c_{-m-1,-m-1} = 0 \iff c_{g,g} = -c_{-m-1,-m-1}\,. \tag{58}$$

Now, by applying sequentially the recursion linking spline function with its partial derivatives and using a knot sequence (or an extended knot sequence) with equal number of knots along both directions:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}u}\frac{\mathrm{d}}{\mathrm{d}v}s_{m+1}(u,v) &= \frac{\mathrm{d}}{\mathrm{d}u}\frac{\mathrm{d}}{\mathrm{d}v}\sum_{i=-m-1}^{g}\sum_{j=-m-1}^{g} c_{ij}B_i^{m+2}(u)B_j^{m+2}(v) \\ &= \frac{\mathrm{d}}{\mathrm{d}v}\sum_{j=-m-1}^{g} B_j^{m+2}(v)\cdot\left(\frac{\mathrm{d}}{\mathrm{d}u}\sum_{i=-m-1}^{g} c_{ij}B_i^{m+2}(u)\right) \\ &= \frac{\mathrm{d}}{\mathrm{d}v}\sum_{j=-m-1}^{g} B_j^{m+2}(v)\cdot\left(\sum_{i=-m}^{g} c_{ij}^{u}B_i^{m+1}(u)\right) \qquad (59) \\ &= \sum_{i=-m}^{g} B_i^{m+1}(u)\left(\frac{\mathrm{d}}{\mathrm{d}v}\sum_{j=-m-1}^{g} c_{ij}^{u}B_j^{m+2}(v)\right) \qquad (60) \\ &= \sum_{i=-m}^{g}\sum_{j=-m}^{g} c_{ij}^{uy}B_i^{m+1}(u)B_j^{m+1}(v)\,. \end{aligned}$$

Since by definition:

$$s_m(u,v) = \sum_{i=-m}^{g}\sum_{j=-m}^{g} b_{ij}B_i^{m+1}(u)B_j^{m+1}(v)\,, \tag{61}$$

by assuming $b_{ij} = c_{ij}^{uy}$ we get the equality:

$$\frac{\mathrm{d}}{\mathrm{d}u}\frac{\mathrm{d}}{\mathrm{d}v}s_{m+1}(u,v) = s_m(u,v)\,. \tag{62}$$

It is now necessary to develop the above expression for the constraint on the coefficients in order to find out the precise relationship between the c_{ij} (coefficients of s_{m+1}) and b_{ij} (coefficients of s_m). This is required in order to derive the solution of eq. (23) by minimizing the first

equation, thus solving an unconstrained optimization problem. First, recall that the coefficients of a univariate spline are related to those of its first order derivative via the relation:

$$\frac{\mathrm{d}}{\mathrm{d}u} s_m(u) = \sum_i \check{c}_i B_i^m(u) = s_{m-1}(u) \qquad \check{c}_i = m \frac{c_i - c_{i-1}}{\lambda_{i+m} - \lambda_i} \,. \tag{63}$$

In the bivariate case, first define c^u_{ij}, for fixed j and $i = -m, \dots, g$, as:

$$c^u_{ij} = (m+1) \frac{c_{i,j} - c_{i-1,j}}{\lambda_{i+m+1,j} - \lambda_{i,j}} \,. \tag{64}$$

Then, iterated application eq. (63) along each dimension gives, for $j = -m, \dots, g$:

$$\begin{aligned} b_{i,j} = c^{uy}_{i,j} &= (m+1) \frac{c^u_{i,j} - c^u_{i,j-1}}{\lambda_{i,j+m+1} - \lambda_{i,j}} = \frac{(m+1)^2}{\lambda_{i,j+m+1} - \lambda_{i,j}} \left(\frac{c_{i,j} - c_{i-1,j}}{\lambda_{i+m+1,j} - \lambda_{i,j}} - \frac{c_{i,j-1} - c_{i-1,j-1}}{\lambda_{i+m+1,j-1} - \lambda_{i,j-1}} \right) \\ &= \frac{(m+1)^2}{\lambda_{i,j+m+1} - \lambda_{i,j}} \left(\frac{c_{i,j} - c_{i-1,j}}{\lambda_{i+m+1,j} - \lambda_{i,j}} - \frac{c_{i,j-1} - c_{i-1,j-1}}{\lambda_{i+m+1,j-1} - \lambda_{i,j-1}} \right) . \end{aligned} \tag{65}$$

This implies that the matrix $\mathbf{B}$ has the following top-left (i.e. $b_{-m,-m}$) and bottom-right (i.e. $b_{g,g}$) entries:

$$b_{-m,-m} = \frac{(m+1)^2}{\lambda_{-m,1} - \lambda_{-m,-m}} \left(\frac{c_{-m,-m} - c_{-m-1,-m}}{\lambda_{1,-m} - \lambda_{-m,-m}} - \frac{c_{-m,-m-1} - c_{-m-1,-m-1}}{\lambda_{1,-m-1} - \lambda_{-m,-m-1}} \right) , \tag{66}$$

$$b_{g,g} = \frac{(m+1)^2}{\lambda_{g,g+m+1} - \lambda_{g,g}} \left(\frac{c_{g,g} - c_{g-1,g}}{\lambda_{g+m+1,g} - \lambda_{g,g}} - \frac{c_{g,g-1} - c_{g-1,g-1}}{\lambda_{g+m+1,g-1} - \lambda_{g,g-1}} \right) . \tag{67}$$

We need conditions for linking the $(g+m+1) \times (g+m+1)$ coefficient matrix $\mathbf{B} = (b_{i,j})_{i,j}$ of the spline function $s_m(u,v)$ and the $(g+m+2) \times (g+m+2)$ coefficient matrix $\mathbf{C} = (c_{i,j})_{i,j}$ of the spline function $s_{m+1}(u,v)$. In the univariate case they are two vectors whose lengths differ by one, and the condition to be imposed consists in the equality of the first and last entry of the coefficient vector of the spline with higher degree. In the bivariate case, instead, $2(g+m+1)+1$ constraints are required:

$$\mathbf{C} = \left[\begin{array}{c|ccc} c_{-m-1,-m-1} & c_{-m-1,-m} & \dots & c_{-m-1,g} \\ \hline c_{-m,-m-1} & & & \\ \vdots & & \bar{\mathbf{C}} & \\ c_{g,-m-1} & & & \end{array} \right] \tag{68}$$

From the previous computations, we obtain the constraint:

$$c_{-m-1,-m-1} = -c_{g,g} \,. \tag{69}$$

We need to incorporate this result obtained from the integral constraint. From eq. (65) we have that[5]:

$$b_{i,j} = \frac{(m+1)^2}{\lambda_{i,j+(m+1)} - \lambda_{i,j}} \left[\frac{1}{\lambda_{i+(m+1),j} - \lambda_{i,j}} (c_{i,j} - c_{i-1,j}) - \frac{1}{\lambda_{i+(m+1),j-1} - \lambda_{i,j-1}} (c_{i,j-1} - c_{i-1,j-1}) \right]$$

[5] We used the notation $\mathbf{A}^i_{j,j}$ to mean the (j,j)-th entry of the diagonal matrix $\mathbf{A}^i$.

