

Uncertainty quantification for functional dependent random variables

Simon Nanty, Céline Helbert, Amandine Marrel, Nadia Pérot, Clémentine Prieur

▶ **To cite this version:**

Simon Nanty, Céline Helbert, Amandine Marrel, Nadia Pérot, Clémentine Prieur. Uncertainty quantification for functional dependent random variables. Computational Statistics, 2017, 32 (2), pp.559-583. 10.1007/s00180-016-0676-0 . hal-01075840v2

HAL Id: hal-01075840
https://hal.science/hal-01075840v2

Submitted on 28 Jul 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Uncertainty quantification for functional dependent random variables

Simon Nanty[1,3], Céline Helbert[2], Amandine Marrel[1], Nadia Pérot[1], and Clémentine Prieur[3]

[1]CEA, DEN, F-13108, Saint-Paul-lez-Durance, France
[2]Université de Lyon, UMR 5208, Ecole Centrale de Lyon, Institut Camille Jordan
[3]Université Joseph Fourier and INRIA, Grenoble, France

Abstract

This paper proposes a new methodology to model uncertainties associated with functional random variables. This methodology allows to deal simultaneously with several dependent functional variables and to address the specific case where these variables are linked to a vectorial variable, called covariate. In this case, the proposed uncertainty modelling methodology has two objectives: to retain both the most important features of the functional variables and their features which are the most correlated to the covariate. This methodology is composed of two steps. First, the functional variables are decomposed on a functional basis. To deal simultaneously with several dependent functional variables, a Simultaneous Partial Least Squares algorithm is proposed to estimate this basis. Second, the joint probability density function of the coefficients selected in the decomposition is modelled by a Gaussian mixture model. A new sparse method based on a Lasso penalization algorithm is proposed to estimate the Gaussian mixture model parameters and reduce their number. Several criteria are introduced to assess the methodology performance: its ability to approximate the functional variables probability distribution, their dependence structure and their features which explain the covariate. Finally, the whole methodology is applied on a simulated example and on a nuclear reliability test case.

1 Introduction

In a large number of fields, like physical or environmental sciences, computer codes or numerical simulators have proved to be an invaluable tool to model and predict phenomena. To describe the features of the studied phenomenon, the simulator computes several output parameters of interest from numerous explanatory input parameters. All these output and input variables can be of various types: scalar, functional, categorical, etc. Furthermore, the knowledge of the phenomenon features is often limited and imprecise, so that the parameters used to describe them in the model are uncertain. These uncertainties of the parameters of the computer model can be due to measurement errors, the intrinsic variability of the phenomenon, model errors, numerical errors, etc. The characterization or modelling of the uncertainties related to these parameters is an important issue in the study and the treatment of uncertainties in computer codes (De Rocquigny et al., 2008; Helton et al., 2006). This characterization is moreover necessary to a better understanding of the behaviour of the computer code and more generally the modelled phenomenon.

In this work, the studied uncertain parameters are functional random variables (or stochastic processes) which can be inputs or outputs of a computer code. Functional data analysis is a topic which has been studied in many references among which one can cite books by Ferraty and Vieu (2006); Horváth and Kokoszka (2012); Bongiorno et al. (2014) or recent publication by Goia and Vieu (2016) and references therein. The objective of the present work is to characterize their uncertainties, *i.e.* to model their joint probability distribution. Furthermore, it must be possible to apply the methodology proposed in this paper regardless of the way functional data have been obtained: if data come from experimental

measurements and are inputs of a model or if they are outputs of a deterministic or stochastic computer code. For instance, in the case where they are outputs of a computer code, no information about the code or its input parameters is used. The uncertainty modelling of a single functional variable has been thoroughly studied in several different contexts. For instance, in the study of stochastic partial differential equations, functional uncertain parameters can be used in boundary or initial conditions of the studied equations. In this context, Ghanem and Spanos (1991) have proposed to conduct the study in two steps. First, the Karhunen-Loève expansion (Loève, 1955), of which is derived Principal Component Analysis (PCA), is used to decompose the variable on a functional basis and thus to reduce the dimension of the problem. In a second step, the random coefficients of the variable on the functional basis are modelled by a polynomial chaos expansion. This very common approach has been explored and improved in many works (Ma and Zabaras 2011; Wan and Zabaras 2014; etc.). In the context of sensitivity analysis, Anstett-Collin et al. (2015) consider that the functional variables in input of the computer code are Gaussian processes, and approximate them by a truncation of the Karhunen-Loève expansion. As the variables are assumed to be Gaussian processes, their coefficients in the Karhunen-Loève expansion are independent and normally distributed. The authors then conduct a sensitivity analysis on the output of the computer code, replacing the functional inputs by their coefficients in the truncated Karhunen-Loève expansion. Hyndman and Shang (2010) propose a visualization tool for functional data, whose first step consists in decomposing the functional data on the two first components of a functional PCA basis (Ramsay and Silverman, 2005), and in computing the joint probability density function of the couple of coefficients by a kernel density estimation procedure (Scott, 2009). In the same spirit, Bongiorno and Goia (2016, 2015) use a decomposition of the functional variables, *via* the Karhunen-Loève expansion, and estimate the joint distribution of the coefficients of the truncated expansion using a kernel density estimator. However, their approach is used in a classification context. They exploit the link between the small-ball probability associated with the functional data and the joint probability density function of the coefficients, stated in Proposition 1 of Bongiorno and Goia (2016) and discussed in Delaigle and Hall (2010). The functional variables uncertainty modelling is then used in their visualization tool. The approach followed by most of these authors can thus be separated into two main steps: the decomposition of the functional variable on a functional basis and the estimation of the joint probability density function associated with the first coefficients of the decomposition.

In this paper, we want to address two additional problems. The first studied issue is to deal simultaneously with several dependent functional variables. The methodology presented in this paper is designed to characterize simultaneously several functional variables, and thus to take into account their dependence. This issue has previously been adressed in Jacques and Preda (2014a,b), in a classification context. In their work, dependent functional variables are decomposed on a basis using an extension of PCA, and Gaussian mixture modelling is used to cluster the decomposition coefficients. In addition to characterizing the uncertainty of functional variables, we also propose to address the case where these random variables can be linked to a scalar or vectorial variable, called hereafter covariate. The covariate is thus here defined as a variable dependent on the functional variables under study. The functional variables and the covariate are thus correlated. For example, the covariate can be the output of a code taking as inputs the functional variables. The objective of uncertainty modelling is usually to find the best approximation of the probability distribution of the functional variables. However, the modelling cannot be perfect and only a part of the characteristics of the functional variables is kept in the model, so that some of the functional variable features which explain the covariate can be lost in the modelling. In the case where functional variables are linked to a covariate, the modelling of the functional variables has to retain both their most important characteristics and their characteristics which are the most correlated to the covariate. The aim is not here to propagate uncertainties to the covariate nor to model the link between the functional variables and the covariate, but only to preserve, in the estimated probability distribution of the functional data, the features which best explain the covariate. Returning to the example where the covariate is the output of a computer code which takes as inputs the functional variables, their estimated probability distribution could be used, in a second step, to generate more realizations of the variables and conduct a sensitivity analysis of the computer code. Other applications of this characterization methodology could be the estimation of probabilities related to the functional variables and the covariate, such as the joint probability for the functional variables and the covariate to exceed some thresholds. The computation of such probabilities is illustrated in section 5.1. The methodology could be also used to build a visualization tool for the functional variables following the method proposed by Hyndman and Shang (2010). This possible application is illustrated in section 5.2.

The proposed uncertainty modelling methodology is composed of two main steps, as in some of the previously presented methods, such as Anstett-Collin et al. (2015). First, the dimension of the problem is reduced by decomposing the functional random variables on a functional basis. If the studied variables are not linked to a covariate, we propose to use Functional Principal Component Analysis (Ramsay and Silverman, 2005). Alternatively, in the presence of a covariate linked to the studied variables, the Partial Least Squares (PLS) decomposition, based on Partial Least Squares regression (Wold, 1966), is proposed to reach a compromise between retaining the most important features of the functional variables and their features which are the most correlated to the covariate. In order to take into account the dependence between the functional random variables, a simultaneous version of PLS decomposition, denoted SPLS, is developed. This means that the decomposition is done on a vector of functional random variables instead of a unique functional random variable. From the SPLS decomposition, a finite number of coefficients are selected to approximate the functional variables; the problem becoming multivariate. The second step of our methodology consists in estimating the joint probability density function of the decomposition coefficients. For this, a Gaussian mixture model is proposed. A new estimation method has been developed to learn sparse Gaussian mixture models in order to reduce the number of parameters in the model. This procedure combines the well-known Expectation-Maximization algorithm (Dempster et al., 1977) and a Lasso penalization-based algorithm for sparse covariance matrices estimation (Bien and Tibshirani, 2011). Several criteria are proposed in this paper to check the efficiency of the methodology. Their objectives are to assess its ability to approximate the probability distribution of the functional variables, to capture their dependence structure or their features related to the covariate.

In the next two sections, the methodology to characterize the uncertainty of dependent functional random variables is fully described. Two proposed dimension reduction methods based on functional principal component analysis and Partial Least Squares regression are presented in section 2. The density estimation step is detailed in section 3. In section 4, criteria chosen to adjust the parameters of the developed methodology and to assess its quality are presented. Tests of the methodology are run on an simulated example in section 5.1, and an application to a nuclear reliability test case is proposed in section 5.2.