$$= \mathbf{D}^i_{j,j} \left[\mathbf{E}^j_{i,i} (\mathbf{K}\mathbf{c}_{:,j})_i - \mathbf{E}^{j-1}_{i,i} (\mathbf{K}\mathbf{c}_{:,j-1})_i \right] , \tag{70}$$

which, using the shorthand $N = (g + m + 1)$ and letting $\{\mathbf{e}_1, \ldots, \mathbf{e}_N\}$ be the canonical basis of the space of $N \times N$ matrices, gives the following expression for the column vector $\mathbf{b}_{:,j}$:

$$\begin{aligned}
\mathbf{b}_{:,j} &= \left[\sum_{k=1}^{N} \mathbf{e}_k \mathbf{e}'_k \otimes (\mathbf{D}^j_{k,k} \mathbf{E}^j_{k,k}) \right] \mathbf{K}\mathbf{c}_{:,j} - \left[\sum_{k=1}^{N} \mathbf{e}_k \mathbf{e}'_k \otimes (\mathbf{D}^{j-1}_{k,k} \mathbf{E}^{j-1}_{k,k}) \right] \mathbf{K}\mathbf{c}_{:,j-1} \\
&= \begin{bmatrix} \mathbf{D}^j_{1,1}\mathbf{E}^j_{1,1} & & & & \\ & \mathbf{D}^j_{2,2}\mathbf{E}^j_{2,2} & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \mathbf{D}^j_{N,N}\mathbf{E}^j_{N,N} \end{bmatrix} \begin{bmatrix} (\mathbf{K}\mathbf{c}_{:,j})_1 \\ (\mathbf{K}\mathbf{c}_{:,j})_2 \\ \vdots \\ (\mathbf{K}\mathbf{c}_{:,j})_N \end{bmatrix} \\
&\quad - \begin{bmatrix} \mathbf{D}^j_{1,1}\mathbf{E}^{j-1}_{1,1} & & & & \\ & \mathbf{D}^j_{2,2}\mathbf{E}^{j-1}_{2,2} & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \mathbf{D}^j_{N,N}\mathbf{E}^{j-1}_{N,N} \end{bmatrix} \begin{bmatrix} (\mathbf{K}\mathbf{c}_{:,j-1})_1 \\ (\mathbf{K}\mathbf{c}_{:,j-1})_2 \\ \vdots \\ (\mathbf{K}\mathbf{c}_{:,j-1})_N \end{bmatrix} .
\end{aligned} \tag{71}$$

For $j = -m, \ldots, g$, we define the $(g + m + 1) \times (g + m + 1)$ diagonal matrix $\mathbf{D}^i$ by:

$$\mathbf{D}^j = \operatorname{diag} \left(\frac{(m+1)^2}{\lambda_{-m,j+m+1} - \lambda_{-m,j}}, \frac{(m+1)^2}{\lambda_{-m+1,j+m+1} - \lambda_{-m+1,j}}, \ldots, \frac{(m+1)^2}{\lambda_{g,j+m+1} - \lambda_{g,j}} \right) . \tag{72}$$

and, for $j = -m - 1, -m, \ldots, g$, we define the $(g + m + 1) \times (g + m + 1)$ diagonal matrix $\mathbf{E}^j$ by:

$$\mathbf{E}^j = \operatorname{diag} \left(\frac{1}{\lambda_{1,j} - \lambda_{-m,j}}, \frac{1}{\lambda_{2,j} - \lambda_{-m+1,j}}, \ldots, \frac{1}{\lambda_{g+m+1,j} - \lambda_{g,j}} \right) . \tag{73}$$

The matrix $\mathbf{K}$ coincides with the matrix representation of the linear operator L which performs first differences $\mathbf{L}_{g+m+1}$ given by the $(g + m + 1) \times (g + m + 2)$:

$$\mathbf{K} = \mathbf{L}_{g+m+1} = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix} . \tag{74}$$

Therefore, for $j = -m, \ldots, g$, the $(g + m + 1) \times 1$ vector of first differences $\mathbf{K}\mathbf{c}_{:,j}$ is given by:

$$\mathbf{K}\mathbf{c}_{:,j} = \begin{bmatrix} c_{-m,j} - c_{-m-1,j} \\ c_{-m+1,j} - c_{-m,j} \\ \vdots \\ c_{g,j} - c_{g-1,j} \end{bmatrix} . \tag{75}$$

In order to rewrite eq. (70) in more compact form, we introduce the following $N \times (N + 1)$ matrices, which allow to select the first (or last, respectively) N columns from another one by

post-multiplication. Let $\{\mathbf{e}_1^N, \dots, \mathbf{e}_N^N\}$ be the canonical basis of the space of square matrices of size $N \times N$ and define the $N \times (N+1)$ matrix $P_{N,m} = [\mathbf{e}_1^N, \dots, \mathbf{e}_{m-1}^N, \mathbf{0}_N, \mathbf{e}_m^N, \dots, \mathbf{e}_N^N]$ with $m \in \{1, \dots, N+1\}$, which, by pre-multiplying a vector of length $(N+1)$, selects all but the m-th entry. The $(N+1) \times N$ matrices $\mathbf{S}_c^f, \mathbf{S}_c^\ell$ defined as follows, instead, when pre-multiplied by a $(N+1) \times (N+1)$ matrix $\mathbf{A}$ select the sub-matrix made with the first (last, respectively) N columns and rows $\mathbf{A}$:

$$\mathbf{S}_c^f = P'_{N,N+1} = \left[\begin{array}{ccc} 1 & & \\ & \ddots & \\ & & 1 \\ \hline 0 & \dots & 0 \end{array}\right], \qquad \mathbf{S}_c^\ell = P'_{N,1} = \left[\begin{array}{ccc} 0 & \dots & 0 \\ \hline 1 & & \\ & \ddots & \\ & & 1 \end{array}\right]. \tag{76}$$

For example, if $\mathbf{A} = [\mathbf{A}^f | \mathbf{a}_{N+1}] = [\mathbf{a}_1 | \mathbf{A}^\ell]$ then $\mathbf{A}\mathbf{S}_c^f = \mathbf{A}^f$ and $\mathbf{A}\mathbf{S}_c^\ell = \mathbf{A}^\ell$. Notice also that the transposed versions, that is $\mathbf{S}_r^f = (\mathbf{S}_c^f)'$ and $\mathbf{S}_r^\ell = (\mathbf{S}_c^\ell)'$, allow to select rows instead of columns. It is now possible to rewrite eq. (70) as:

$$\begin{aligned} \operatorname{vec}(\mathbf{B}) &= \begin{bmatrix} \mathbf{D}^{-m}\mathbf{E}^{-m} & & \\ & \ddots & \\ & & \mathbf{D}^g\mathbf{E}^g \end{bmatrix} \operatorname{vec}\left(\mathbf{S}_r^\ell \tilde{\mathbf{C}} \mathbf{S}_c^\ell\right) - \begin{bmatrix} \mathbf{D}^{-m}\mathbf{E}^{-m-1} & & \\ & \ddots & \\ & & \mathbf{D}^g\mathbf{E}^{g-1} \end{bmatrix} \operatorname{vec}\left(\mathbf{S}_r^f \tilde{\mathbf{C}} \mathbf{S}_c^f\right) \\ &= \mathbf{D}\mathbf{E} \operatorname{vec}\left(\mathbf{S}_r^\ell \tilde{\mathbf{C}} \mathbf{S}_c^\ell\right) - \mathbf{D}\mathbf{F} \operatorname{vec}\left(\mathbf{S}_r^f \tilde{\mathbf{C}} \mathbf{S}_c^f\right), \end{aligned} \tag{77}$$

where, letting $\{\mathbf{e}_{-m}, \dots, \mathbf{e}_g\}$ be the canonical basis for the space of square matrices of size $(g+m+1) \times (g+m+1)$, we defined:

$$\mathbf{D} = \sum_{i=-m}^{g} \mathbf{e}_i \mathbf{e}_i' \otimes \mathbf{D}^i \qquad \text{size } (g+m+1)^2 \times (g+m+1)^2$$

$$\mathbf{E} = \sum_{i=-m}^{g} \mathbf{e}_i \mathbf{e}_i' \otimes \mathbf{E}^i \qquad \text{size } (g+m+1)^2 \times (g+m+1)^2$$

$$\mathbf{F} = \sum_{i=-m-1}^{g-1} \mathbf{e}_i \mathbf{e}_i' \otimes \mathbf{E}^i \qquad \text{size } (g+m+1)^2 \times (g+m+1)^2$$

$$\tilde{\mathbf{C}} = \left[\mathbf{K}\mathbf{c}_{:,-m} | \dots | \mathbf{K}\mathbf{c}_{:,g+1}\right] \qquad \text{size } (g+m+1+1) \times (g+m+1+1),$$

and $\mathbf{S}_r^\ell \tilde{\mathbf{C}} \mathbf{S}_c^\ell$ (respectively, $\mathbf{S}_r^f \tilde{\mathbf{C}} \mathbf{S}_c^f$) select the bottom-right (respectively, top-left) square submatrix of $\tilde{\mathbf{C}}$ of size $(g+m+1) \times (g+m+1)$, denoted $(\tilde{\mathbf{C}})_{-m}^{g}$ (respectively, $(\tilde{\mathbf{C}})_{-m-1}^{g-1}$).

Now, we need to include in eq. (77) the information obtained in eq. (69) from the integral constraint in eq. (23), that is:

$$c_{-m-1,-m-1} = -c_{g,g}. \tag{78}$$

To this end, notice that there is no possibility of defining a unique matrix $\mathbf{K}$ which is able to give the desired result by a suitable choice of its entries (as opposed to the univariate case). This is mainly due to the fact that in the linear system representation of $\tilde{\mathbf{C}}$ the terms $c_{-m-1,-m-1}$ and $c_{g,g}$ are never in the same equation and the constraints on the other coefficients of $\mathbf{K}$ prevent to obtain the result. We propose to solve this issue as follows. Instead of transforming the matrix $\mathbf{C}$ and then vectorize it, reverse the order, that is, vectorize $\mathbf{C}$ then apply a suitable transformation. Then, define the $((g+m+1+1)^2 - 1) \times 1$ vector:

$$\tilde{\mathbf{c}} = P_{(g+m+1+1)^2-1,1} \cdot \operatorname{vec}(\mathbf{C}), \tag{79}$$

which corresponds to the vectorization of the matrix $\mathbf{C}$ without the element[6] $c_{-m-1,-m-1}$. Consider the $N \times (N+1)$ matrix $\mathbf{L}_N$ representing the first difference operator L:

$$\mathbf{L}_N = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix}. \tag{80}$$

Since we are dealing with the vectorisation of a matrix, $\mathrm{vec}\,(\mathbf{C})$, we must keep in mind that taking first differences of the whole vector implies taking differences also between the first entry of a column and the last of the previous one, which is undesired. Therefore we need to modify the structure of the difference operator matrix accordingly: we can do it by "shifting" to the right the blocks of non-zero entries every $N-1$ rows, where N is the number of rows of the original matrix $\mathbf{C}$. Moreover, taking into account the integral constraint[7] in eq. (69) we get in top right corner 1 instead of 0 whereas the first column of the difference operator matrix is removed. Consequently, we define $\mathbf{K}^*$ to be the matrix with number of columns equal to the size of $\tilde{\mathbf{c}}$ (that is, $(g+m+1+1)^2-1$) and number of rows equal to the number of entries of $\mathbf{C}$ minus a row (that is, $(g+m+1)(g+m+2)$), which is lost by taking differences. We obtain the $(g+m+1)(g+m+2) \times ((g+m+2)^2-1)$ block diagonal matrix:

$$\mathbf{K}^* = \left[\begin{array}{c|c|c|c|c} \mathbf{K}^{11} & & & & \mathbf{K}^{1N} \\ \hline & \mathbf{K}^{-m} & & & \\ \hline & & \mathbf{K}^{-m+1} & & \\ \hline & & & \ddots & \\ \hline & & & & \mathbf{K}^{g} \end{array}\right], \tag{81}$$

where each $(g+m+1) \times (g+m+2)$ matrix $\mathbf{K}^i = \mathbf{L}_{g+m+1}$, for $i = -m, \ldots, g$, whereas $\mathbf{K}^{1N}$ is a $(g+m+1) \times (g+m+2)$ with all zeros but the top-right entry and the $(g+m+1) \times (g+m+1)$ square matrix $\mathbf{K}^{11}$ is given by:

$$\mathbf{K}^{11} = \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix}. \tag{82}$$

For example, let I_R, I_C be the number of rows and columns in $\mathbf{C}$, respectively. Then the size of the matrix $\mathbf{K}^*$ is $(I_C(I_R-1)) \times (I_C I_R - 1)$.

By exploiting the previously defined diagonal matrices $\mathbf{D}^j, \mathbf{E}^j$, we define the following $(g+m+1)^2 \times (g+m+1)^2$ block diagonal matrices:

$$\mathbf{D} = \sum_{i=-m}^{g} \mathbf{e}_i \mathbf{e}_i' \otimes \mathbf{D}^i = \left[\begin{array}{c|c|c|c} \mathbf{D}^{-m} & & & \\ \hline & \mathbf{D}^{-m-1} & & \\ \hline & & \ddots & \\ \hline & & & \mathbf{D}^{g} \end{array}\right], \tag{83}$$

[6] We remove from $\mathrm{vec}\,(\mathbf{C})$ all the entries equal to $c_{-m-1,-m-1}$, thus obtaining a vector $\mathbf{c}^*$ of length equal to the length of $\mathrm{vec}\,(\mathbf{C})$ minus the number of occurrences of $c_{-m-1,-m-1}$.

[7] The constraint here has the opposite sign as compared to the univariate case, but it is coherent. In fact, it stems from the integral constraint and in the univariate case the integral is obtained by taking the difference between the value at the extrema of integration, while in the bivariate case the values at the top-right and bottom-left corners are added while those at the other two vertices of the rectangle are subtracted.

$$\mathbf{E} = \sum_{i=-m}^{g} \mathbf{e}_i \mathbf{e}_i' \otimes \mathbf{E}^i = \left[\begin{array}{c|c|c|c} \mathbf{E}^{-m} & & & \\ \hline & \mathbf{E}^{-m-1} & & \\ \hline & & \ddots & \\ \hline & & & \mathbf{E}^{g} \end{array}\right], \tag{84}$$

$$\mathbf{F} = \sum_{i=-m}^{g} \mathbf{e}_i \mathbf{e}_i' \otimes \mathbf{E}^{i-1} = \left[\begin{array}{c|c|c|c} \mathbf{E}^{-m-1} & & & \\ \hline & \mathbf{E}^{-m} & & \\ \hline & & \ddots & \\ \hline & & & \mathbf{E}^{g-1} \end{array}\right]. \tag{85}$$

Finally, we define $\mathbf{T}^f, \mathbf{T}^l$ to be two selection matrices of size $(g+m+1)^2 \times (g+m+1)(g+m+2)$ which select entries from a vector of length $(g+m+1)(g+m+2)$ by pre-multiplication:

$$\mathbf{T}^f = \left[\begin{array}{ccc|ccc} 1 & & & 0 & \dots & 0 \\ & \ddots & & \vdots & & \vdots \\ & & 1 & 0 & \dots & 0 \end{array}\right], \qquad \mathbf{T}^l = \left[\begin{array}{ccc|ccc} 0 & \dots & 0 & 1 & & \\ \vdots & & \vdots & & \ddots & \\ 0 & \dots & 0 & & & 1 \end{array}\right]. \tag{86}$$