2 Functional decomposition

Let us define the probability space $(\Omega, \mathcal{F}, P)$. The functional random variables $f_1, \dots, f_m : \Omega \times I \to \mathbb{R}$, with $I \subset \mathbb{R}$, are defined on this probability space and take their values in the Hilbert space $\mathcal{L}^2([0,1])$ endowed with canonical inner product and associated norm. $f_i(\omega, .) : I \to \mathbb{R}$, for $i \in \{1, \dots, m\}$ and $\omega \in \Omega$, is thus a one-dimensional function. These variables are the inputs of the computer code $\mathcal{M}$. The output of $\mathcal{M}$ is the scalar variable Y, called hereafter a covariate. In the following, it is considered that a sample of n vectors of m functions $f_{1,j}, \dots, f_{m,j}$, $j \in \{1, \dots, n\}$ is known. The corresponding outputs of $\mathcal{M}$, $y_j = \mathcal{M}(f_{1,j}, \dots, f_{m,j})$, are also known. The functions are discretized on the points $t_1, \dots, t_p$ of the interval I. The discretized version of the function $f_{i,j}$ is noted $\mathbf{f}_{i,j} \in \mathbb{R}^p$, $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, n\}$, such that $f_{i,j}(t_k) = \mathbf{f}_{i,j,k}$, for $k \in \{1, \dots, p\}$.

The objective of this section is to approximate simultaneously the m functional random variables $f_1, \dots, f_m$ on a basis. The decomposition of a single functional random variable f_i, for $i \in \{1, \dots, m\}$, is first presented. The sample functions $f_{i,1}, \dots, f_{i,n}$ are approximated on a truncated basis $\left(\varphi_1^{(i)}, \dots, \varphi_d^{(i)}\right)$ of size $d \in \mathbb{N}$:

$$f_{i,j}(t) \approx e^{(i)}(t) + \sum_{k=1}^{d} \alpha_{j,k}^{(i)} \varphi_k^{(i)}(t), \tag{1}$$

with $t \in \mathbb{R}$, $i \in \{1, \dots, m\}$, $j \in \{1, \dots, n\}$, $e^{(i)} = \frac{1}{n} \sum_{j=1}^{n} f_{i,j}$ is the mean function and $\alpha_{j,k}^{(i)}$ is the coefficient of the j^{th} curve on the k^{th} component. Two decompositions have been investigated: simultaneous Principal Components Analysis (SPCA) and simultaneous Partial Least Squares decomposition (SPLS).

2.1 Principal Components Analysis decomposition

The functional principal components analysis (FPCA) method, proposed by Ramsay and Silverman (2005), is an adaptation of Principal Component Analysis (PCA), first proposed by Pearson (1901). For

a sample of functions $(f_{i,j})_{1\leqslant j\leqslant n}$, FPCA searches for the basis functions $\varphi_1^{(i)},\dots,\varphi_d^{(i)}$ and the coefficients $\alpha_{j,k}^{(i)}$, $j\in\{1,\dots,n\}$, $k\in\{1,\dots,d\}$ that minimize

$$\sum_{j=1}^{n}\int_I\left(f_{i,j}(t)-e^{(i)}(t)-\sum_{k=1}^{d}\alpha_{j,k}^{(i)}\varphi_k^{(i)}(t)\right)^2dt,$$

such that the functions $\varphi_1^{(i)},\dots,\varphi_d^{(i)}$ are orthonormal. In practice, different approaches exist to solve this optimization problem. Ramsay and Silverman (2005) proposes to expand the functions as linear combinations of spline basis functions. Then PCA can be applied to the coefficients of the functions on the spline basis. In particular, one can use a linear spline basis which is equivalent to apply PCA directly to the discretized functions. This method is applied here. $F^{(i)}$ is the matrix of the n discretized functions such that $F_{k,j}^{(i)}=\mathbf{f}_{i,j,k}$. The PCA decomposition is found by singular value decomposition of the matrix $F^{(i)}$:

$$F^{(i)}=U^{(i)}D^{(i)}{V^{(i)}}^T,$$

where $U^{(i)}$ and $V^{(i)}$ are orthogonal matrices and $D^{(i)}$ is diagonal. The functions $\varphi_k^{(i)}$ are the columns of $V^{(i)}$.

Van Deun et al. (2009) and Ramsay and Silverman (2005) propose to decompose simultaneously multivariate functional data as a single FPCA basis to handle the dependence between the functional random variables. In the following, this method is called SPCA. To this end, the PCA decomposition is applied to the vectors $\mathbf{f}_j$ of concatenated discretized functions, such that

$$\mathbf{f}_j=[\mathbf{f}_{1,j}/N_1,\dots,\mathbf{f}_{m,j}/N_m]\in\mathbb{R}^{mp},\ j\in\{1,\dots,n\},$$

where $N_1,\dots,N_m$ are normalization factors. Moreover, if the curves f_i are correlated, this simultaneous decomposition is hoped to help reducing the number of components. For the same number of components, simultaneous decomposition can, in some cases, give a better approximation than decompositions on each functional random variable independently. The choice of the normalization factors is important, as it must ensure that each functional random variable has an equivalent influence on the decomposition. Three normalization factors are proposed here and compared in section 5.1:

- the maximum of the functional random variable: $N_i=\max\limits_{\substack{1\leqslant j\leqslant n\\1\leqslant k\leqslant p}}\mathbf{f}_{i,j,k}$,
- the sum of the standard deviations at each time step k: $N_i=\sum\limits_{k=1}^{p}\sqrt{\mathrm{Var}(\mathbf{f}_{i,.,k})}$,
- the square root of the sum of the variances at each time step: $N_i=\left(\sum\limits_{k=1}^{p}\mathrm{Var}(\mathbf{f}_{i,.,k})\right)^{1/2}$.

2.2 Partial Least Squares decomposition

The second considered decomposition basis, the Partial Least Squares (PLS) decomposition, is also built from the available data, and is based on the PLS regression technique, proposed by Wold (1966). Compared to PCA decomposition, PLS decomposition can take into account the link between the functional random variables and a vectorial covariate. The PLS decomposition is here applied to the discretized version of the functional data to be decomposed. A detailed description of PLS regression and decomposition can be found in Höskuldsson (1988). The aim of PLS regression is to explain the variable Y with linear combinations of the variables $X_1,\dots,X_p$, where the variables $X_1,\dots,X_p$ are standardized and centered. Let us define the samples of n realizations $Y_1,\dots,Y_n$ and $X_{i,1},\dots,X_{i,n}$ for $i\in\{1,\dots,p\}$. The PLS algorithm is initialized to $X_0=X$, the matrix whose column vectors are $(X_{1,1},\dots,X_{1,n})^T,\dots,(X_{p,1},\dots,X_{p,n})^T$. At each step $h>0$, the vector u_h of weights for the linear combination solves the following equation:

$$\max_{\|u_h\|=1}\mathrm{Cov}(X_{h-1}u_h,Y).$$

The h^{th} predictor of the regression is defined as $\alpha_h = X_{h-1}u_h$, with u_h the solution of the previous optimization problem. Finally, the matrix X_h is the so-called deflation of X_{h-1}: $X_h = X_{h-1} - \alpha_h \varphi_h^T$, where the vector φ_h is defined as follows:

$$\varphi_h = \frac{X_{h-1}^T \alpha_h}{\alpha_h^T \alpha_h}.$$

This procedure is repeated for each step h from 1 to d.

To derive the PLS decomposition of f_i, $i = 1, \ldots, m$, this regression technique is applied to the matrix X, such that the elements $X_{j,k} = \mathbf{f}_{i,j,k}$, $j = 1, \ldots, n$ and $k = 1, \ldots, p$. d steps are computed. Then, for $i = 1, \ldots, m$ and for $j = 1, \ldots, n$, the discretized sample functions can be approximated in this way:

$$\mathbf{f}_{i,j} \approx \sum_{h=1}^{d} \alpha_{hj} \varphi_h.$$

Each obtained vector φ_h is then the h^{th} basis function in the PLS decomposition and the h^{th} predictor α_h is the vector of coefficients associated to the h^{th} function basis, for $h = 1, \ldots, d$. As for PCA, the PLS regression can be applied to the concatenated discretized functional random variables, so that these variables are decomposed simultaneously on a PLS basis. This decomposition is called SPLS in the following. No normalization is applied to the variables, as the data is centered and standardized in the PLS algorithm.

The choice of the decomposition depends on the studied case. In the absence of covariate, SPCA is preferable in order to optimize the approximation of the functional variables. On the contrary, SPLS could be a better choice to add information about this covariate.

3 Probability density estimation

The objective of this section is to model the probability density function of coefficients of the decomposition, with number of coefficients d being fixed. In practice, the number of coefficients depends on quality criteria defined in section 4 and whose usage is detailed in section 5.