Finally, define $\mathbf{A} = [\mathbf{E}\mathbf{T}^f - \mathbf{F}\mathbf{T}^l]\mathbf{K}^*$. We obtain the following equation relating the vectorized matrices of spline coefficients $\mathbf{B}$ and $\mathbf{C}$:

$$\overline{\mathbf{b}} = \mathrm{vec}\,(\mathbf{B}) = \mathbf{D}\left[\mathbf{E}\mathbf{T}^f\mathbf{K}^*\tilde{\mathbf{c}} - \mathbf{F}\mathbf{T}^l\mathbf{K}^*\tilde{\mathbf{c}}\right] = \mathbf{D}\left[\mathbf{E}\mathbf{T}^f - \mathbf{F}\mathbf{T}^l\right]\mathbf{K}^*\tilde{\mathbf{c}} = \mathbf{D}\mathbf{A}\tilde{\mathbf{c}}\,, \tag{87}$$

The next step consists in re-writing the objective function of the optimization problem (23) using matrix notation. First of all, since the B-spline basis for the bivariate spline is the product of two univariate B-splines, given a sample $(\mathbf{z}, \mathbf{u}, \mathbf{v}) = \{z_i, (u_i, v_i)\}_{i=1}^n$ we define the modified version of the matrix $\mathbf{C}^{m+1}(\mathbf{u})$ used in the univariate case as follows:

$$\mathbf{C}^{m+1}(\mathbf{u}, \mathbf{v}) = \begin{bmatrix} B_{-m}^{m+1}(u_1)B_{-m}^{m+1}(v_1) & \dots & B_{g}^{m+1}(u_1)B_{-m}^{m+1}(v_1) & \dots & B_{g}^{m+1}(u_1)B_{g}^{m+1}(v_1) \\ \vdots & & & & \vdots \\ B_{-m}^{m+1}(u_n)B_{-m}^{m+1}(v_n) & \dots & B_{g}^{m+1}(u_n)B_{-m}^{m+1}(v_n) & \dots & B_{g}^{m+1}(u_n)B_{g}^{m+1}(v_n) \end{bmatrix}, \tag{88}$$

whose size is $n \times (g+m+1)^2$ and generic entry $\mathbf{C}_{i,j}^{m+1}(\mathbf{u}, \mathbf{v}) = B_{j_1}^{m+1}(u_i)B_{j_2}^{m+1}(v_i)$, with j_1, j_2 obtained by inverting[8] the linear indexing $j = j_1 + (j_2 - 1)(g+m+1)$. Each row is constructed by first fixing the index for the B-spline along the direction of v, that is $B_j^{m+1}(v_i)$, then considering all the combinations with the B-spline along the direction of u, that is $B_h^{m+1}(u_i)$. Then the index j is incremented and the process is iterated until the exhaustion of the basis. Notice that each row of the matrix corresponds to the same observation point. This construction is necessary to rewrite a bivariate spline function given the observation points $\{(u_i, v_i)\}_{i=1}^n$ in matrix form, obtaining a vector of size $n \times 1$:

$$s_m(\mathbf{u}, \mathbf{v}) = \mathbf{C}^{m+1}(\mathbf{u}, \mathbf{v})\overline{\mathbf{b}}\,. \tag{89}$$

In order to write the matrix form of the integral of the squared derivative (of order $\ell \leq m-1$) of the bivariate spline function, start by defining the $(g+m+1-\ell)^2 \times (g+m+1-\ell)^2$ matrix

[8] The inversion is obtained by solving a linear system with two equations and two unknowns, j_1, j_2, which has a unique solution: $j_1 = j - (j_2 - 1)(g+m+1)$ and $j_2 = \lfloor j/(g+m+1) \rfloor$, where $\lfloor x \rfloor$ denote the integer part of x.

of inner products of the B-spline basis functions of order $m-\ell$ as follows:

$$\mathbf{M}_{m,\ell} = \begin{bmatrix} \langle B_{-m+\ell}^{m+1-\ell} B_{-m+\ell}^{m+1-\ell},\, B_{-m+\ell}^{m+1-\ell} B_{-m+\ell}^{m+1-\ell} \rangle & \dots & \langle B_{g}^{m+1-\ell} B_{g}^{m+1-\ell},\, B_{-m+\ell}^{m+1-\ell} B_{-m+\ell}^{m+1-\ell} \rangle \\ \vdots & & \vdots \\ \langle B_{-m+\ell}^{m+1-\ell} B_{-m+\ell}^{m+1-\ell},\, B_{g}^{m+1-\ell} B_{g}^{m+1-\ell} \rangle & \dots & \langle B_{g}^{m+1-\ell} B_{g}^{m+1-\ell},\, B_{g}^{m+1-\ell} B_{g}^{m+1-\ell} \rangle \end{bmatrix}, \tag{90}$$

where the generic entry is $\mathbf{M}_{m,\ell;i,j} = \langle B_{j_1}^{m+1-\ell} B_{j_2}^{m+1-\ell}, B_{i_1}^{m+1-\ell} B_{i_2}^{m+1-\ell} \rangle$, where i_1, i_2, j_1, j_2 are obtained, as for $\mathbf{C}_{i,j}^{m+1}(\mathbf{u},\mathbf{v})$, by inverting the linear indexing $i = i_1 + (i_2-1)(g+m+1-\ell)$ and $j = j_1 + (j_2-1)(g+m+1-\ell)$. The inner product is defined in the usual way (see Algorithm 5.22 in Schumaker (2007) for numerical computation) as:

$$\begin{aligned} &\langle B_i^{m+1-\ell} B_j^{m+1-\ell},\, B_h^{m+1-\ell} B_l^{m+1-\ell} \rangle \\ &= \int_{\lambda_0}^{\lambda_g} \int_{\lambda_0}^{\lambda_g} B_i^{m+1-\ell}(u)\, B_j^{m+1-\ell}(v)\, B_h^{m+1-\ell}(u)\, B_l^{m+1-\ell}(v)\ \mathrm{d}u\, \mathrm{d}v \geq 0\,. \end{aligned} \tag{91}$$

Under the assumptions made in the text, that is $(a_1, b_1) = (\lambda_{0,0}, \lambda_{g,g})$, $(a_2, b_2) = (\lambda_{u,0}, \lambda_{u,g})$, $\ell_1 = \ell_2 = \ell$, $n_1 = n_2 = n$, the objective function in eq. (23) can be re-written as the sum of two terms:

$$\begin{aligned} J_\ell(s_m) &= \int_{a_1}^{b_1} \int_{a_2}^{b_2} \left[s_m^{(\ell_1,\ell_2)}(u,v) \right]^2 \mathrm{d}v\, \mathrm{d}u + \alpha \left[\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} w_{ij} \left(z_{ij} - s_m(u_i, v_j) \right)^2 \right] \\ &= \int_{\lambda_{0,0}}^{\lambda_{g,g}} \int_{\lambda_{u,0}}^{\lambda_{u,g}} \left[s_m^{(\ell,\ell)}(u,v) \right]^2 \mathrm{d}v\, \mathrm{d}u + \alpha \left[\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \left(z_{ij} - s_m(u_i, v_j) \right)^2 \right] \\ &= \int_{\lambda_{0,0}}^{\lambda_{g,g}} \int_{\lambda_{u,0}}^{\lambda_{u,g}} \left[s_m^{(\ell,\ell)}(u,v) \right]^2 \mathrm{d}v\, \mathrm{d}u + \alpha \left[\sum_{i'=1}^{n'} w_{i'} \left(z_{i'} - s_m(u_{i'}, v_{i'}) \right)^2 \right] \qquad (92) \\ &= J_\ell^1(s_m) + J_\ell^2(s_m)\,. \qquad (93) \end{aligned}$$