3.1 Gaussian Mixture model and EM algorithm

Let $\alpha_j = (\alpha_{j,1}, \ldots, \alpha_{j,d})$, $j \in \{1, \ldots, n\}$ be the vectors of coefficients of the decomposition. The density of the sample of vectors $\alpha_1, \ldots, \alpha_n$ is estimated with a Gaussian mixture model (GMM). Let us define G the number of clusters in the mixture, μ_g, Σ_g, $g \in \{1, \ldots, G\}$, the vectors of means and matrices of covariance of the clusters and τ_g the proportions of the clusters in the mixture. The probability density function g of the GMM is written for all $\alpha \in \mathbb{R}^d$,

$$g(\alpha) = \sum_{g=1}^{G} \frac{\tau_g}{\sqrt{\det(2\pi\Sigma_g)}} \exp\left(-(\alpha - \mu_g)^T \Sigma_g^{-1} (\alpha - \mu_g)/2\right). \tag{2}$$

The parameters of the probability density function are estimated by the Expectation-Maximization algorithm (EM), introduced by Dempster et al. (1977). More complete reviews of the EM algorithm can be found in Mclachlan and Krishnan (1997). This algorithm maximizes the likelihood of the model by replacing the data α by the so-called complete data (α, z), where z is called the hidden variable. In the case of GMM, the hidden variables are defined in this way for $i = 1, \ldots, n$ and $g = 1, \ldots, G$:

$$z_{ig} = \begin{cases} 1 & \text{if } \alpha_i \text{ belongs to group } g \\ 0 & \text{otherwise.} \end{cases}$$

The log-likelihood of the complete data is:

$$\begin{aligned} \ell\left(\alpha, z |\, \tau_g, \mu_g, \Sigma_g, g = 1, \ldots, G\right) &= -\frac{np\log(2\pi)}{2} + \\ &\sum_{g=1}^{G}\sum_{i=1}^{n} z_{ig} \log \tau_g - \frac{1}{2}\sum_{g=1}^{G}\sum_{i=1}^{n} z_{ig} \left[\log\det(\Sigma_g) + \right. \\ &\left. (\alpha_i - \mu_g)^T \Sigma_g^{-1} (\alpha_i - \mu_g)\right]. \end{aligned} \tag{3}$$

In the EM algorithm, two steps are repeated until convergence. The Expectation step consists of computing the conditional expectation of the log-likelihood given the actual estimation of the parameters. The Maximization step consists of determining the parameters maximizing the conditional expectation computed in the previous step. Wu (1983) show that, under some regularity conditions, the EM algorithm converges to a local minimum of the log-likelihood. In practice, the minimum reached at the end of the algorithm depends strongly on the initialization of the algorithm. The EM algorithm is therefore repeated with different initializations. In the case of GMM, the Expectation step consists of computing this expression:

$$z_{ig} = \frac{\tau_g f_g(\alpha_i|\theta_g)}{\sum_{k=1}^{G} \tau_k f_k(\alpha_i|\theta_k)}, \tag{4}$$

where $f_g : \alpha \mapsto \frac{\exp\left(-(\alpha-\mu_g)^T \Sigma_g^{-1}(\alpha-\mu_g)/2\right)}{\sqrt{\det(2\pi\Sigma_g)}}$, for $g = 1, \ldots, G$.

For the Maximization step, the three following equations are computed:

$$\tau_g = \frac{1}{n}\sum_{i=1}^{n} z_{ig} \tag{5}$$

$$\mu_g = \frac{\sum_{i=1}^{n} z_{ig}\alpha_i}{\sum_{i=1}^{n} z_{ig}} \tag{6}$$

$$\Sigma_g = \frac{1}{\sum_{i=1}^{n} z_{ig}} \sum_{i=1}^{n} z_{ig}(\alpha_i - \mu_g)(\alpha_i - \mu_g)^T. \tag{7}$$

The EM algorithm for estimating the parameters of a GMM is given in Algorithm 1.

Algorithm 1

1. *Initialize the parameters* $\tau_k^{(0)}$, $\mu_k^{(0)}$ *and* $\Sigma_k^{(0)}$, $k \in \{1, \ldots, G\}$.
2. *Expectation Step: Compute* $z_{ik}^{(j)}$, $k \in \{1, \ldots, G\}$, $i \in \{1, \ldots, n\}$, *using equation (4).*
3. *Maximization Step: Compute* $\tau_k^{(j+1)}$, $\mu_k^{(j+1)}$ *and* $\Sigma_k^{(j+1)}$, $k \in \{1, \ldots, G\}$ *using equations (5), (6) et (7) respectively.*
4. *Repeat 2–3 until convergence.*

The number of clusters G in the Gaussian mixture is not selected by the EM algorithm and must be chosen by the user. Many criteria have been developed to select this quantity. In this work, we consider a widely used information theoretic criteria based on a penalization of the log-likelihood. This criterion, called the Bayesian Information Criterion (BIC), has been introduced by Schwarz (1978) and is defined as follows:

$$\text{BIC} = -2\ell + k \ln n, \tag{8}$$

where ℓ is the log-likelihood of the model, k is the number of parameters and N is the sample size. BIC is computed for models estimated with different numbers of clusters and the number of clusters G which maximizes this criterion is selected.

This criterion can also be used to determine the optimal number of clusters in sparse Gaussian mixture models on which our methodology is based and that are introduced in the next section.

3.2 Sparse Gaussian Mixture estimation

The total number N of GMM parameters increases with the dimension and the number of clusters:

$$N = G - 1 + Gd + G\frac{d(d+1)}{2},$$

because $G-1$ proportions, G mean vectors and G symmetric covariance matrices have to be estimated. There can be overfitting if the number of parameters becomes too high with respect to the number of

data points. To avoid this, it can be interesting to reduce the number of parameters. The idea of the developed method is to estimate a GMM with sparse covariance matrices. In an unpublished article[1], Krishnamurthy has proposed to estimate a GMM with sparse covariance by adding a Lasso penalization on the inverse of the covariance matrices. This algorithm is based on the method of Friedman et al. (2008) to estimate the sparse structure of inverse covariance matrices. However, the penalization of the inverse of a covariance matrix enforces the inverse to be sparse but not necessarily the covariance matrix. A matrix can be sparse whereas its inverse is not.

We propose to follow a scheme close to the one of Krishnamurthy, but applying directly the Lasso penalization on the covariance matrix. For this, we use the method of Bien and Tibshirani (2011) which estimates sparse covariance matrices, by maximizing the penalized log-likelihood.

Instead of maximizing the log-likelihood of the GMM, given in (3), we propose to maximize the penalized log-likelihood. The maximization problem can be defined as follows for each cluster $g = 1, \ldots, G$:

$$\hat{\Sigma}_g = \text{argmax}_{S \in \mathcal{S}_d} \left[-\sum_{i=1}^{n} z_{ig} \left(\log\det(S) + \lambda \|P * S\|_1 + (\alpha_i - \mu_g)^T S^{-1} (\alpha_i - \mu_g) \right) \right]. \quad (9)$$

$\mathcal{S}_d$ is the space of symmetric definite positive $d \times d$ matrices, the symbol $*$ denotes the Hadamard product of two matrices, $\lambda \in \mathbb{R}_+$ is a penalization parameter, the norm $\|.\|_1$ is such that $\|A\|_1 = \sum_{i,j} |A_{ij}|$ and P is the penalization matrix. In Bien and Tibshirani (2011), three penalization matrices P have been proposed such that for $i, j \in \{1, \ldots, n\}$,

$$P_{ij}^{(1)} = 1, \; P_{ij}^{(2)} = 1 - \delta_{ij} \text{ or } P_{ij}^{(3)} = \frac{1 - \delta_{ij}}{|\,(\Sigma_g)_{ij}\,|}, \quad (10)$$

where δ_{ij} is the Kronecker delta which is equal to one when $i = j$ and is null otherwise, and

$$\Sigma_g = \frac{\sum_{i=1}^{n} z_{ig} (\alpha_i - \mu_g)(\alpha_i - \mu_g)^T}{\sum_{i=1}^{n} z_{ig}}$$

is the empirical covariance matrix for group g.

Dividing the maximization problem (9) by $\sum_{i=1}^{n} z_{ig}$, one gets:

$$\begin{aligned} \hat{\Sigma}_g &= \text{argmin}_{S \in \mathcal{S}_d} \left[\log\det(S) + \lambda \|P * S\|_1 + \frac{\sum_{i=1}^{n} z_{ig} (\alpha_i - \mu_g)^T S^{-1} (\alpha_i - \mu_g)}{\sum_{i=1}^{n} z_{ig}} \right] \\ \hat{\Sigma}_g &= \text{argmin}_{S \in \mathcal{S}_d} \log\det(S) + \text{tr}(S^{-1}\Sigma_g) + \lambda \|P * S\|_1. \end{aligned} \quad (11)$$

Bien and Tibshirani (2011) have proposed a method to solve the optimization problem (11). It relies on the fact that the objective function is the sum of a convex function $S \mapsto \text{tr}(S^{-1}\Sigma_g) + \lambda \|P * S\|_1$ and a concave function $S \mapsto \log\det S$. The optimization of such a function is a classical problem and can be solved by Majorization-Minimization algorithm. Wang (2013) proposed a new algorithm based on coordinate descent algorithm to solve (11). According to the results of Wang (2013), this new algorithm is faster and numerically more stable for most cases than the algorithm of Bien and Tibshirani (2011).

The EM algorithm can be thus modified by adding these G penalized problems. At each maximization step, the covariance matrices are estimated as in the EM algorithm by equation (7), and then the matrices are re-estimated by Wang's algorithm. The covariance matrix estimated with (7) can be used as initial value for Wang's algorithm. The proposed algorithm is summarized in Algorithm 2.

Algorithm 2

1. *Initialize the parameters* $\tau_k^{(0)}$, $\mu_k^{(0)}$ *and* $\Sigma_k^{(0)}$, $k \in \{1, \ldots, G\}$.
2. *Expectation Step: Compute* $z_{ik}^{(j)}$, $k \in \{1, \ldots, G\}$, $i \in \{1, \ldots, n\}$, *using equation (4).*
3. *Maximization Step: Compute* $\tau_k^{(j+1)}$, $\mu_k^{(j+1)}$ *and* $\Sigma_k^{(j+1)}$, $k \in \{1, \ldots, G\}$ *using equations (5), (6) and (7) respectively.*

[1] www.cs.cmu.edu/~akshaykr/files/sgmm_paper.pdf

4. $\Sigma_k^{(j+1)} \leftarrow \operatorname{argmin}_{S \in \mathcal{S}_d} \log\det S - \operatorname{tr}(S^{-1}\Sigma_k^{(j+1)}) - \lambda \|P * S\|_1$.

5. Repeat 2–4 until convergence.

The choice of the penalization parameter is important. Bien and Tibshirani (2011) propose to choose it by cross-validation. The ensemble $\{1, \ldots, n\}$ is partitioned into K subsets $A_1, \ldots, A_K$. For a fixed penalization parameter and for each $k \in \{1, \ldots, K\}$, the sparse EM algorithm is applied to all points except those of A_k. The log-likelihood of the estimated model is then computed on the points of A_k. This is repeated for several values of the penalization parameter λ, and the value of λ maximizing the computed log-likelihood is selected.

4 Criteria to assess the methodology quality

In this section, 5 criteria are proposed to both asses the quality of the proposed modelling and to determine the number d of selected components in the decomposition. The way these criteria can be used to determine d is detailed in Sections 5.1 and 5.2. The first two criteria address specifically the functional decomposition step while the following three criteria evaluate the whole uncertainty modelling methodology.

4.1 Criteria for the functional decomposition step

The functional decomposition of the functional variables is performed with two objectives: retaining the features which are the most important for the approximation and the most correlated to the covariate. Hence, two criteria are defined to evaluate the ability of the functional decomposition to answer these two objectives. To assess the approximation quality of the variables on the basis, the first criterion is the explained variance. Let us denote the discretized versions of the functions by $\mathbf{f}_{1,j}, \ldots, \mathbf{f}_{m,j}$ for $j = 1, \ldots, n$ and their approximation by $\hat{\mathbf{f}}_{1,j}, \ldots, \hat{\mathbf{f}}_{m,j}$. The explained variance, denoted as criterion C_1, is then defined by this expression:

$$C_1 = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\left(\mathbf{f}_{i,j} - \hat{\mathbf{f}}_{i,j}\right)^T\left(\mathbf{f}_{i,j} - \hat{\mathbf{f}}_{i,j}\right)}{\sum_{i=1}^{m}\sum_{j=1}^{n}\left(\mathbf{f}_{i,j} - \bar{\mathbf{f}}_{i}\right)^T\left(\mathbf{f}_{i,j} - \bar{\mathbf{f}}_{i}\right)}, \tag{12}$$

with $\bar{\mathbf{f}}_i = \frac{1}{n}\sum_{j=1}^{n}\mathbf{f}_{i,j}$.

In order to quantify how well the link between the covariate and the coefficients is preserved, a metamodel (Sacks et al., 1989) can be used to predict the covariate as a function of the coefficients. If the metamodel predicts efficiently the covariate, this could confirm that the functional variables features which explain the covariate are well captured by the decomposition coefficients. Among all the metamodel-based solutions (polynomials, splines, neural networks, etc.), we focus our attention on the Gaussian process model (Oakley and O'Hagan, 2002; Rasmussen and Williams, 2006). Many authors (e.g. Welch et al. 1992; Marrel et al. 2008) have shown how the Gp model can be used as an efficient emulator of code responses, even in high dimensional cases. The quality of the model can be assessed by the Q^2 coefficient. For a validation sample $Y_1, \ldots, Y_{n_t}$ with $n_t \in \mathbb{N}$, the Q^2 is defined as follows:

$$Q^2 = 1 - \frac{\sum_{j=1}^{n_t}\left(Y_j - \hat{Y}_j\right)^2}{\sum_{j=1}^{n_t}\left(Y_j - \bar{Y}\right)^2}, \tag{13}$$

where $\bar{Y} = \frac{1}{n_t}\sum_{j=1}^{n_t} Y_j$ is the output mean, and, for $j = 1, \ldots, n_t$, $\hat{Y}_j$ is the estimation of Y_j by the Gaussian process model. In practice, the Q^2 can be computed by cross-validation (Hastie and Tibshirani, 1990). In the validation procedure, the dataset is partitioned into n_f disjoint ensembles. At each of the

n_f steps, a metamodel is learned on the data from $n_f - 1$ ensembles and the metamodel is applied to the points in the only ensemble not used to train the metamodel. The Q^2 criterion is finally computed using all the metamodel outputs. It constitutes the second proposed criterion to assess the quality of the methodology and is denoted C_2. In the tests conducted in section 5, cross-validation with $n_f = 10$ partitions has been used to compute C_2 criterion.

4.2 Criteria for the whole uncertainty modelling methodology

Three criteria have been chosen to assess the whole methodology of probability density estimation. First, the estimated probability distribution function of the coefficients is evaluated. To this end, new samples of coefficients are simulated from the estimated GMM. Their joint probability density function is compared to the one of a test sample of coefficients using a multivariate goodness-of-fit test. The selected test is a kernel-based two-sample goodness-of-fit test, which has been developed by Fromont et al. (2012). This test has been chosen among all existing multivariate goodness-of-fit test because it is proven to be exactly of level α and not only asymptotically. The test is carried out on multiple pairs of test basis and simulated samples of coefficients. The proposed criterion, denoted as C_3^a, is then the acceptance rate of the goodness-of-fit over these multiple runs.

The second criterion evaluates the methodology ability to reproduce the correlations between the functional variables. The studied correlations are pointwise correlations at each point of I. The $\frac{m(m-1)}{2}$ pointwise correlation between variables f_i and f_j for $i, j = 1, \ldots, m$, $i \neq j$, is defined in this way:

$$c_{i,j}(t) = \mathrm{Corr}\left(f_i(t), f_j(t)\right), t \in I. \tag{14}$$

The test basis is composed of realizations of the functional variables and the simulated basis contains functions simulated using the characterization methodology. The mean square error between the pointwise correlations of the test and those of the simulated bases is used as criterion and is noted C_3^b, for $i, j = 1, \ldots, m$:

$$C_3^b = \int_I \left(c_{i,j}(t) - \hat{c}_{i,j}(t)\right)^2 dt. \tag{15}$$

Finally, the ability of the methodology to reproduce the behaviour of the covariate is also tested. Similarly to the first criterion C_3^a, a goodness-of-fit test is used to evaluate the estimated probability density function of the covariate. Test samples of the covariate are computed by applying the model $\mathcal{M}$ to known realizations of $(f_1, \ldots, f_m)$, and simulated samples of covariates are computed by applying the model $\mathcal{M}$ to functions simulated with the characterization methodology. The Kolmogorov-Smirnov two-sample test (Conover, 1971) is applied between multiple pairs of simulated and test samples of the covariate. This test is a classical, simple and efficient one-dimensional goodness-of-fit test. The third criterion C_3^c is defined as the acceptance rate of all these tests.

5 Applications

In this section, the proposed methodology is applied to two test cases: an simulated model and a nuclear reliability test case. Different options of the methodology are studied and compared: the normalization factors for SPCA decomposition (see section 2.1), the different decomposition methods (see section 2) and the GMM estimation algorithms (see section 3). Moreover, the benefit of the simultaneous decompositions is shown. In the tests presented below, the algorithm proposed in section 3.2 and the algorithm developed by Krishnamurthy are called respectively sEM2 and sEM in the following. The sEM2 algorithm with penalization matrix $P^{(1)}$, $P^{(2)}$ or $P^{(3)}$ is called respectively sEM2.1, sEM2.2, sEM2.3.

5.1 Simulated example

The presented characterization methodology is tested in this section on an simulated model. The two studied temporal random variables are defined by these equations:

$$\begin{aligned} f_1(t, A_1, A_2, A_3) &= 0.8 A_2 BB(t) + A_1 + c_1(t) + h(t, A_3) \\ f_2(t, A_1, A_2, A_3) &= A_2 BB(t) + A_1 + c_2(t, A_3) \end{aligned}$$

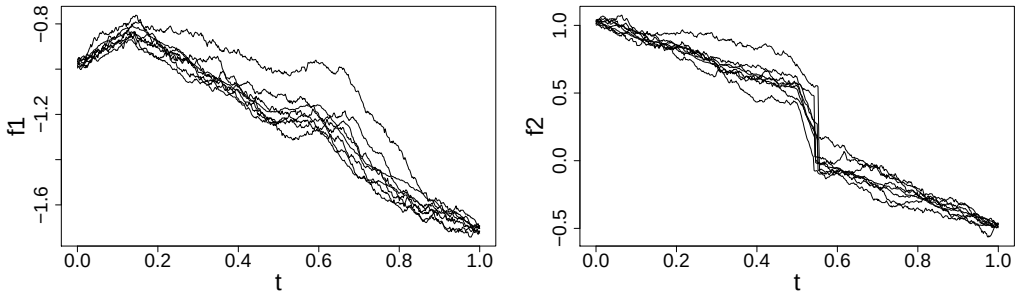


Figure 1: Simulated example: samples of 600 realizations of functional variables f_1 (left) and f_2 (right).

where t is a continuous variable defined on $I = [0, 1]$. The random variables A_1, A_2 and A_3 follow independent uniform laws on respectively $[0, 0.5]$, $[0.05, 0.2]$ and $[2, 3]$. The functions h, c_1 and c_2 are defined as follows, where BB is a Brownian bridge.

$$\begin{aligned} h(t, A_3) &= 0.15\left(1 - \left|\frac{t - 100A_3}{60}\right|\right) \\ c_1(t) &= \begin{cases} t - 1 & \text{if } t < \frac{35}{256} \\ \frac{93}{128} - t & \text{otherwise} \end{cases} \\ c_2(t, A_3) &= \begin{cases} 1 - t & \text{if } t < 0.5 \\ \frac{64}{5A_3} - 0.5t & \text{if } 0.5 < t < 0.5 + \frac{5A_3}{256} \\ 0.5 - t & \text{otherwise} \end{cases} . \end{aligned}$$

The covariate Y is defined as the output of the function $\mathcal{M}$:

$$\begin{aligned} Y(A_1, A_2, A_3) &= \mathcal{M}(f_1(., A_1, A_2, A_3), f_2(., A_1, A_2, A_3)) \\ &= \int_0^1 (f_1 + f_2)(t, A_1, A_2, A_3)dt \end{aligned} \tag{16}$$

A sample of $n = 600$ realizations of the triplet $(A_1^{(j)}, A_2^{(j)}, A_3^{(j)})$ is available and provides 600 realizations $f_{i,j} = f_i(., A_1^{(j)}, A_2^{(j)}, A_3^{(j)})$, $i \in \{1, 2\}$ $j \in \{1, \ldots, n\}$ of the two variables. These realizations constitute the learning sample. The corresponding outputs of $\mathcal{M}$, $Y_j = \mathcal{M}(f_{1,j}, f_{2,j})$, are also known. This sample is represented on Figure 1. The functions are discretized on $t_1, \ldots, t_p \in I$, with $p = 512$.