The double sum has been reduced to a single sum under the hypothesis that the sample consists of a value z_i and a point (u_i, v_i). The third line has been obtained after vectorization. The extrema of integration are the same as in the univariate case, but now it is necessary to stress formally that when integrating with respect to v, the extrema of integration in principle may depend on u. The idea is nonetheless simple: the area of integration consists of the points included in the square with vertices $(\lambda_{0,0}, \lambda_{0,g}, \lambda_{g,0}, \lambda_{g,g})$. Concerning the derivative, by choosing $\ell_1 = \ell_2 = \ell = 2$ we are performing second-order derivative, thus obtaining a solution in the class of cubic spline functions.

Let $\mathbf{Z} = (z_{ij})_{ij}$, $\mathbf{W} = (w_{ij})_{ij}$ be $n \times n$ matrices and $\mathbf{z} = \mathrm{vec}\,(\mathbf{Z})$, $\mathbf{w} = \mathrm{vec}\,(\mathbf{W})$ be their vectorization. By exploiting eq. (89), the second component of eq. (93) can be written in matrix form as:

$$J_\ell^2(\overline{\mathbf{b}}) = \alpha \left(\mathbf{z} - \mathbf{C}^{m+1}(\mathbf{u},\mathbf{v})\overline{\mathbf{b}} \right)' \mathbf{W} \left(\mathbf{z} - \mathbf{C}^{m+1}(\mathbf{u},\mathbf{v})\overline{\mathbf{b}} \right)\,. \tag{94}$$

As for the first addendum, since the derivative of spline is another spline of lower degree, it can be represented in matrix form. In particular, it is given by the product of the vectorised coefficient matrix $\overline{\mathbf{b}}$ and a vector of B-spline basis functions $\mathbf{g}(u,v)$ defined as:

$$\mathbf{g}(u,v) = \left[B_{-m}^{m+1} B_{-m}^{m+1}(v), \dots, B_g^{m+1}(u) B_{-m}^{m+1}(v), \dots, B_g^{m+1}(u) B_g^{m+1}(v) \right]\,. \tag{95}$$

This allows to write:

$$
\begin{aligned}
J_\ell^1(s_m) &= \int_{\lambda_{0,0}}^{\lambda_{g,g}} \int_{\lambda_{u,0}}^{\lambda_{u,g}} [s_m^{(\ell,\ell)}(u,v)]^2 \, \mathrm{d}v \, \mathrm{d}u = \int_{\lambda_{0,0}}^{\lambda_{g,g}} \int_{\lambda_{u,0}}^{\lambda_{u,g}} \overline{\mathbf{b}}^{(\ell)\prime} \mathbf{g}(u,v)' \mathbf{g}(u,v) \overline{\mathbf{b}}^{(\ell)} \, \mathrm{d}v \, \mathrm{d}u \\
&= \overline{\mathbf{b}}^{(\ell)\prime} \left[\int_{\lambda_{0,0}}^{\lambda_{g,g}} \int_{\lambda_{u,0}}^{\lambda_{u,g}} \mathbf{g}(u,v)' \mathbf{g}(u,v) \, \mathrm{d}v \, \mathrm{d}u \right] \overline{\mathbf{b}}^{(\ell)} \\
&= \overline{\mathbf{b}}^{(\ell)\prime} \mathbf{M}_{m,\ell} \overline{\mathbf{b}}^{(\ell)} = J_\ell^1(\overline{\mathbf{b}}) \,, \qquad (96)
\end{aligned}
$$

where the last line follows from the definition of the matrix $\mathbf{M}_{m,\ell}$. We are left to find an explicit form for the vectorised coefficient matrix of the original spline of degree m, $\overline{\mathbf{b}}$, and that of its ℓ-th derivative, $\overline{\mathbf{b}}^{(\ell)}$. Recall the previous manipulation of the integral constraint gave a linear relation between $\overline{\mathbf{b}}$ and $\tilde{\mathbf{c}}$, with a restriction was accounted for in the construction of the matrix $\mathbf{K}^*$. In the problem at hand there no constraints, therefore we use the matrix $\mathbf{K}$ defined in eq. (74). Finally, notice that we can compute the ℓ-th derivative of $s_m(u,v)$ by simply iterating ℓ times the procedure previous used for the first order derivative, with a slight modification of the matrices involved. In fact, the size of the matrices need to shrink at each derivation step (in fact, the degree of a spline determines the length of its coefficient vector). Therefore, by indexing each matrix with a subscript corresponding to the order of the derivative, we obtain for the ℓ-th order derivative (similar to eq. (87)):

$$
\overline{\mathbf{b}}^{(\ell)} = \mathbf{S}_\ell \overline{\mathbf{b}} = \left[\prod_{h=1}^{\ell} \mathbf{D}_h \left[\mathbf{E}_h \mathbf{T}_h^f - \mathbf{F}_h \mathbf{T}_h^l \right] \mathbf{K}_h \right] \overline{\mathbf{b}} \,. \qquad (97)
$$

All the definitions are provided below in eq. (98), (100), (101), (102), (103). Notice however that they are just simple generalisations of the matrices used when we dealing with the integral constraint: in fact in that case we were considering a first order derivative, while here we are considering a ℓ-th order derivative. The only significant difference consists in the substitution of the matrix $\mathbf{K}$ with the difference operator matrix $\mathbf{K}_h$ defined in eq. (100). Let $\{\mathbf{e}_{-m+h}, \ldots, \mathbf{e}_g\}$ be the canonical basis of the space of $(g+m+1-h) \times (g+m+1-h)$ matrices, with $h = 1, \ldots, \ell$. The block diagonal $(g+m+1-h)^2 \times (g+m+1-h)^2$ matrix $\mathbf{D}_h$ is given by:

$$
\mathbf{D}_h = \sum_{i=-m+h}^{g} \mathbf{e}_i \mathbf{e}_i' \otimes \mathbf{D}_h^i = \left[\begin{array}{c|c|c|c} \mathbf{D}_h^{-m+h} & & & \\ \hline & \mathbf{D}_h^{-m+h+1} & & \\ \hline & & \ddots & \\ \hline & & & \mathbf{D}_h^g \end{array} \right] , \qquad (98)
$$

where each block $\mathbf{D}_h^j$, for $j = -m+h, \ldots, g$, $h = 1, \ldots, \ell$, is a $(g+m+1-h) \times (g+m+1-h)$ diagonal matrix:

$$
\mathbf{D}_h^j = \mathrm{diag} \left(\frac{(m+1-h)^2}{\lambda_{-m+h,j+m+1-h} - \lambda_{-m+h,j}}, \frac{(m+1-h)^2}{\lambda_{-m+h+1,j+m+1-h} - \lambda_{-m+h+1,j}}, \ldots, \frac{(m+1-h)^2}{\lambda_{g,j+m+1-h} - \lambda_{g,j}} \right) . \qquad (99)
$$

Notice that when $\ell = 1$ and the original spline has degree $m+1$ we are back in the previous case. As previously noted, in dealing with derivatives without constraints, the matrix $\mathbf{K}$ can be substituted by the difference operator matrix defined in eq. (80). It has the same structure and entries for each $h = 1, \ldots, \ell$, but with different size. Since we are dealing with the iterative vectorisation of a matrix, we must account that at each derivative the last row of the original

matrix is lost due to differentiation. This is reflected in a reduction of the number of rows $\mathbf{K}_h$ by a factor of hN at each step, where N is the number of columns of the original matrix. Finally, since differencing is performed iteratively, the vector to be differenced is the outcome of the previous iteration, hence its length (which is equal to the number of columns of $\mathbf{K}_h$) corresponds to the number of rows of $\mathbf{K}_h$ plus hN. Summarizing, for $h = 1, \ldots, \ell$ we define the $(g+m+1)(g+m+1-h) \times (g+m+1)(g+m+2-h)$ matrix:

$$\mathbf{K}_h = \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix} . \tag{100}$$

Similarly, she selection matrices $\mathbf{T}_h^f, \mathbf{T}_h^l$, with $h = 1, \ldots, \ell$, have the structure as $\mathbf{T}^f, \mathbf{T}^l$, but their size is $(g+m+1-h)^2 \times (g+m+1)(g+m+1-h)$. They are defined as follows:

$$\mathbf{T}_h^f = \left[\begin{array}{ccc|ccc} 1 & & & 0 & \ldots & 0 \\ & \ddots & & \vdots & & \vdots \\ & & 1 & 0 & \ldots & 0 \end{array}\right] \qquad \mathbf{T}_h^l = \left[\begin{array}{ccc|ccc} 0 & \ldots & 0 & 1 & & \\ \vdots & & \vdots & & \ddots & \\ 0 & \ldots & 0 & & & 1 \end{array}\right] . \tag{101}$$

Finally, the matrices $\mathbf{E}$ and $\mathbf{F}$ are generalized to obtain the $(g+m+1-h)^2 \times (g+m+1-h)^2$ block diagonal matrices:

$$\mathbf{E}_h = \sum_{i=-m+h}^{g} \mathbf{e}_i \mathbf{e}_i' \otimes \mathbf{E}_h^i = \left[\begin{array}{c|c|c|c} \mathbf{E}_h^{-m} & & & \\ \hline & \mathbf{E}_h^{-m-1} & & \\ \hline & & \ddots & \\ \hline & & & \mathbf{E}_h^{g} \end{array}\right] , \tag{102}$$

$$\mathbf{F}_h = \sum_{i=-m+h}^{g} \mathbf{e}_i \mathbf{e}_i' \otimes \mathbf{E}_h^{i-1} = \left[\begin{array}{c|c|c|c} \mathbf{E}_h^{-m-1} & & & \\ \hline & \mathbf{E}_h^{-m} & & \\ \hline & & \ddots & \\ \hline & & & \mathbf{E}_h^{g-1} \end{array}\right] , \tag{103}$$

where each $\mathbf{E}_h^j$, with $j = -m-1+h, \ldots, g$, $h = 1, \ldots, \ell$, is the $(g+m+1-h) \times (g+m+1-h)$ diagonal matrix:

$$\mathbf{E}_h^j = \operatorname{diag}\left(\frac{1}{\lambda_{1,j} - \lambda_{-m+h,j}}, \frac{1}{\lambda_{2,j} - \lambda_{-m+h+1,j}}, \ldots, \frac{1}{\lambda_{g+m+1-h,j} - \lambda_{g,j}} \right) . \tag{104}$$

To sum up, we can re-write the first addendum of the objective function of the optimization problem in eq. (93) in compact form as follows:

$$J_\ell^1(\overline{\mathbf{b}}) = \overline{\mathbf{b}}^{(\ell)'} \mathbf{M}_{m,\ell} \overline{\mathbf{b}}^{(\ell)} = \overline{\mathbf{b}}' \mathbf{S}_\ell' \mathbf{M}_{m,\ell} \mathbf{S}_\ell \overline{\mathbf{b}} . \tag{105}$$

Putting together eq. (105) and (94) we obtain the following matrix representation of the objective function of the optimisation problem in eq. (23):

$$J_\ell(\overline{\mathbf{b}}) = \overline{\mathbf{b}}' \mathbf{S}_\ell' \mathbf{M}_{m,\ell} \mathbf{S}_\ell \overline{\mathbf{b}} + \alpha \left(\mathbf{z} - \mathbf{C}^{m+1}(\mathbf{u}, \mathbf{v}) \overline{\mathbf{b}} \right)' \mathbf{W} \left(\mathbf{z} - \mathbf{C}^{m+1}(\mathbf{u}, \mathbf{v}) \overline{\mathbf{b}} \right) . \tag{106}$$

We can now exploit the linear relation obtained in eq. (87) by working out the integral constraint and substitute it in eq. (106). This transforms the constrained optimization problem in eq. (23) into an unconstrained optimisation problem for $\tilde{\mathbf{c}}$, with objective function:

$$J_\ell(\tilde{\mathbf{c}}) = \tilde{\mathbf{c}}'\mathbf{A}'\mathbf{D}'\mathbf{S}_\ell'\mathbf{M}_{m,\ell}\mathbf{S}_\ell\mathbf{D}\mathbf{A}\tilde{\mathbf{c}} + \alpha\left(\mathbf{z} - \mathbf{C}^{m+1}(\mathbf{u},\mathbf{v})\mathbf{D}\mathbf{A}\tilde{\mathbf{c}}\right)'\mathbf{W}\left(\mathbf{z} - \mathbf{C}^{m+1}(\mathbf{u},\mathbf{v})\mathbf{D}\mathbf{A}\tilde{\mathbf{c}}\right) . \quad (107)$$

The system of first order necessary conditions for an optimum is obtained from:

$$\begin{aligned}\frac{\mathrm{d}J_\ell(\tilde{\mathbf{c}})}{\mathrm{d}\tilde{\mathbf{c}}'} &= 2\mathbf{A}'\mathbf{D}'\mathbf{S}_\ell'\mathbf{M}_{m,\ell}\mathbf{S}_\ell\mathbf{D}\mathbf{A}\tilde{\mathbf{c}} - 2\alpha\mathbf{A}'\mathbf{D}'\mathbf{C}^{m+1}(\mathbf{u},\mathbf{v})'\mathbf{W}\mathbf{z} \\ &\quad + 2\alpha\mathbf{A}'\mathbf{D}'\mathbf{C}^{m+1}(\mathbf{u},\mathbf{v})'\mathbf{W}\mathbf{C}^{m+1}(\mathbf{u},\mathbf{v})\mathbf{D}\mathbf{A}\tilde{\mathbf{c}} = 0 .\end{aligned} \quad (108)$$

Define the following variables for easing the notation:

$$\begin{aligned}\mathbf{N}_{m,\ell} &= \mathbf{A}'\mathbf{D}'\mathbf{S}_\ell'\mathbf{M}_{m,\ell}\mathbf{S}_\ell\mathbf{D}\mathbf{A} , \\ \mathbf{H}(\mathbf{u},\mathbf{v}) &= \mathbf{C}^{m+1}(\mathbf{u},\mathbf{v})\mathbf{D}\mathbf{A} .\end{aligned}$$