The sample of realizations is decomposed on the SPCA and SPLS bases. The three normalization factors given in section 2.1 are first compared. Concerning the SPCA basis, the normalization by the maximum of the functional variable yields different approximation errors for both variables, while the two others give equivalent weights to f_1 and f_2. In the following, the normalization by the sum of the standard deviations is used, but the normalization by the square root of the variances could be used as well.

The explained variances C_1 of SPCA and SPLS are compared with these of PCA and PLS respectively. The idea is to evaluate the benefit of simultaneous decomposition against non simultaneous decomposition for the same number of total components. For this, figures 3 and 2, represent in abscissa the number of components selected in SPLS (resp. SPCA) decomposition and in ordinate the number of components selected in the PLS (resp. PCA) of only one functional variable. For instance, the use of 2 components in the decomposition of f_1 and 3 in the decomposition of f_2 is compared to the use of 5 components in the simultaneours decomposition. The black and red curves represent the maximal number of components selected in the decompositions of each variable f_1 and f_2 separately such that these PLS (resp. PCA) decompositions have an explained variance lower or equal to the explained variance of the SPLS (resp. SPCA) decomposition, for each SPLS (resp. SPCA) basis size. The dotted line is the reference curve $y = x$. If the sum of the number of components of each PLS (resp. PCA), the blue line, is over the curve $y = x$, SPLS (resp. SPCA) gives better approximations of the curves for the same number of coefficients. For 3 (resp. 2) or more components, SPCA (resp. SPLS) better approximates the sample

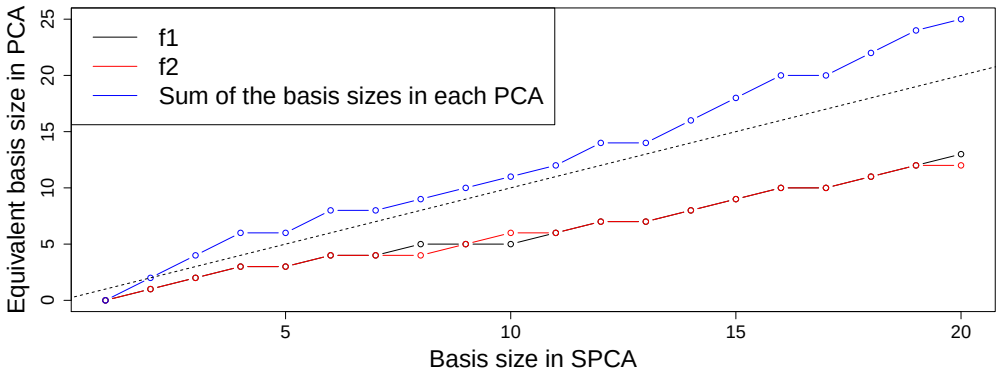


Figure 2: Simulated example: the maximal number of PCA components of each variable such that the sum of the explained variances of these decompositions is lower than the explained variance of SPCA (red and black curves). Sum of black and red curves (blue curve). Reference curve $y = x$ (dotted line).

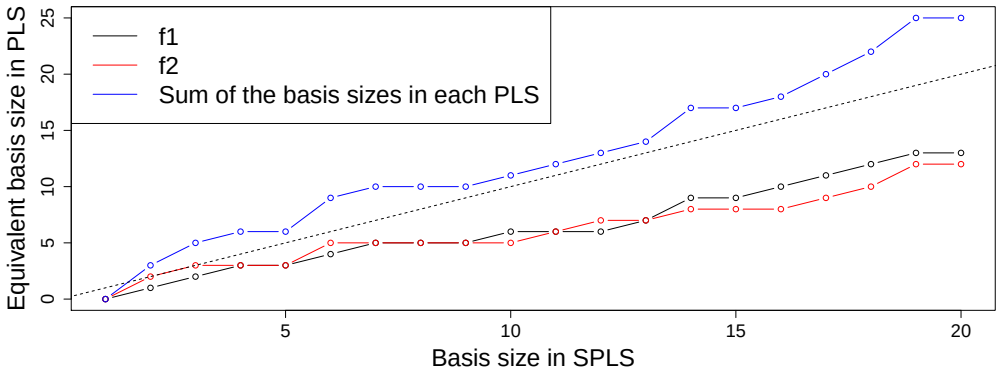


Figure 3: Simulated example: the maximal number of PLS components of each variable such that the sum of the explained variances of these decompositions is lower than the explained variance of SPLS (red and black curves). Sum of black and red curves (blue curve). Reference curve $y = x$ (dotted line).

than two individual PCAs (resp. PLS) on f_1 and f_2 separately for an equal total number of components. The simultaneous decompositions are therefore more efficient than individual ones when more than one component is retained.

Then, SPLS and SPCA are compared in Figures 4 and 5 based on criteria C_1 and C_2. In Figure 4, the percentage of explained variance as defined in equation (12) is drawn as a function of the basis size for SPCA (black circles) and for SPLS (red crosses). The explained variance of SPCA is higher than the one of SPLS by definition of SPCA. However, for basis with more than 8 components, the difference between the two explained variances becomes quite low. A Gaussian process model is fitted between the coefficients of SPLS (resp. SPCA) and the covariate for different basis sizes. The Figure 5 shows the Q^2 of this Gaussian process model, defined in equation (13). The Q^2 of SPLS is higher than the one of SPCA. The difference is low for basis with more than 5 components. SPLS decomposition preserves more the features of the two functional variables which are linked to the covariate than SPCA. It seems to be a better compromise between the two objectives.

In the following, we focus on the SPLS decomposition. For different basis sizes, the probability density functions of the coefficients is estimated with the five different estimation methods, EM, sEM, sEM2.1, sEM2.2 and sEM2.3. The criteria C_3^a, C_3^b and C_3^c presented in section 4.2 are computed for these different basis sizes. The criteria C_3^a and C_3^c are computed with 10 test basis and 10 simulated

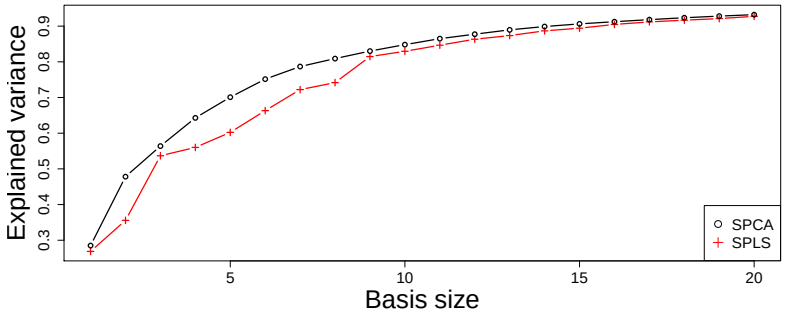


Figure 4: Simulated example - criterion C_1: explained variance by SPCA (black circles) and SPLS (red crosses) as a function of the decomposition basis size.

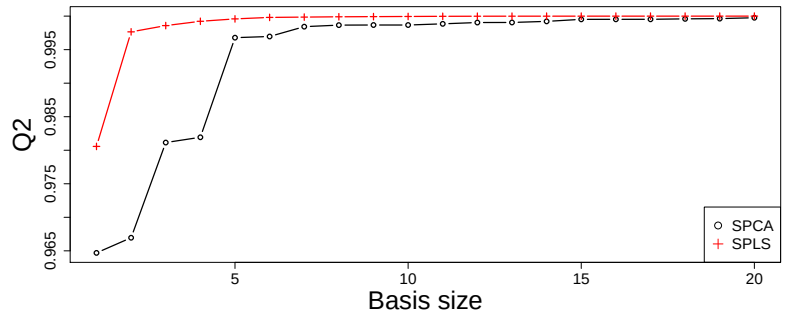


Figure 5: Simulated example - criterion C_2: Q^2 coefficient of the Gaussian process model between the coefficients of SPCA (black circles) or SPLS (red crosses) and the covariate Y as a function of the decomposition basis size.

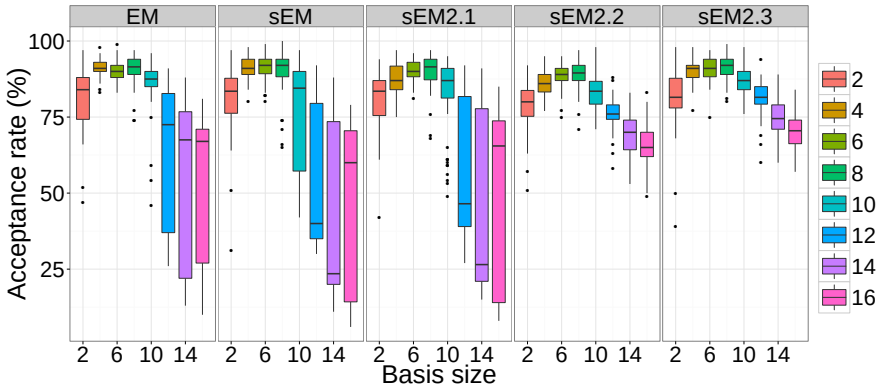


Figure 6: Simulated example - criterion C_3^a: boxplot of the acceptance rates for the goodness-of-fit test between the estimated coefficients probability density and the true density as a function of the basis size and for each of the 5 estimation algorithms.

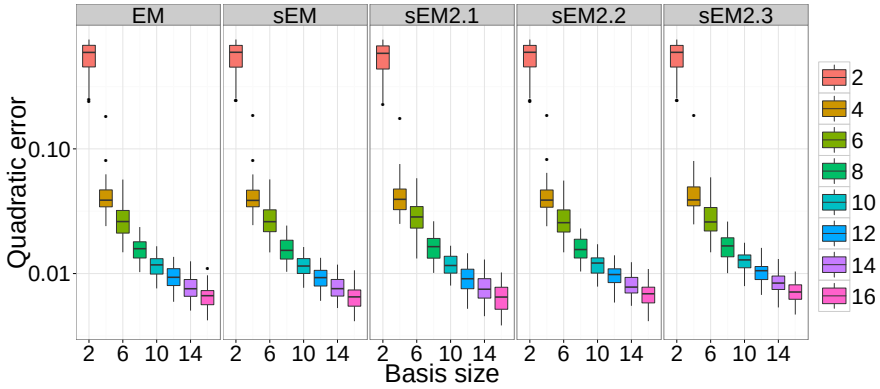


Figure 7: Simulated example - criterion C_3^b: boxplot of the mean square errors for the pointwise correlations between etimates of f_1 and f_2 as a function of the basis size and for each of the 5 estimation algorithms.

samples, containing 10^3 realizations. For the criterion C_3^b, one test basis with 10^5 realizations and 10 simulated samples of 10^3 realizations are used. The values of the three criteria are computed on 50 different learning basis of size n. The boxplots of these values are given as a function of the basis size for each estimation algorithm in Figures 6, 7 and 8.

In Figure 6, the criterion C_3^a evolves in the same manner for each estimation algorithm. For each algorithm, the acceptance rate of the goodness-of-fit test is increasing until a basis of size 8, then it decreases quickly. Compared to the acceptance rates for sEM2.2 and sEM2.3 algorithms, the acceptance rates for EM, sEM and sEM2.1 have much more variability and are lower for basis with 12 or more functions. Therefore, the use of sEM2 with penalization matrices 2 and 3, without penalization on the diagonal, improves the results in higher basis sizes. In Figure 7, the criterion C_3^b is represented in logarithmic scale. It decreases quickly with the basis size. Moreover, the errors are quite low for high basis sizes as it is around 0.01 for a decomposition basis with 8 or 10 functions. Finally, the values of the criterion C_3^c, in Figure 8, are quite constant for all algorithms except sEM2.1. Moreover, they are about 90% or higher for these algorithms. On the contrary, the acceptance rates of algorithm sEM2.1 vary much more, and even decrease as a function of the basis size, at first. Moreover, studies conducted on learning samples of sizes from $n = 200$ to 1000 show that the three criteria increase as n increases.

With the five considered algorithms, the highest basis size such that the median of the criterion C_3^a is above 80% (resp. 90%) is 10 (resp. 8). The criterion C_3^c is over 90% with all algorithms except sEM2.1 for which it is about 85%. The criterion C_3^b is about 0.016 for bases with 8 functions and 0.012 for bases with 10 functions. Criteria C_3^a, C_3^b and C_3^c can be used to choose the decomposition basis size. As the

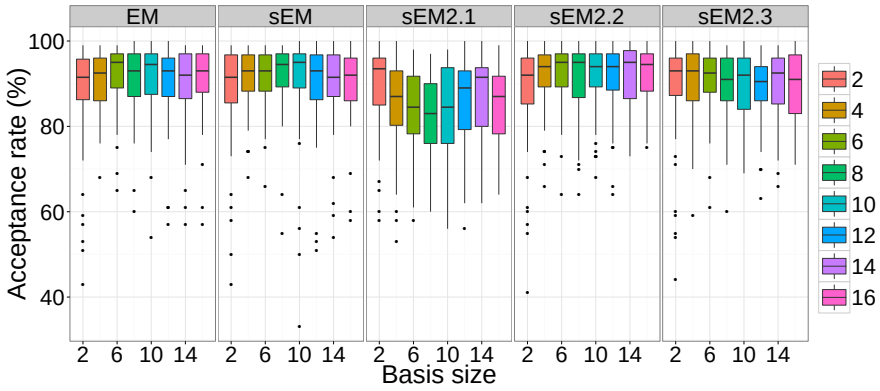


Figure 8: Simulated example - criterion C_3^c: boxplot of the acceptance rates for the goodness-of-fit test between the estimated covariates probability density and the true density as a function of the basis size and for each of the 5 estimation algorithms.

values of both C_3^b and C_3^b criteria seem acceptable for bases with 8 or 10 functions, the chosen value for the basis size could be here 8 or 10, depending on what we prefer to better represent: the dependence between the functional variables or the coefficients probability distribution. If 10 is chosen, the criterion C_3^a is a little worsened but C_b^3 is improved. A 8 functions basis is used in the rest of the section as it minimizes the error on the estimated distribution.

In this simulated example, the proposed methodology has proven its efficiency to characterize the functional variables and their link to the covariate. The sparse estimation algorithm sEM2 with penalization matrices 2 and 3 seems to improve the criterion C_3^a for higher basis sizes. However, overall, there are few differences between the various estimation methods. This may be due to the low number of parameters in this example, because the sparse methods are the most helpful when the ratio between the number of parameters to be estimated and the learning data size is high. For instance, for a decomposition basis of 8 functions and 3 clusters in the GMM, the number of parameters is only 89.

Finally, this uncertainty modelling method can be used to estimate probabilities for the studied variables to exceed a given threshold. No error bound is available for this estimation method, so that the efficiency of the method is not theoretically guaranteed. A bootstrap method could be used to measure the uncertainty on the computed probability and thus assess method stability. Let us define a probability to estimate:

$$\begin{aligned} p &= P\Big(\left(t \in I : f_1(t, A_1, A_2, A_3) > -0.8 \right) \cup \\ &\quad \left(\min_{t \in I} \left(f_2(t, A_1, A_2, A_3) < \frac{1}{2} \right) < \frac{270}{512} \right) \cup \left(Y < -1 \right) \Big). \end{aligned}$$

The reference value for p is computed on a sample of 10^5 realizations of the functional variables and covariate. The computed value, 0.272, is considered as the true value in the following. An estimation $\hat{p}$ of p is estimated with a sample of 10^5 realizations of the estimated GMM. The relative approximation error

$$100 \frac{|p - \hat{p}|}{p}$$

is computed for 50 learning bases and different decomposition basis sizes. Figure 9 represents this absolute error as a function of the basis size and for each estimation algorithm. The obtained ranges of errors show that it is not possible to have a very precise estimation of the probability but rather a good estimate of its order of magnitude. For instance for bases of size 10, the medians of the errors are between 16 and 20%, which corresponds to an error of about 0.05. This analysis is not surprising considering that the probability p has been estimated using a method designed to approximate a whole density probability function and not uniquely a single probability, and considering that a sample of only 600 realizations is used. EM and sEM algorithms give slightly lower errors than other algorithms. Moreover, the convergence to the real value is first very fast. Then, the decrease of the error slows down for higher basis sizes.

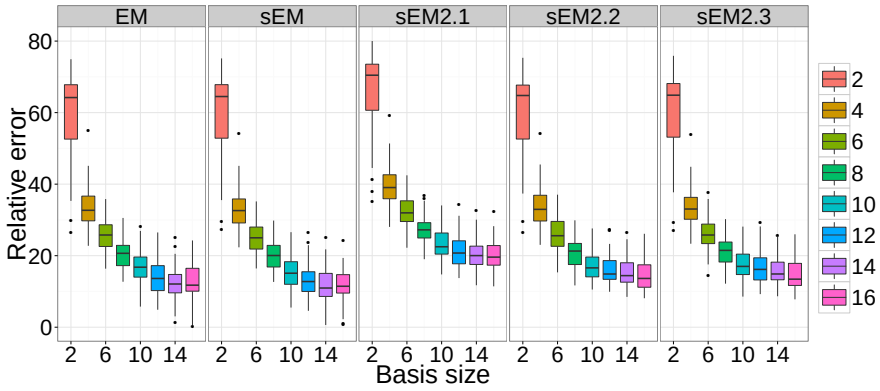


Figure 9: Simulated example: boxplot of the relative approximation error, $100\,|p-\hat{p}|\,/p$, between the estimated probability $\hat{p}$ and p as a function of the basis size and for each of the 5 estimation algorithms.

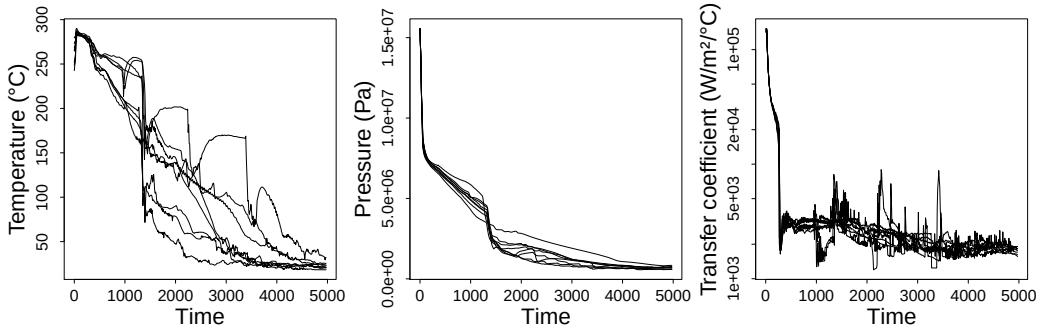


Figure 10: Nuclear reliability example: sample of 400 curves of temperature (left panel), pressure (center) and transfer coefficient (right).