Therefore, one gets:

$$\begin{aligned}\mathbf{N}_{m,\ell}\tilde{\mathbf{c}} + \alpha\mathbf{H}(\mathbf{u},\mathbf{v})'\mathbf{W}\mathbf{H}(\mathbf{u},\mathbf{v})\tilde{\mathbf{c}} &= \alpha\mathbf{H}(\mathbf{u},\mathbf{v})'\mathbf{W}\mathbf{z} \\ \left[\mathbf{N}_{m,\ell} + \alpha\mathbf{H}(\mathbf{u},\mathbf{v})'\mathbf{W}\mathbf{H}(\mathbf{u},\mathbf{v})\right]\tilde{\mathbf{c}} &= \alpha\mathbf{H}(\mathbf{u},\mathbf{v})'\mathbf{W}\mathbf{z} .\end{aligned} \quad (109)$$

If the condition of the Rouché-Capelli theorem for the system to admit solution is satisfied and the matrix $\left[\mathbf{N}_{m,\ell} + \alpha\mathbf{H}(\mathbf{u},\mathbf{v})'\mathbf{W}\mathbf{H}(\mathbf{u},\mathbf{v})\right]$ has full rank, then the system has a unique solution $\tilde{\mathbf{c}}^*$. By contrast, if the matrix is singular the problem admits an infinite number of solutions which can be obtained by computing the Moore-Penrose pseudo-inverse (denoted by †):

$$\tilde{\mathbf{c}}^* = \alpha\left[\mathbf{N}_{m,\ell} + \alpha\mathbf{H}(\mathbf{u},\mathbf{v})'\mathbf{W}\mathbf{H}(\mathbf{u},\mathbf{v})\right]^\dagger\mathbf{H}(\mathbf{u},\mathbf{v})'\mathbf{W}\mathbf{z} , \quad (110)$$

with a slight abuse of notation. Among this set of solutions, we choose the one with smallest norm. As a final step, we use eq. (87) and plug-in the optimal value of $\tilde{\mathbf{c}}$ for obtaining the optimal value of the coefficients $\overline{\mathbf{b}}^*$:

$$\overline{\mathbf{b}}^* = \mathbf{D}\mathbf{A}\tilde{\mathbf{c}}^* . \quad (111)$$

B.2 Eigenproblem

In the following we show the computations required for obtaining eq. (29). In this section, differently from the previous ones, we explicitly denote all the arguments of a function for making notation clearer. Thus, for example, we have that $\breve{f}_t(\cdot) = \breve{f}_t(\cdot,\cdot)$, where the first is follows the notation of the previous sections, while the second is according to the notation of this section. We use the standard estimator for the sample covariance (alternative non-parametric estimatiors have been proposed by Hall et al. (2006), Li and Hsing (2010), Yao et al. (2005) and Staniswalis and Lee (1998)), that is:

$$v(l_1,m_1,l_2,m_2) = \frac{1}{T}\sum_{t=1}^{T}\breve{f}_t(l_1,m_1)\breve{f}_t(l_2,m_2) . \quad (112)$$

Then, the eigenproblem can be formulated as follows, for $j = 1,2,\ldots$:

$$\int_{a_1}^{b_1}\int_{a_2}^{b_2} v(\cdot,\cdot,l_2,m_2)\xi_j(l_2,m_2)\ \mathrm{d}l_2\,\mathrm{d}m_2 = \rho_j\xi_j(\cdot,\cdot) . \quad (113)$$

In order to solve this problem, we choose to express the eigenfunctions as finite linear combinations of the same set of basis functions used for the functions $\breve{f}_t(\cdot,\cdot)$, that is the basis B-spline functions $\boldsymbol{\psi}(\cdot,\cdot) = (\psi_1(\cdot,\cdot),\ldots,\psi_K(\cdot,\cdot))'$ in eq. (26). Define the coefficient vectors $\mathbf{a}_j = (a_{j,1},\ldots,a_{j,K})'$. To summarize, we have:

$$\breve{f}_t(\cdot,\cdot) = \mathbf{d}_t'\boldsymbol{\psi}(\cdot,\cdot) = \sum_{k=1}^{K} d_{t,k}\psi_k(\cdot,\cdot)\,, \tag{114}$$

$$\xi_j(\cdot,\cdot) = \mathbf{a}_j'\boldsymbol{\psi}(\cdot,\cdot) = \sum_{k=1}^{K} a_{j,k}\psi_k(\cdot,\cdot)\,. \tag{115}$$

By stacking all data together in $\breve{\mathbf{f}}(\cdot,\cdot) = (\breve{f}_1(\cdot,\cdot),\ldots,\breve{f}_T(\cdot,\cdot))'$ and $\mathbf{D} = (\mathbf{d}_1,\ldots,\mathbf{d}_T)$, we obtain:

$$\mathbf{f}(\cdot,\cdot) = \mathbf{D}\boldsymbol{\psi}(\cdot,\cdot)\,. \tag{116}$$

We can thus rewrite eq. (113) in matrix notation:

$$\frac{1}{T}\int_a^b\int_a^b \boldsymbol{\psi}(\cdot,\cdot)'\mathbf{D}'\mathbf{D}\boldsymbol{\psi}(l_2,m_2)\boldsymbol{\psi}(l_2,m_2)'\mathbf{a}_j \;\, \mathrm{d}l_2\,\mathrm{d}m_2 = \rho_j\boldsymbol{\psi}(\cdot,\cdot)'\mathbf{a}_j \tag{117}$$

$$\frac{1}{T}\boldsymbol{\psi}(\cdot,\cdot)'\mathbf{D}'\mathbf{D}\left[\int_a^b\int_a^b \boldsymbol{\psi}(l_2,m_2)\boldsymbol{\psi}(l_2,m_2)' \;\, \mathrm{d}l_2\,\mathrm{d}m_2\right]\mathbf{a}_j = \rho_j\boldsymbol{\psi}(\cdot,\cdot)'\mathbf{a}_j\,, \tag{118}$$

then define the matrix of inner products:

$$\mathbf{M} = \int_a^b\int_a^b \boldsymbol{\psi}(l_2,m_2)\boldsymbol{\psi}(l_2,m_2)' \;\, \mathrm{d}l_2\,\mathrm{d}m_2 = \begin{bmatrix} \langle\psi_1(\cdot,\cdot),\psi_1(\cdot,\cdot)\rangle & \ldots & \langle\psi_1(\cdot,\cdot),\psi_K(\cdot,\cdot)\rangle \\ \vdots & & \vdots \\ \langle\psi_K(\cdot,\cdot),\psi_1(\cdot,\cdot)\rangle & \ldots & \langle\psi_K(\cdot,\cdot),\psi_K(\cdot,\cdot)\rangle \end{bmatrix} \tag{119}$$

$$\langle\psi_i(\cdot,\cdot),\psi_j(\cdot,\cdot)\rangle = \int_a^b\int_a^b \psi_i(l,m)\psi_j(l,m) \;\, \mathrm{d}l\,\mathrm{d}m\,, \tag{120}$$

thus obtaining:

$$T^{-1}\mathbf{D}'\mathbf{D}\mathbf{M}\mathbf{a}_j = \rho_j\mathbf{a}_j\,. \tag{121}$$

In order to obtain a positive semi-definite matrix, we apply the linear transformation $\mathbf{u}_j = \mathbf{M}^{1/2}\mathbf{a}_j$, where $\mathbf{A}^{1/2}$ is the principal square root of the positive definite matrix $\mathbf{A}$. Then, rewrite the previous equation as an eigenproblem for $\mathbf{u}_j$ as follows:

$$T^{-1}\mathbf{M}^{1/2}\mathbf{D}'\mathbf{D}\mathbf{M}^{1/2}\mathbf{u}_j = \rho_j\mathbf{u}_j\,, \tag{122}$$

which is a standard multivariate eigenproblem for the matrix $T^{-1}\mathbf{M}^{1/2}\mathbf{D}'\mathbf{D}\mathbf{M}^{1/2}$. The number of components to take, J, is determined by the fraction of variability explained: we sort the estimated eigenvalues $\widehat{\rho}_j$, for $j = 1,2,\ldots$, in decreasing order. Then, we fix a threshold $\bar{d}$ and retain all the pairs of eigenvalues and eigenvectors until the corresponding cumulated proportion of explained variability reaches, that is $J = \arg\min_j\{\sum_j \widehat{\rho}_j \geq \bar{d}\}$.