5.2 Nuclear reliability application

In the scope of nuclear reliability and nuclear power plant lifetime program, physical modelling tools have been developed to assess the component reliability of nuclear plants in numerous scenarios of use or accident. In the framework of nuclear plant risk assessment studies, the evaluation of component reliability during accidental conditions is a major issue required for the safety case. A thermal-hydraulic system code (code 1) models the behaviour of the considered component subjected to highly hypothetical accidental conditions. Three functions of time, fluid temperature, transfer coefficient and pressure are computed. Then, a thermal-mechanical code (code 2), taking as input code 1 results along with some mechanical scalar parameters, calculates the absolute mechanical strength of the component and the mechanical applied load. From these two quantities, a safety criterion Y is deduced. In accidental conditions, the component behaviour depends on several uncertain parameters which are input variables of the two computer codes. The functional outputs of code 1 are thus uncertain too. The objective is here to characterize the three dependent functional random variables, temperature, pressure and transfer coefficient, linked to the safety criterion. A learning dataset of 400 temperature, pressure and transfer coefficient functions is available. The safety criteria corresponding to the available functions are computed with constant mechanical parameters. Penalizing values from a safety point of view have been given to mechanical input parameters of code 2. A sample of the learning dataset is represented in Figure 10.

SPLS and SPCA decompositions are first compared on the three functional variables. The safety criterion is considered as the covariate in the SPLS decomposition. To apply both simultaneous decompositions, the functions are discretized on a regular grid of $p = 512$ points. The three normalization factors (see section 2.1) are first compared for SPCA. As in section 5.1, SPCA with the normalization by the maximum of the functional variable favours one variable. The normalization by the sum of the standard deviations is used here.

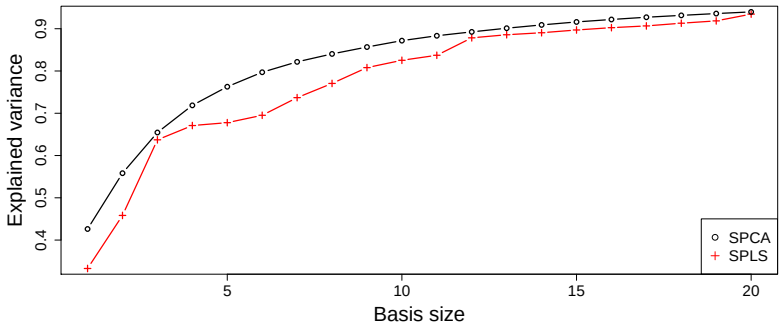


Figure 11: Nuclear reliability example - criterion C_1: explained variance by SPCA (black circles) and SPLS (red crosses) as a function of the decomposition basis size.

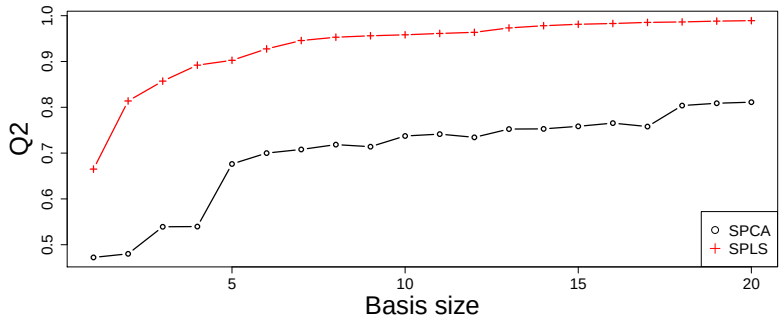


Figure 12: Nuclear reliability example - criterion C_2: Q^2 coefficient of the Gaussian process model between the coefficients of SPCA (black circles) or SPLS (red crosses) and the covariate Y as a function of the decomposition basis size.

Figure 11 represents the variance explained by the SPLS (red crosses) and SPCA (black circles) decompositions. Variance explained by SPCA is above the variance explained for every basis size. The explained variance of SPLS decomposition is though quite close to the one of SPCA and becomes closer for basis sizes higher than 14. In Figure 12, the Q^2 of the Gaussian process model between the coefficients of the decomposition and the covariate is represented. The Q^2 computed for SPLS clearly outperforms the Q^2 of SPCA. As it was expected, SPLS decomposition retains more than SPCA functional variables features which are correlated to the covariate. SPLS decomposition is used in the rest of the section, as it is a good compromise between the two objectives of the characterization.

The probability density function of the SPLS coefficients is estimated using the EM, sEM and sEM2 algorithms for different basis sizes. The criteria C_3^a and C_3^b, described in section 4.2, are computed. They are averaged on 50 simulated samples of size 1000. These samples are compared to a test dataset of 1000 functions. Figures 13 and 14 show respectively the criteria C_3^a and C_3^b as a function of the decomposition basis size. The results for algorithms EM, sEM, sEM2.1, sEM2.2 and sEM2.3 are plotted respectively in black, red, green, dark blue and light blue. In Figure 13, the acceptance rates are quite high for basis sizes lower than 10. For higher sizes, the rates decrease quickly and the sEM2 algorithm with the three penalization matrices performs much better than EM and sEM algorithms. In Figure 14, the mean square errors on the correlation between temperature and pressure, temperature and transfer coefficient, and pressure and transfer coefficient are presented from left to right. The errors decrease quickly as a function of the basis size. Moreover, they are quite low for basis sizes over 6 or 8. From these two criteria, a decomposition basis with 10 functions is chosen, as it gives an acceptance rate about 80% for each algorithm and as the errors on the correlations are quite low for this basis size.

For the criterion C_3^c, such intensive tests could not have been applied because of the computation time of code 2. However, criterion C_3^c has been computed for eight retained coefficients, its value is

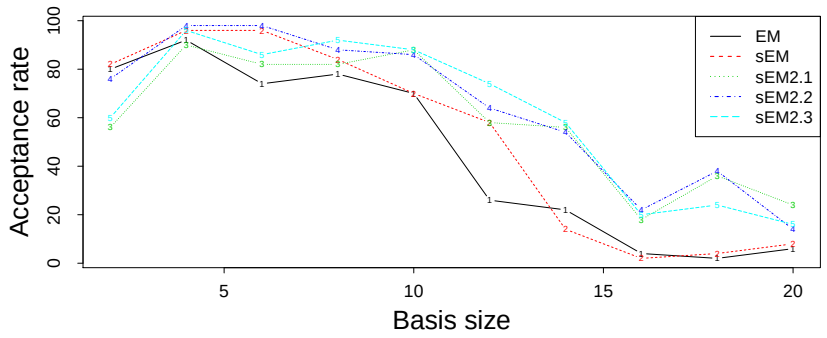


Figure 13: Nuclear reliability example - criterion C_3^a: acceptance rates for the goodness-of-fit test on the estimated coefficients density as a function of the basis size.

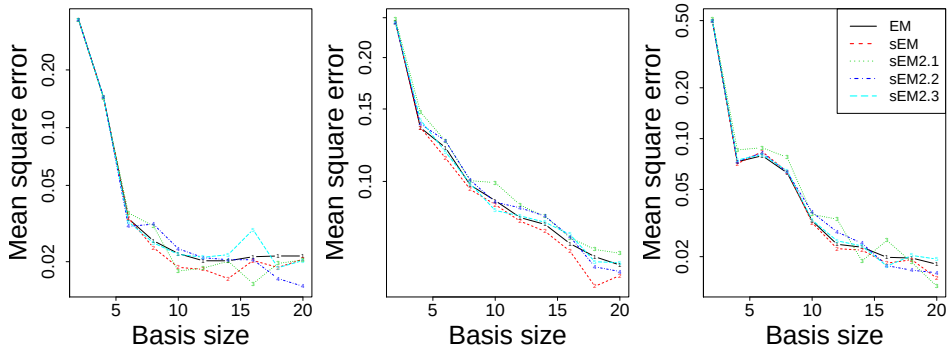


Figure 14: Nuclear reliability example - criterion C_3^b: mean square errors for the pointwise correlations as a function of the basis size.

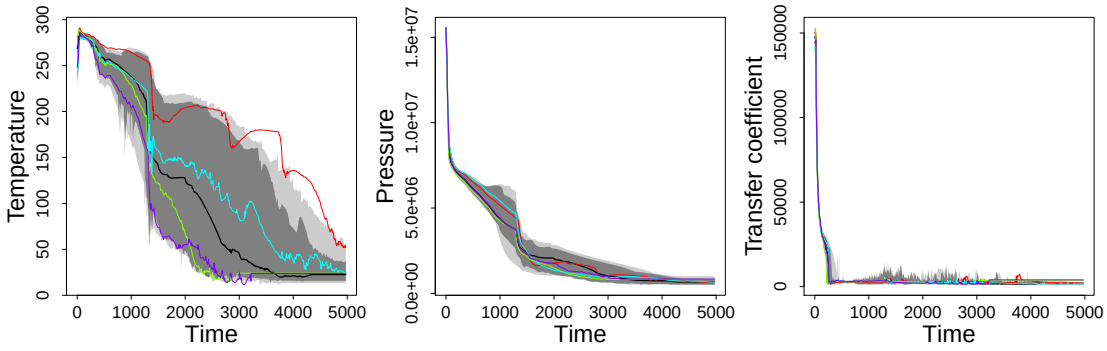


Figure 15: Nuclear reliability example: proposed extension of the Highest Density Region (HDR) boxplot of the temperature, the pressure and the transfer coefficient functional variables.

approximately 98%, based on 50 Kolmogorov-Smirnov tests. Hence, the covariate seems to be well reproduced. For the sake of comparison, criterion C_3^c has also been computed with models based on 9 and 11 components, averaged on 50 goodness-of-fit tests. Its values are respectively 92% and 94%. Both are slightly inferior to the value obtained with 8 components. With 9 components, even though the Gaussian mixture model is more accurate, less components are used, thus information is missing in the model. On the contrary, the lower value obtained with 11 components is due to the fact that the Gaussian mixture model is less accurate, and that its lower accuracy is not enough counterbalanced by the higher number of components.

Finally, a possible application of this uncertainty modelling methodology could be to provide a tool to visualize simultaneously several dependent functional data. For this, we propose to adapt the visualization technique developed by Hyndman and Shang (2010) and called Highest Density Region (HDR) boxplot. This technique is an extension of boxplot visualization to functional data in the sense that it helps identifying a central curve, zones containing a certain proportion (e.g. 50%) of most central curves and outlying curves. However, its shape (as in the boxplot bivariate extension) is very different to the one of boxplot. It is worth noting that it has already been applied in the context of nuclear reliability study in Popelin and Iooss (2013). The method of Hyndman and Shang (2010) is based on the uncertainty characterization of the functional variables. They propose to decompose the functional data on a PCA basis and to select the first two basis functions. Then, the joint probability of the coefficients is estimated with a kernel density estimation (Rosenblatt, 1956). The estimated probability density function $\hat{f}$ of the coefficients is used as a probability density function of the corresponding functions. The function whose coefficients has the highest density is called the functional mode. Conversely, the functions whose corresponding coefficients have the lowest density are considered as outliers. A HDR is defined as

$$R_\alpha = \{x : \hat{f}(x) \geq f_\alpha\}, \tag{17}$$

where f_α is such that $\int_{R_\alpha} \hat{f}(x)\mathrm{d}x = 1 - \alpha$, and $0 < \alpha < 1$ is a probability. The points outside $R_{1\%}$ are considered as outliers. We propose here to replace the characterization step by the methodology proposed in this paper. The rest of the visualization method is unchanged. This modified version of the HDR boxplot is applied simultaneously to the temperature, the pressure and the transfer coefficient and the result is represented on Figure 15. A SPCA basis with 8 functions is used in the characterization step. The black curve is the functional mode. The dark and light gray zones are the regions bounded by all curves whose corresponding coefficients are in $R_{50\%}$ and $R_{1\%}$ respectively. The colored curves are outliers, *i.e.* curves whose corresponding coefficients are not in $R_{1\%}$.

6 Conclusion

In this article, we have proposed a methodology to model the uncertainties of several functional random variables. This method allows to deal simultaneously with several dependent functional variables and to address the specific case where these variables are linked to a scalar or vectorial variable, called covariate.

In this case, the two objectives of the method are thus to preserve the most important characteristics of the functional variables and their features which best explain the covariate. The proposed method is composed of two main steps: the decomposition of the functional variables on a reduced functional basis and the modelling of the probability density function of the coefficients of the variables in the functional basis. The first step is carried out by the Simultaneous Principal Component Analysis, if the variables are not linked to a covariate and otherwise by the developed Simultaneous Partial Least Squares decomposition. The latter one has the advantage to maximize the covariance between the covariate and the approximated functional variables. In the second step, the joint probability density function of the selected coefficients is modelled by a Gaussian mixture model. A new algorithm, using Lasso penalization, is proposed in this paper to estimate the parameters of the Gaussian mixture model with sparse covariance matrices and hence reduce the number of model parameters to be estimated.

This uncertainty modelling methodology has been successfully applied to an simulated example with two functional random variables and to a nuclear reliability test case. In both presented test examples, the SPLS algorithm has been shown to better preserve the variable features which explain the covariate, and the sparse algorithm has improved the estimation of the GMM parameters. A possible application of the methodology has been exposed: the joint probability for the functional variables and the covariate to exceed thresholds is estimated using the probability density function estimated in the methodology. Another presented application is to use the characterization methodology to build a visualization tool for functional data. Finally, if the covariate is the output of a computer code whose inputs are the functional variables, it enables to simulate new samples of inputs and thus to run uncertainty propagation or sensitivity analysis studies on the computer code. However, tests, which are not displayed here, have shown that the ability of the method to reproduce the covariate distribution depends strongly on the definition of the covariate. This is the topic of future works.

References

Anstett-Collin, F., Goffart, J., Mara, T., Denis-Vidal, L.: Sensitivity analysis of complex models: Coping with dynamic and static inputs. Reliability Engineering & System Safety, 134, 268 – 275 (2015)

Bien, J., Tibshirani, R.J.: Sparse estimation of a covariance matrix. Biometrika, 98, 807–820 (2011)

Bongiorno, E., Goia, A.: Some insights about the small ball probability factorization for hilbert random elements (2015). http://arxiv.org/abs/1501.04308

Bongiorno, E., Goia, A.: Classification methods for hilbert data based on surrogate density. Computational Statistics & Data Analysis, 99, 204 – 222 (2016). http://arxiv.org/abs/1506.03571

Bongiorno, E.G., Salinelli, E., Goia, A., Vieu, P.: Contributions in infinite-dimensional statistics and related topics. Societa Editrice Esculapio (2014)

Conover, W.J.: Practical Nonparametric Statistics (1971)

De Rocquigny, E., Devictor, N., Tarantola, S.: Uncertainty in industrial practice. Wiley (2008)

Delaigle, A., Hall, P.: Defining probability density for a distribution of random functions. The Annals of Statistics, pp. 1171–1193 (2010)

Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 39, 1–38 (1977)

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

Friedman, J., Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical Lasso. Biostatistics, 9, 432–441 (2008)

Fromont, M., Laurent, B., Lerasle, M., Reynaud-Bouret, P.: Kernels based tests with non-asymptotic bootstrap approaches for two-sample problem. In: 25th Annual Conference on Learning Theory, 23, 1–22 (2012)

Ghanem, R.G., Spanos, P.D.: Stochastic finite elements: a spectral approach. Springer (1991)

Goia, A., Vieu, P.: An introduction to recent advances in high/infinite dimensional statistics. Journal of Multivariate Analysis, 146, 1 – 6 (2016). Special Issue on Statistical Models and Methods for High or Infinite Dimensional Spaces

Hastie, T., Tibshirani, R.: Generalized Additive Models. Chapman and Hall/CRC (1990)

Helton, J., Johnson, J., Sallaberry, C., Storlie, C.: Survey of sampling-based methods for uncertainty and sensitivity analysis. Reliability Engineering & System Safety, 91, 10-11, 1175 – 1209 (2006)

Horváth, L., Kokoszka, P.: Inference for functional data with applications, volume 200. Springer Science & Business Media (2012)

Höskuldsson, A.: PLS regression methods. Journal of chemometrics, 2, 211–228 (1988)

Hyndman, R.J., Shang, H.L.: Rainbow plots, bagplots, and boxplots for functional data. Journal of Computational and Graphical Statistics, 19, 29–45 (2010)

Jacques, J., Preda, C.: Functional data clustering: a survey. Advances in Data Analysis and Classification, 8, 3, 231–255 (2014a)

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

Loève, M.: Probability theory. Springer (1955)

Ma, X., Zabaras, N.: Kernel principal component analysis for stochastic input model generation. Journal of Computational Physics, 230, 19, 7311–7331 (2011)

Marrel, A., Iooss, B., Van Dorpe, F., Volkova, E.: An efficient methodology for modeling complex computer codes with gaussian processes. Computational Statistics & Data Analysis, 52, 10, 4731–4744 (2008)

Mclachlan, J., Krishnan, T.: The EM algorithm and extension. Wiley inter-science (1997)

Oakley, J., O'Hagan, A.: Bayesian inference for the uncertainty distribution of computer model outputs. Biometrika, 89, 4, 769–784 (2002)

Pearson, K.: On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2, 559–572 (1901)

Popelin, A.L., Iooss, B.: Visualization tools for uncertainty and sensitivity analyses on thermal-hydraulic transients. In: SNA + MC 2013 - Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo (2013)

Ramsay, J.O., Silverman, B.W.: Functional Data Analysis. Springer Series in Statistics. Springer (2005)

Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press (2006)

Rosenblatt, M.: Remarks on some nonparametric estimates of a density function. The Annals of Mathematical Statistics, 27, 3, 832–837 (1956)

Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P.: Design and analysis of computer experiments. Statistical Science, 4, 4, 409–423 (1989). doi:doi:10.2307/2245858

Schwarz, G.: Estimating the dimension of a model. The Annals of Statistics, 6, 461–464 (1978)

Scott, D.W.: Multivariate density estimation: theory, practice, and visualization, volume 383. John Wiley & Sons (2009)

Van Deun, K., Smilde, A., van der Werf, M., Kiers, H., Van Mechelen, I.: A structured overview of simultaneous component based data integration. BMC Bioinformatics, 10, 246–261 (2009)

Wan, J., Zabaras, N.: A probabilistic graphical model based stochastic input model construction. Journal of Computational Physics, 272, 664–685 (2014)

Wang, H.: Coordinate descent algorithm for covariance graphical Lasso. Statistics and Computing, 6, 1–9 (2013)

Welch, W.J., Buck, R.J., Sacks, J., Wynn, H.P., Mitchell, T.J., Morris, M.D.: Screening, predicting, and computer experiments. Technometrics, 34, 1, 15–25 (1992)

Wold, H.: Estimation of Principal Components and Related Models by Iterative Least squares, pp. 391–420. Academic Press (1966)

Wu, C.F.J.: On the convergence properties of the EM algorithm. The Annals of Statistics, 11, 95–103 (1983)