C Additional plots

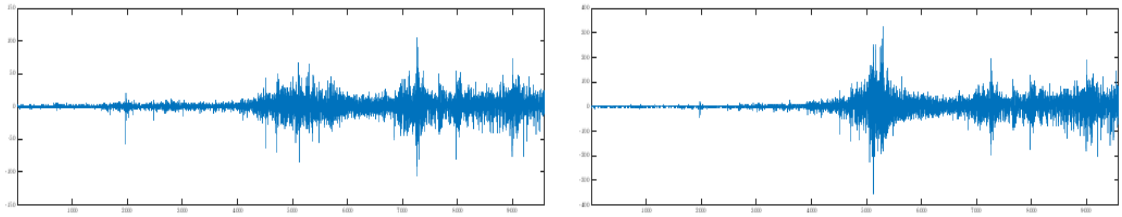

Figure 6: First differenced series of S&P500 (*left*) and NASDAQ (*right*).

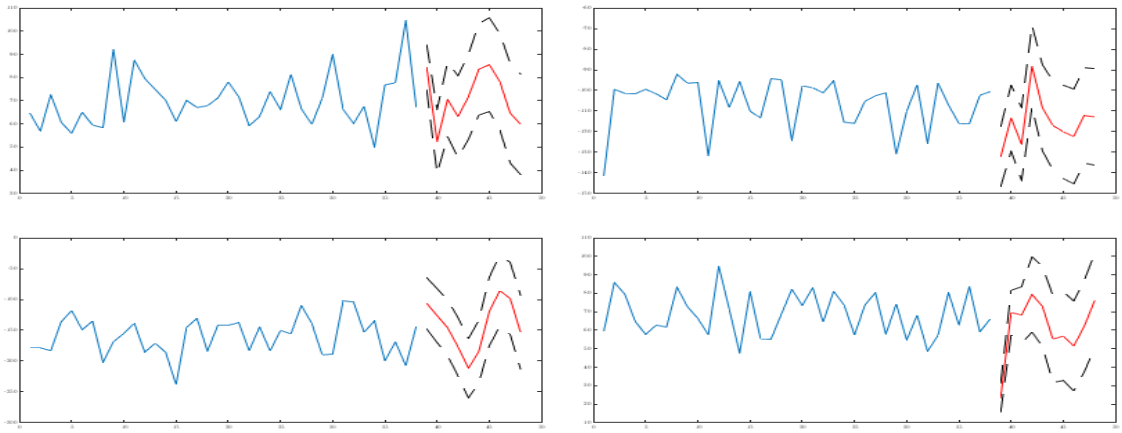

Figure 7: Estimated time series (*solid, blue*) and forecast (*solid, red*) with 95% confidence intervals (*dashed, black*) of each entry of the vector of fPCA scores $\{\widehat{\boldsymbol{\beta}}_t\}_t$,from $j = 1$ (*top left*) to $j = J$ (*bottom*).

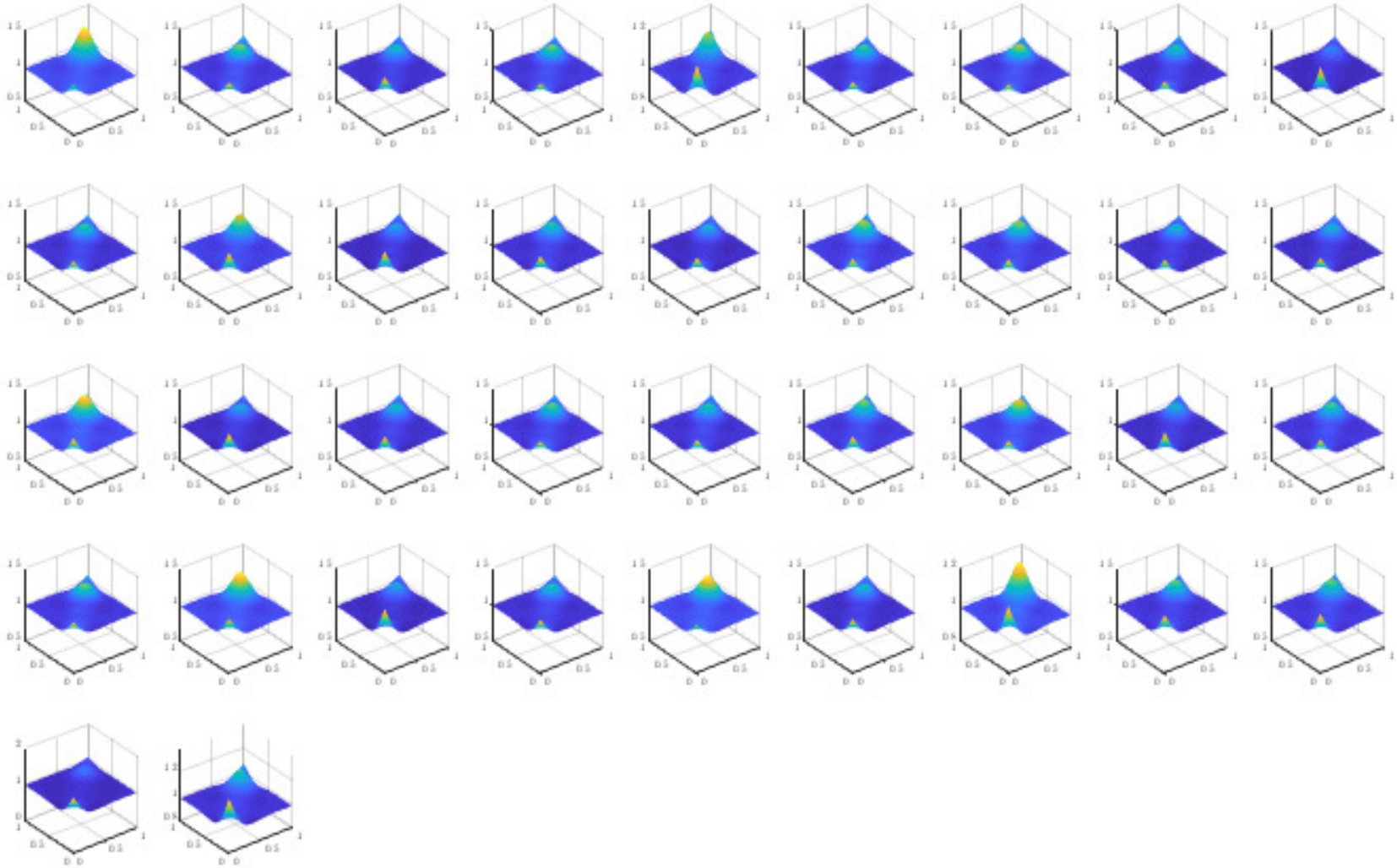

Figure 8: 3D density plot of time series of bivariate copula pdfs, approximated via fPCA, for each year $t = 1, \ldots, T$, starting from $t = 1$ in the top-left panel.